CoNLL 2025

The 29th Conference on Computational Natural Language Learning (CoNLL 2025)

Proceedings of the Conference

July 31 - August 1, 2025

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

Platinum

DeepMind

©2025 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

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

ISBN 979-8-89176-271-8

Introduction

Welcome to CoNLL, the annual conference organized by the ACL Special Interest Group on Natural Language Learning (SIGNLL). As in previous years, CoNLL is a single-track conference with a focus on theoretically, cognitively and scientifically motivated approaches to computational linguistics, particularly exploring interactions between theoretical issues and computational modeling. This year, CoNLL is held alongside ACL 2025.

For our conference program, we received a total of 217 archival and 20 non-archival paper submissions, out of which we accepted 40 archival submissions (18%) and 6 non-archival submissions (30%). In addition to direct submissions, our program includes one presentation of a Findings paper accepted to the ACL main conference. Accepted papers are presented in two poster sessions and four oral sessions. In addition, the program includes a discussion session as well as two invited talks, by Raquel Fernández and Jean-Rémi King.

Our program committee consisted of 22 area chairs and 189 reviewers, whom we'd like to thank very much for providing constructive reviews and thoughtful recommendations. Special thanks go to our publication chairs, Emily Cheng and Selina Meyer, and publicity chairs, Snigdha Chaturvedi and Anvesh Rao Vijjini, for their crucial support. The SIGNLL board and last year's organizers provided numerous useful tips and important information. We are grateful for Google DeepMind's sponsoring of the Best Paper Award.

And, last but very much not least, a big thank you to the authors and the community, without whom this conference would not exist.

Gemma and Michael
CoNLL 2025 conference co-chairs

Organizing Committee

Program Chairs

Gemma Boleda, Universitat Pompeu Fabra / ICREA, Spain
Michael Roth, University of Technology Nuremberg, Germany

Publicity Chairs

Snigdha Chaturvedi, University of North Carolina at Chapel Hill, USA
Anvesh Rao Vijjini, University of North Carolina at Chapel Hill, USA

Publication Chairs

Emily Cheng, Universitat Pompeu Fabra, Spain
Selina Meyer, University of Technology Nuremberg, Germany

Program Committee

Program Chairs

Gemma Boleda, ICREA and Universitat Pompeu Fabra
Michael Roth, University of Technology Nuremberg

Area Chairs

Christian Bentz, Universität Passau
Zhenguang Cai, Chinese University of Hong Kong
Tanise Ceron, Bocconi University
Jackie CK Cheung, McGill University
Kilian Evang, Heinrich Heine University Düsseldorf
Agnieszka Falenska, University of Stuttgart
Aina Garí Soler, Télécom-Paris
Ximena Gutierrez, Universidad Nacional Autónoma de México
Dieuwke Hupkes, Meta
Xixian Liao, Barcelona Supercomputing Center
Brielen Madureira, University of Potsdam
Yohei Oseki, University of Tokyo
Sandro Pezzelle, University of Amsterdam
Emily Prud'hommeaux, Boston College
Carina Silberer, University of Stuttgart
Ece Takmaz, Utrecht University
Max Johannes Van Duijn, Leiden University
Tessa Verhoef, Leiden University
Yang Xu, University of Toronto
Wei Zhao, University of Aberdeen
Iria de-Dios-Flores, Universitat Pompeu Fabra

Reviewers

The list of all reviewers for CoNLL 2025 (in alphabetical order). Area chairs indicated outstanding reviewers in their track. These are marked *.

Omri Abend, Gulinigeer Abudouwaili, Rodrigo Agerri, Albina Akhmetgareeva, Afra Alishahi, Samuel Joseph Amouyal, Avinash Anand, Jacob Andreas*, Tatiana Anikina, David Arps*, Daiki Asami, Eleftherios Avramidis

Roberto Basili, Anna Bavaresco*, Christian Bentz*, Gábor Berend, Swarnadeep Bhar*, Debasmita Bhattacharya, Shohini Bhattasali, Arianna Bisazza*, Tatiana Bladier, Gosse Bouma, Tomas J.f. Breedveld*, David Broneske, Khac-Hoai Nam Bui

Jose Camacho-Collados*, Jie Cao, Spencer Caplan, Giovanni Cassani*, Tanise Ceron, Guanyi Chen, Long Chen, Xinchi Chen, Yiwen Chen, Emmanuele Chersoni, Pelin Çelikkol

Verna Dankers*, Forrest Davis, Marianne De Heer Kloots*, Mathieu Dehouck*, Shumin Deng, Aniket Deroy, Antoine Doucet, Rotem Dror, Xiaotang Du, Jonathan Dunn, Ritam Dutt*, Iria de-Dios-Flores

Yo Ehara, Saman Enayati

Abdellah Fourtassi, Dayuan Fu, Yingxue Fu*, Yoshinari Fujinuma, Richard Futrell

Lingyu Gao, Aina Garí Soler*, Mengshi Ge, Mohammad Reza Ghasemi Madani, Shinjini Ghosh, Maria Gini, Mario Giulianelli*, Goran Glavaš, Carlos-Emiliano González-Gallardo, Michael Eric Goodale, Karan Gupta

Sherzod Hakimov, Lifeng Han, Caren Han*, Daniel Hershcovich*, Mark Hopkins, Xiaoyu Hu

Cassandra L Jacobs, Abhik Jana, Ganesh Jawahar, Yichen Jiang

Kohei Kajikawa*, Ratna Kandala, N J Karthika, Marc A. Kastner, Sedrick Keh, Casey Kennington, Tracy Holloway King, Tom Kouwenhoven*, Chalamalasetti Kranti, Tatsuki Kuribayashi

Leo Laugier, Jochen L. Leidner*, Martha Lewis*, Junlin Li, Meng Li, Xia Li, Xixian Liao, Dongqi Liu, Jiangming Liu, Yunfei Luo, Kai Lv

Subhankar Maity*, Biswadip Mandal, Stella Markantonatou, Yuichiroh Matsubayashi, Yevgen Matusevych*, Maximilian Maurer, R. Thomas McCoy*, Kate McCurdy*, Robert Mercer*, Selina Meyer*, Yisong Miao, Timothee Mickus, Yusuke Miyao, Misael Mongiovì*, Manuel Montes, Steven Moran, Samuel Musker, Arianna Muti

Yee Man Ng, Vinh Van Nguyen, Dmitry Nikolaev, Tadashi Nomoto

Kazumasa Omura, Yohei Oseki, Jessica Ouyang*

Vishakh Padmakumar*, Minkyung Park, Olga Pelloni, Simon Petitjean, Tiago Pimentel*, Priya Pitre, Lidia Pivovarova, Flor Miriam Plaza-del-Arco*, Maja Popovic, Emily Prud'hommeaux*, Peter Van Der Putten

Siya Qi*

Ella Rabinovich, Alexandre Rademaker, Aida Ramezani, Anthony Rios, Jesse Roberts, Nathan Roll*, Carolyn Rose, Sumegh Roychowdhury, Adrián Rubio, Maria Ryskina

Philipp Sadler, Parisa Safikhani, Tanja Samardzic, Malaikannan Sankarasubbu, David Schlangen, William Schuler, Sabine Schulte Im Walde*, Carina Silberer*, Mong Yuan Sim, Kanishk Singh, Kevin Small*, Yingjin Song*, Shane Steinert-Threlkeld, Egon Stemle, Ruolin Su, Wenjun Sun

Thi Hong Hanh Tran

Michiel Van Der Meer, Bram Van Dijk, Jacob Hoover Vigly*, Prashanth Vijayaraghavan, Esaú Villatoro-tello, Pavlos Vougiouklis

Yuiga Wada, Jianyu Wang, Xiaomeng Wang, Franziska Weeber*, Alina Wróblewska*, Ting-Wei Wu

Qihui Xu

Zukang Yang, Roman Yangarber

Sina Zarrieß, Ningyu Zhang, Wei Emma Zhang, Xiang Zhang, Yuhan Zhang, Yuqing Zhang, Yusen Zhang, Guangzhen Zhao, Jiachen Zhao, Jin Zhao, Kai Zhao, Wei Zhao, Heike Zinsmeister*, Jinan Zou

Table of Contents

HKCanto-Eval: A Benchmark for Evaluating Cantonese Language Understanding and Cultural Comprehension in LLMs
Tsz Chung Cheng, Chung Shing Cheng, Chaak-ming Lau, Eugene Lam, Wong Chun Yat, Hoi On Yu and Cheuk Hei Chong 1

Quasi-symbolic Semantic Geometry over Transformer-based Variational AutoEncoder
Yingji Zhang, Danilo Carvalho and Andre Freitas 12

LawToken: a single token worth more than its constituents
Yu-Hsiang Tseng, Hsin-Yu Chou and Shu-Kai Hsieh 30

Interpersonal Memory Matters: A New Task for Proactive Dialogue Utilizing Conversational History
Bowen Wu, Wenqing Wang, Lihaoran Lihaoran, Yunhan Deng, Ying Li, Jingsong Yu and Baoxun Wang 47

WinoWhat: A Parallel Corpus of Paraphrased WinoGrande Sentences with Common Sense Categorization
Ine Gevers, Victor De Marez, Luna De Bruyne and Walter Daelemans 68

Planning for Success: Exploring LLM Long-term Planning Capabilities in Table Understanding
Thi-Nhung Nguyen, Hoang Ngo, Dinh Phung, Thuy-Trang Vu and Dat Quoc Nguyen 81

Derivational Probing: Unveiling the Layer-wise Derivation of Syntactic Structures in Neural Language Models
Taiga Someya, Ryo Yoshida, Hitomi Yanaka and Yohei Oseki 93

Short-circuiting Shortcuts: Mechanistic Investigation of Shortcuts in Text Classification
Leon Eshuijs, Shihan Wang and Antske Fokkens 105

A Linguistically Motivated Analysis of Intonational Phrasing in Text-to-Speech Systems: Revealing Gaps in Syntactic Sensitivity
Charlotte Pouw, Afra Alishahi and Willem Zuidema 126

Experiential Semantic Information and Brain Alignment: Are Multimodal Models Better than Language Models?
Anna Bavaresco and Raquel Fernández 141

What is an Abstract Reasoner"? Revisiting Experiments and Arguments about Large Language Models
Tian Yun, Chen Sun and Ellie Pavlick 156

Do Construction Distributions Shape Formal Language Learning In German BabyLMs?
Bastian Bunzeck, Daniel Duran and Sina Zarrieß 169

Adapting Large Language Models for Movie Domain with Narrative Understanding Tasks
Siqi Shen and Amanmeet Garg 187

From Stories to Statistics: Methodological Biases in LLM-Based Narrative Flow Quantification
Amal Sunny, Advay Gupta, Yashashree Chandak and Vishnu Sreekumar 201

Components of Creativity: Language Model-based Predictors for Clustering and Switching in Verbal Fluency
Sina Zarrieß, Simeon Junker, Judith Sieker and Özge Alacam 216

An Appraisal Theoretic Approach to Modelling Affect Flow in Conversation Corpora
Alok Debnath, Yvette Graham and Owen Conlan 233

Principal Parts Detection for Computational Morphology: Task, Models and Benchmark
Dorin Keshales, Omer Goldman and Reut Tsarfaty 251

Accelerating Large Language Model Pretraining via LFR Pedagogy: Learn, Focus, and Review
Neha Prakriya, Jui-Nan Yen, Cho-Jui Hsieh and Jason Cong 268

What does memory retrieval leave on the table? Modelling the Cost of Semi-Compositionality with MINERVA2 and sBERT
Sydelle De Souza, Ivan Vegner, Francis Mollica and Leonidas A. A. Doumas 291

Polarity inversion operators in PLM
David Kletz, Pascal Amsili and Marie Candito 312

Dynamic Epistemic Friction in Dialogue
Timothy Obiso, Kenneth Lai, Abhijnan Nath, Nikhil Krishnaswamy and James Pustejovsky . 323

A Three-Tier LLM Framework for Forecasting Student Engagement from Qualitative Longitudinal Data
Ahatsham Hayat, Helen Martinez, Bilal Khan and Mohammad Rashedul Hasan 334

Bridging the Socioeconomic Gap in Education: A Hybrid AI and Human Annotation Approach
Nahed Abdelgaber, Labiba Jahan, Arham Vinit Doshi, Rishi Suri, Hamza Reza Pavel and Jia Zhang 348

Construction Identification and Disambiguation Using BERT: A Case Study of NPN
Wesley Scivetti and Nathan Schneider 365

Evidence of Generative Syntax in LLMs
Mary Kennedy 377

Timestep Embeddings Trigger Collapse in Diffusion Text Generation
Ryota Nosaka and Takuya Matsuzaki 397

Investigating Psychometric Predictive Power of Syntactic Attention
Ryo Yoshida, Yushi Sugimoto and Yohei Oseki 407

A Continuous Approach to Metaphorically Motivated Regular Polysemy in Language Models
Anna Temerko, Marcos Garcia and Pablo Gamallo 419

Is Incremental Structure Prediction Process Universal across Languages?: Revisiting Parsing Strategy through Speculation
Taiga Ishii and Yusuke Miyao 437

Lost in Variation? Evaluating NLI Performance in Basque and Spanish Geographical Variants
Jaione Bengoetxea, Itziar Gonzalez-Dios and Rodrigo Agerri 452

Compositionality and Event Retrieval in Complement Coercion: A Study of Language Models in a Low-resource Setting
Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci and Giosuè Baggio 469

DLU: Dictionary Look-Up Data and Prediction
David Strohmaier, Gladys Tyen, Hongyi gu, Diane Nicholls, Zheng Yuan and Paula Buttery . 481

IPA CHILDES & G2P+: Feature-Rich Resources for Cross-Lingual Phonology and Phonemic Language Modeling
Zebulon Goriely and Paula Buttery 502

BabyLM's First Words: Word Segmentation as a Phonological Probing Task
Zebulon Goriely and Paula Buttery 522

GCG-Based Artificial Languages for Evaluating Inductive Biases of Neural Language Models
Nadine El-Naggar, Tatsuki Kuribayashi and Ted Briscoe 540

Beyond Accuracy: Revisiting Out-of-Distribution Generalization in NLI Models
Zahra Delbari and Mohammad Taher Pilehvar 557

Spatial relation marking across languages: extraction, evaluation, analysis
Barend Beekhuizen 571

Human-likeness of LLMs in the Mental Lexicon
Bei Xiao, Xufeng Duan, David A. Haslett and Zhenguang Cai 586

Vorm: Translations and a constrained hypothesis space support unsupervised morphological segmentation across languages
Barend Beekhuizen 602

Do large language models solve verbal analogies like children do?
Tamar Johnson, Mathilde ter Veen, Rochelle Choenni, Han van der Maas, Ekaterina Shutova and Claire E Stevenson 627

Program

Thursday, July 31, 2025

09:20 - 09:30 *Opening Remarks*

09:30 - 10:30 *Invited talk: Raquel Fernández*

10:30 - 11:00 *Coffee break*

11:00 - 12:20 *Oral session 1: Phonology, Morphology, and Syntax*

A Linguistically Motivated Analysis of Intonational Phrasing in Text-to-Speech Systems: Revealing Gaps in Syntactic Sensitivity
Charlotte Pouw, Afra Alishahi and Willem Zuidema

Principal Parts Detection for Computational Morphology: Task, Models and Benchmark
Dorin Keshales, Omer Goldman and Reut Tsarfaty

Is Incremental Structure Prediction Process Universal across Languages?: Revisiting Parsing Strategy through Speculation
Taiga Ishii and Yusuke Miyao

GCG-Based Artificial Languages for Evaluating Inductive Biases of Neural Language Models
Nadine El-Naggar, Tatsuki Kuribayashi and Ted Briscoe

12:20 - 14:00 *Lunch break*

14:00 - 15:30 *Poster session 1*

HKCanto-Eval: A Benchmark for Evaluating Cantonese Language Understanding and Cultural Comprehension in LLMs
Tsz Chung Cheng, Chung Shing Cheng, Chaak-ming Lau, Eugene Lam, Wong Chun Yat, Hoi On Yu and Cheuk Hei Chong

LawToken: a single token worth more than its constituents
Yu-Hsiang Tseng, Hsin-Yu Chou and Shu-Kai Hsieh

Interpersonal Memory Matters: A New Task for Proactive Dialogue Utilizing Conversational History
Bowen Wu, Wenqing Wang, Lihaoran Lihaoran, Yunhan Deng, Ying Li, Jingsong Yu and Baoxun Wang

WinoWhat: A Parallel Corpus of Paraphrased WinoGrande Sentences with Common Sense Categorization
Ine Gevers, Victor De Marez, Luna De Bruyne and Walter Daelemans

Planning for Success: Exploring LLM Long-term Planning Capabilities in Table Understanding
Thi-Nhung Nguyen, Hoang Ngo, Dinh Phung, Thuy-Trang Vu and Dat Quoc Nguyen

Short-circuiting Shortcuts: Mechanistic Investigation of Shortcuts in Text Classification
Leon Eshuijs, Shihan Wang and Antske Fokkens

Do Construction Distributions Shape Formal Language Learning In German BabyLMs?
Bastian Bunzeck, Daniel Duran and Sina Zarrieß

Adapting Large Language Models for Movie Domain with Narrative Understanding Tasks
Siqi Shen and Amanmeet Garg

What does memory retrieval leave on the table? Modelling the Cost of Semi-Compositionality with MINERVA2 and sBERT
Sydelle De Souza, Ivan Vegner, Francis Mollica and Leonidas A. A. Doumas

Polarity inversion operators in PLM
David Kletz, Pascal Amsili and Marie Candito

Dynamic Epistemic Friction in Dialogue
Timothy Obiso, Kenneth Lai, Abhijnan Nath, Nikhil Krishnaswamy and James Pustejovsky

A Three-Tier LLM Framework for Forecasting Student Engagement from Qualitative Longitudinal Data
Ahatsham Hayat, Helen Martinez, Bilal Khan and Mohammad Rashedul Hasan

Bridging the Socioeconomic Gap in Education: A Hybrid AI and Human Annotation Approach
Nahed Abdelgaber, Labiba Jahan, Arham Vinit Doshi, Rishi Suri, Hamza Reza Pavel and Jia Zhang

Construction Identification and Disambiguation Using BERT: A Case Study of NPN
Wesley Scivetti and Nathan Schneider

Thursday, July 31, 2025 (continued)

Timestep Embeddings Trigger Collapse in Diffusion Text Generation
Ryota Nosaka and Takuya Matsuzaki

Lost in Variation? Evaluating NLI Performance in Basque and Spanish Geographical Variants
Jaione Bengoetxea, Itziar Gonzalez-Dios and Rodrigo Agerri

Compositionality and Event Retrieval in Complement Coercion: A Study of Language Models in a Low-resource Setting
Matteo Radaelli, Emmanuele Chersoni, Alessandro Lenci and Giosuè Baggio

DLU: Dictionary Look-Up Data and Prediction
David Strohmaier, Gladys Tyen, Hongyi gu, Diane Nicholls, Zheng Yuan and Paula Buttery

IPA CHILDES & G2P+: Feature-Rich Resources for Cross-Lingual Phonology and Phonemic Language Modeling
Zebulon Goriely and Paula Buttery

BabyLM's First Words: Word Segmentation as a Phonological Probing Task
Zebulon Goriely and Paula Buttery

Beyond Accuracy: Revisiting Out-of-Distribution Generalization in NLI Models
Zahra Delbari and Mohammad Taher Pilehvar

Spatial relation marking across languages: extraction, evaluation, analysis
Barend Beekhuizen

Human-likeness of LLMs in the Mental Lexicon
Bei Xiao, Xufeng Duan, David A. Haslett and Zhenguang Cai

Vorm: Translations and a constrained hypothesis space support unsupervised morphological segmentation across languages
Barend Beekhuizen

Do large language models solve verbal analogies like children do?
Tamar Johnson, Mathilde ter Veen, Rochelle Choenni, Han van der Maas, Ekaterina Shutova and Claire E Stevenson

Thursday, July 31, 2025 (continued)

15:30 - 16:00 *Coffee break*

16:00 - 17:20 *Oral session 2: Semantics, Affect, and Multimodality*

Experiential Semantic Information and Brain Alignment: Are Multimodal Models Better than Language Models?
Anna Bavaresco and Raquel Fernández

A Continuous Approach to Metaphorically Motivated Regular Polysemy in Language Models
Anna Temerko, Marcos Garcia and Pablo Gamallo

What is an Abstract Reasoner"? Revisiting Experiments and Arguments about Large Language Models
Tian Yun, Chen Sun and Ellie Pavlick

An Appraisal Theoretic Approach to Modelling Affect Flow in Conversation Corpora
Alok Debnath, Yvette Graham and Owen Conlan

17:20 - 18:00 *Discussion session*

Friday, August 1, 2025

09:30 - 10:30 *Invited talk: Jean-Rémi King*

10:30 - 11:00 *Coffee break*

11:00 - 12:20 *Oral session 3: LLMs, Linguistic Structure, and Psycholinguistics*

Evidence of Generative Syntax in LLMs
Mary Kennedy

Derivational Probing: Unveiling the Layer-wise Derivation of Syntactic Structures in Neural Language Models
Taiga Someya, Ryo Yoshida, Hitomi Yanaka and Yohei Oseki

Investigating Psychometric Predictive Power of Syntactic Attention
Ryo Yoshida, Yushi Sugimoto and Yohei Oseki

Components of Creativity: Language Model-based Predictors for Clustering and Switching in Verbal Fluency
Sina Zarrieß, Simeon Junker, Judith Sieker and Özge Alacam

12:20 - 14:00 *Lunch break*

14:00 - 15:30 *Poster session 2 (Poster Session 2 will feature the same poster presentations as Poster Session 1)*

15:30 - 16:00 *Coffee break*

16:00 - 17:00 *Oral session 4: Best Paper Session*

Quasi-symbolic Semantic Geometry over Transformer-based Variational AutoEncoder
Yingji Zhang, Danilo Carvalho and Andre Freitas

Accelerating Large Language Model Pretraining via LFR Pedagogy: Learn, Focus, and Review
Neha Prakriya, Jui-Nan Yen, Cho-Jui Hsieh and Jason Cong

Friday, August 1, 2025 (continued)

	From Stories to Statistics: Methodological Biases in LLM-Based Narrative Flow Quantification Amal Sunny, Advay Gupta, Yashashree Chandak and Vishnu Sreekumar
17:00 - 17:15	*Closing*

HKCanto-Eval: A Benchmark for Evaluating Cantonese Language Understanding and Cultural Comprehension in LLMs

Tsz Chung Cheng[1], **Chung Shing Cheng**[2], **Chaak Ming Lau**[3],
Eugene Tin-Ho Lam[4], **Chun Yat Wong**[5], **Hoi On Yu**[6], **Cheuk Hei Chong**[7,8]

[1] Kyushu University, [2] hon9kon9ize, [3] The Education University of Hong Kong,
[4] Independent Researcher, [5] The University of Hong Kong,
[6] Independent Researcher, [7] Votee AI, [8] Beever AI

Correspondence: Tsz Chung Cheng: jed.cheng@mag.ed.kyushu-u.ac.jp,
Chung Shing Cheng joseph.cheng@hon9kon9ize.com Chaak Ming Lau: lchaakming@eduhk.hk

Abstract

The ability of language models to comprehend and interact in diverse linguistic and cultural landscapes is crucial. The Cantonese language used in Hong Kong presents unique challenges for natural language processing due to its rich cultural nuances and lack of dedicated evaluation datasets. The HKCanto-Eval benchmark addresses this gap by evaluating the performance of large language models (LLMs) on Cantonese language understanding tasks, extending to English and Written Chinese for cross-lingual evaluation. HKCanto-Eval integrates cultural and linguistic nuances intrinsic to Hong Kong, providing a robust framework for assessing language models in realistic scenarios. Additionally, the benchmark includes questions designed to tap into the underlying linguistic metaknowledge of the models. Our findings indicate that while proprietary models generally outperform open-weight models, significant limitations remain in handling Cantonese-specific linguistic and cultural knowledge, highlighting the need for more targeted training data and evaluation methods. The code can be accessed at https://github.com/hon9kon9ize/hkeval2025

1 Introduction

Recent advancements in large language models (LLMs) such as GPT-4, Gemini, and various open-weight models have demonstrated remarkable capabilities in natural language understanding across multiple languages (Xu et al., 2024). However, the performance of most models significantly declines when applied to languages other than English, yielding particularly poor outcomes for low-resource languages (LRLs). These languages are under-represented lingua francas that play a crucial role in certain communities, and it is imperative to improve multilingual support for LRLs by creating benchmarks to guide the future development of multilingual LLMs. Since they are poorly supported due to the lack of training data, if there is a close language with more resources, this problem can potentially be mitigated through few-shot learning. A notable example of this strategy is the use of Bahasa Indonesian to handle regional languages in Indonesia (Aji et al., 2022; Winata et al., 2022). This strategy aligns with the spirit of language sustainability and AI support for marginalised communities (Du et al., 2020), which is also applicable to Cantonese.

This paper investigates the status of LLM support for Cantonese (ISO 639-3: *yue*), a member of the *Sinitic* ("Chinese") branch of the Sino-Tibetan language family, and a distinct variety unintelligible to users of Mandarin, the standard variety of Chinese used in Mainland China (Pǔtōnghuà) and Taiwan (Guóyǔ). Cantonese, spoken by over 85 million people according to *Ethnologue* (Eberhard et al., 2024), serves as the most common and *de facto* official language of Hong Kong and Macau, and is also widely used in parts of Guangdong, Guangxi, Malaysia, and Singapore. Additionally, it is used as a diasporic language in countries such as Canada (Sachdevl et al., 1987), the United States (Leung and Uchikoshi, 2012), Australia (Zhang et al., 2023), and the United Kingdom (Bauer, 2016; Tsapali and Wong, 2023). Despite its widespread use, Cantonese is still considered a low-resource language (Xiang et al., 2024) due to the lack of quality written resources. This scarcity results from a "diglossia" that requires Written Chinese (which resembles Mandarin) to be used in formal settings[1], and a longstanding, ideologically-driven stigmatisation of Cantonese as an informal/vulgar language (Lau, 2024), further confines written Cantonese to informal con-

---

[1] Even in Mandarin-like Written Chinese, there are persistent lexical differences with other regions due to vastly different governmental, legal and education systems. For instance, the word "taxi" is rendered as "出租車" in mainland China, "計程車" in Taiwan, and "的士" in Hong Kong and Macau.

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 1–11
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

texts like social media and texting.

Cantonese is partially supported by certain LLMs, with models like GPT-4 and Gemini capable of comprehending and responding in Cantonese (Fu et al., 2024; Hong et al., 2024; Jiang et al., 2024). There are models dedicated to better supporting Chinese languages and dialects: The Hong Kong government is developing an internal tool based on locally developed LLMs for administrative use (Yiu, 2024); SenseTime released SenseChat (Cantonese), a model trained on 6 billion tokens of Hong Kong-specific data (SenseTime, 2024). However, the current support level is mostly contributed to by small pockets of Cantonese presented in the sheer volume of Written Chinese training data. There have been comparisons between Chinese and Western models on how well languages spoken in China are handled (Wen-Yi et al., 2025), showing that Chinese models outperformed Western ones on Mandarin, but the same cannot be said for Cantonese or other languages in China. The following section outlines how current benchmarking studies have yet to provide a comprehensive evaluation for Cantonese and Hong Kong-related tasks that tap into the in-depth representation of underlying aspects of the language, which we believe is the prerequisite for accurate comprehension in uncommon scenarios.

2 Related Benchmarks

The development of LLMs has spurred significant research into evaluating their performance and comparing their capabilities to human reasoning across general and domain-specific tasks. A prominent benchmark in this area is the MMLU dataset (Hendrycks et al., 2020), which comprises 57 tasks ranging from elementary to university-level multiple-choice questions. Despite its widespread use, MMLU has been criticised for containing flawed questions and answers (Gema et al., 2024; Gupta et al., 2024). To address these shortcomings, alternative benchmarks such as BIG-Bench (Srivastava et al., 2022), MMLU-Pro (Taghanaki et al., 2024), and MMLU-Pro+ (Wang et al., 2024) have been introduced, aiming to improve accuracy while presenting more diverse and challenging questions.

In addition to comprehensive benchmarks, researchers have developed domain-specific, expert-curated datasets to evaluate the reasoning capabilities of LLMs in specialised fields such as programming (HumanEval (Chen et al., 2021); NL2Code (Zan et al., 2022)) and mathematical reasoning (GSM8K (Cobbe et al., 2021); MATH (Hendrycks et al., 2021); MATH 401 (Yuan et al., 2023); Omni-MATH (Gao et al., 2024)).

Although most existing LLM benchmarks focus on English-language tasks, culturally-aware datasets integrating machine-translated questions, native datasets, and exam questions have been developed in other languages, including Arabic (Koto et al., 2024), Basque (Etxaniz et al., 2024a,b), Spanish (Plaza et al., 2024), Indic languages (Verma et al., 2024), and Korean (Son et al., 2024). Similar benchmarks have been published for Chinese, such as CMMLU (Li et al., 2023) and C-Eval (Huang et al., 2024) that gathered questions from various academic and professional exams in mainland China, and TMLU (Chen et al., 2024) and TMMLU+ (Tam et al., 2024) that evaluate knowledge in Traditional Chinese in the context of Taiwan.

These benchmarks are not applicable to the Hong Kong context due to the aforementioned diglossia and regional lexical differences. Recently, Jiang et al. (2024) introduced a Cantonese evaluation benchmark that combines four datasets translated from other languages (ARC, GSM8K, CMMLU, and Truthful-QA)[2], resulting in a dataset that is heavily biased towards American culture (16.9% entries in the Truthful-QA dataset reference the United States) or mainland Chinese exams (CMMLU) (see Appendix A).

3 Methodology

HKCanto-Eval introduces a specialised benchmark to address the lack of systematic tests for evaluating the Cantonese capabilities and Hong Kong knowledge of an LLM in these aspects: (1) **Language Proficiency**, the capability in an accurate and nuanced understanding of Cantonese and local-flavoured Written Chinese, as well as generating fluent, idiomatic, genre-appropriate Cantonese text in question and answering, translation, and summarisation tasks; (2) **Cultural Knowledge**, in-depth knowledge about not only general historical and geographical facts related to Hong Kong, but also everyday practices, local customs, beliefs and values, and cultural references from

---

[2] It also contains a translation evaluation component for English-Cantonese and Simplified-to-Traditional Chinese translations, but its data sources and evaluation methods are not fully transparent.

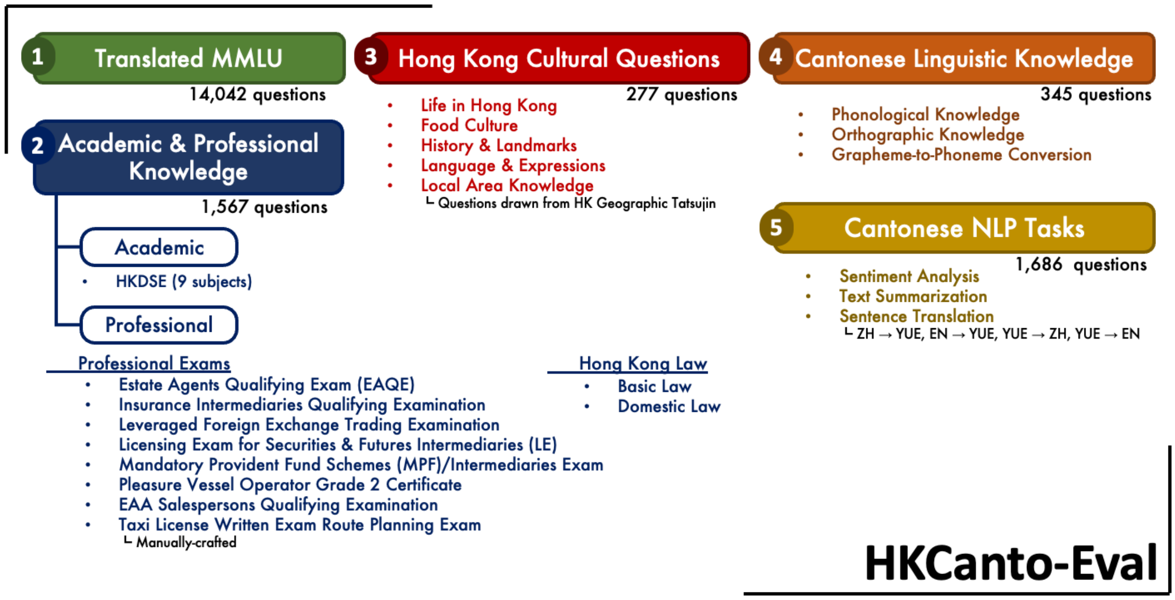

1 Translated MMLU
14,042 questions
2 Academic & Professional Knowledge
1,567 questions
Academic
HKDSE (9 subjects)
Professional
Professional Exams
Estate Agents Qualifying Exam (EAQE)
Insurance Intermediaries Qualifying Examination
Leveraged Foreign Exchange Trading Examination
Licensing Exam for Securities & Futures Intermediaries (LE)
Mandatory Provident Fund Schemes (MPF)/Intermediaries Exam
Pleasure Vessel Operator Grade 2 Certificate
EAA Salespersons Qualifying Examination
Taxi License Written Exam Route Planning Exam
└ Manually-crafted
Hong Kong Law
Basic Law
Domestic Law
3 Hong Kong Cultural Questions
277 questions
Life in Hong Kong
Food Culture
History & Landmarks
Language & Expressions
Local Area Knowledge
└ Questions drawn from HK Geographic Tatsujin
4 Cantonese Linguistic Knowledge
345 questions
Phonological Knowledge
Orthographic Knowledge
Grapheme-to-Phoneme Conversion
5 Cantonese NLP Tasks
1,686 questions
Sentiment Analysis
Text Summarization
Sentence Translation
└ ZH → YUE, EN → YUE, YUE → ZH, YUE → EN
HKCanto-Eval

Figure 1: Diagram showing the tasks of the HKCanto-Eval Benchmark

movies, music, literature, and internet culture; (3) **Reasoning and Problem-Solving**, reasoning and problem-solving skills within a Cantonese and/or Hong Kong-based context, including reasoning about the sound and written forms of the language.

These aspects are incorporated into the five datasets outlined below.

3.1 Translated MMLU Dataset

The first dataset comprises 14,042 questions from the original MMLU dataset in English (Hendrycks et al., 2020) and their Cantonese translation[3]. This allows us to compare how LLMs perform when handling knowledge in a wide range of subjects in Cantonese rather than in English (See Appendix B).

3.2 Academic and Professional Dataset

The Academic and Professional Dataset is a set of multiple-choice questions curated to measure LLMs' reasoning and problem-solving abilities in domain-specific knowledge. The dataset contains multiple-choice questions from 3 sub-categories: (1) **Academic**: Questions sourced from Hong Kong Diploma of Secondary Education (HKDSE), a territory-wide high-school graduate-level exam; extracted and manually corrected from scanned PDFs and are believed to have never appeared online in a plain-text form; (2) **Professional**: Questions from seven professional qualification exams, extracted from text PDF files found on the corresponding official sites (in which the model answers were not on the same page as the questions, avoiding data contamination concerns), and an additional set of Taxi Licensing Exam Styled Route Planning questions on Hong Kong roads and geographical features; (3) **Law**: Questions about law in Hong Kong across 15 categories sourced from the Internet, and an additional subset of the Basic Law edited by the authors.

All questions are in Written Chinese (in the Traditional script). We also included an English version if it is available. The details of this dataset can be found in Appendix C.

3.3 Hong Kong Cultural Questions Dataset

This dataset contains 277 manually crafted questions divided into five categories that capture cultural knowledge common to people who have lived or grown up in Hong Kong, that are often not learned in schools. The categories are **Food Culture**, **History and Landmarks**, **Language and Expressions**, **Life in Hong Kong** and **Local Area Knowledge**. The questions were collected in a way to capture knowledge from all walks of life. 244 questions were developed by the authors and volunteers for the first four categories, and the last category comes from an online quiz. Questions were created so that they were non-trivial and at

---

[3] The translation was done by the Google Gemini 1.5 Flash API, which offers a balance between top performance and cost as one would find in the later section. To address concerns regarding the accuracy of LLM translation, we have selected 4 questions from each category for human checking. 202 out of 228 sentences were judged to be good by the raters.

the same time not too obscure, and have been verified by all the authors. Details can be found in Appendix D.

3.4 Linguistic Knowledge Dataset

This is an assessment of the linguistic knowledge represented in the models, inspired by the approach of PhonologyBench (Suvarna et al., 2024) for English. To our knowledge, this innovative approach has never been incorporated into existing Cantonese or Chinese benchmarks in general.

3.4.1 Phonological Knowledge

The dataset contains 100 questions that evaluate phonological knowledge about characters and words of an LLM, including the judgment of homophones and rhyming and other non-trivial reasoning tasks based on word pronunciation. These are particularly important in the Cantonese context, as the writing system does not provide reliable cues about the pronunciation of words, and Cantonese materials are not accompanied by sound transcription. This knowledge needs to be present in the training data for tasks that require sound-related operations or reasoning (See Appendix E.1).

3.4.2 Orthographic Knowledge

The Orthographic Knowledge Dataset evaluates the character meta-knowledge of an LLM. Cantonese users from Hong Kong need to know around 4,000 characters by the age of 12 and will have built sound knowledge about the representation of the characters. This subset contains 100 questions about the strokes, structure, arrangement, and radical and constituent components of common characters. Cantonese uses the Traditional Chinese script (ISO 15924: Hant) in Hong Kong and Macau, and the script is also used in Taiwan. There could be influence from Mandarin data or Taiwan usage not shared by Cantonese. It is also expected that certain models may produce incorrect answers due to the over-reliance on simplified Chinese data (See Appendix E.2).

3.4.3 Grapheme-to-Phoneme (G2P) Conversion

This dataset addresses the task of converting a string of written text represented in Traditional Chinese characters into Jyutping, a widely adopted romanisation standard of Cantonese[4]. This is similar to typical G2P tasks except that Jyutping is used instead of the International Phonetic Alphabet (IPA) as the output. G2P functionalities have been implemented by PyCantonese (Lee et al., 2022), a Cantonese NLP package, Hambaanglaang Converter[5] and Visual Fonts[6]. As the task is non-deterministic, rule-based conversions are bound to be unreliable (although Visual Fonts have achieved very high accuracy now). There is also no reliable non-rule-based G2P system to our best knowledge. This part of the dataset contains 150 pairs of Character-Jyutping sentences from both Standard Written Chinese and Cantonese and in a range of formality levels, manually checked by professional linguists from the Linguistic Society of Hong Kong, the organisation that established and maintains the Jyutping system. The score calculation method is discussed in Appendix E.3.

3.5 NLP Tasks Dataset

Multiple-choice questions offer a structured approach to assess LLM factual knowledge and reasoning, but they are insufficient for evaluating real-world language understanding and generation. Open-ended tasks, including translation and summarisation, were incorporated.

A translation dataset comprising 20 Cantonese sentences with complex linguistic nuances was created, with each sentence manually translated into English and written Chinese (resulting in 4 translation pairs per sentence) (See Appendix F). For summarisation, 10 Cantonese articles and 10 TED talk subtitles were used. The importance of transcription-based summarisation, reflecting Cantonese's prevalence in oral communication, is emphasised by the inclusion of TED talks (See Appendix G).

Performance on traditional NLP tasks like sentiment analysis was also evaluated. Leveraging the OpenRice dataset (toastynews, 2020) (restaurant reviews categorised as positive, neutral, or negative), 1200 reviews (avg. 309 characters) with a balanced sentiment distribution were included. Additionally, a new dataset of 399 Facebook comments (avg. 24 characters), labelled by paid interns, was created (See Appendix H).

3.6 Evaluation Method

The evaluation process of multiple-choice questions follows the standard 5-shot evaluation procedures in MMLU formulation. However, for the

---

[4] https://lshk.org/jyutping-scheme

[5] https://test.hambaanglaang.hk

[6] https://visual-fonts.com

Hong Kong Cultural Questions Dataset, a zero-shot evaluation was also conducted to emulate actual usage. The translated MMLU dataset used the same system prompt as the original MMLU dataset. For other multiple-choice questions, a short sentence with the name of the exam or question subcategory is added.

For the G2P dataset, character error rates (CER) and Levenshtein distance were both used to calculate the discrepancy between the model output and the gold standard in a five-shot evaluation. The summarisation tasks were evaluated without any example to avoid exceeding the context length of any model, while zero and three-shot evaluations were carried out for the translation task.

The outputs of both translation and summarisation evaluation were evaluated and graded by paid undergraduate students and teaching assistants. The rubric can be found in Appendix F and G. As technology improves, future LLMs can perform the task to offer scalability. Nonetheless, the results from this human evaluation will be useful for verifying the validity and consistency of LLM-as-a-judge in the future.

3.7 Model Selection

13 model families were selected for evaluation. Proprietary models including OpenAI GPT4o (Hurst et al., 2024) and GPT4-mini (OpenAI, 2024), Google Gemini 1.5 Flash and Gemini 1.5 Pro (Gemini Team et al., 2024) and Anthropic Claude 3.5 Sonnet (Anthropic, 2024) were selected for their reported superior performance across different languages.

Three proprietary models from Chinese companies, including Doubao Pro from ByteDance (Doubao, 2024), Erne 4.0 from Baidu (Baidu Inc., 2023) and SenseChat (Cantonese) from SenseTime (SenseTime, 2024), were also incorporated. All proprietary models were accessed through their API, except SenseChat, which was accessed via the web interface due to a failure to get verified to use their API.

Popular multilingual open-weight models including Aya 23 8B (Aryabumi et al., 2024), Gemma 2 2B, 9B and 27B (Gemma Team et al., 2024), Llama 3.1 8B, 70B and 405B (Dubey et al., 2024), and Mistral Nemo Instruct 2407 12B (Mistral, 2024) were included to assess their cross-lingual ability. The collection also included two open-weight multilingual models from Chinese companies, Yi 1.5 6B, 9B and 34B (Young et al., 2024) and Qwen2 7B and 72B (Yang et al., 2024). In addition, CantoneseLLM (CLLM) v0.5 6B and 34B[7] are two of the few open-weight models trained specifically on Cantonese data. They were trained by fine-tuning Yi 1.5 6B and 34B models with around 400 million tokens of Hong Kong-related content. Open-weight instructions fine-tuned models smaller than 70B parameters were evaluated using Nvidia H100 GPUs. The 70B and 405B models were evaluated using the API of SiliconFlow[8].

4 Results

4.1 MMLU

Table 1 shows the results of the multiple-choice questions. Proprietary models and open-weight models like Llama 3.1 70B, 405B, and Qwen 2 72B performed well in MMLU, but experienced an average of 7.46 percentage point drop when questions were in Cantonese. Considering potential errors from machine translations, this is evidence of *Cantonese reasoning and problem-solving ability*.

4.2 Academic and Professional Questions

The results of this dataset showed expected problem-solving abilities across models in different subject areas, in particular, general weaknesses in handling secondary school-level mathematics and strong performance in legal questions. Proprietary models generally performed better than open-weight models. The sub-scores in the individual tasks show that most models struggled with academic questions that were never posted online. It is worth noting that some open-weight models (e.g. CLLM v0.5 34B and Qwen2 72B) outperformed most models, and we can conduct further investigation on what additional training data was used to achieve this performance. Written Chinese yielded better overall results, and this is attributed to the Law dataset, which only came in Chinese. Discounting this set, Written Chinese caused a slight drop in performance. This indicates that *multi-lingual open-weight LLMs showed cross-lingual capabilities*, maintaining similar performance across both languages.

---

[7] https://huggingface.co/hon9kon9ize/CantoneseLLMChat-v0.5

[8] https://siliconflow.cn

	MMLU		Academic & Professional		Cultural		Average	
Model	EN	YUE	EN	ZH	0-shot	5-shot	EN	ZH/YUE
Claude 3.5 Sonnet	**85.0%**	**81.5%**	75.1%	75.2%	71.7%	75.0%	80.1%	75.8%
Doubao Pro	79.8%	74.2%	60.8%	70.5%	70.7%	75.0%	70.3%	72.6%
Ernie 4.0	81.0%	75.2%	70.4%	72.4%	68.2%	75.2%	75.7%	72.8%
Gemini 1.5 Flash	79.0%	73.1%	67.4%	68.3%	61.0%	64.0%	73.2%	66.6%
Gemini 1.5 Pro	83.2%	77.6%	71.0%	71.7%	74.0%	73.8%	77.1%	74.3%
GPT4o	84.8%	80.3%	**77.6%**	75.3%	**77.5%**	77.2%	**81.2%**	**77.6%**
GPT4o-mini	76.7%	69.4%	62.0%	65.6%	55.6%	60.6%	69.4%	62.8%
SenseChat	78.7%	70.1%	73.6%	**75.6%**	67.4%	**77.4%**	76.1%	68.8%
Aya 23 8B	56.6%	47.1%	44.8%	49.0%	39.5%	37.7%	50.7%	43.3%
CLLM v0.5 6B	58.6%	51.7%	50.9%	53.5%	52.0%	56.1%	54.7%	53.3%
CLLM v0.5 34B	75.9%	69.9%	66.8%	69.9%	72.5%	76.7%	71.3%	72.3%
Yi 1.5 6B	64.1%	54.0%	53.7%	58.3%	47.7%	50.7%	58.9%	52.7%
Yi 1.5 9B	70.9%	60.8%	59.2%	63.3%	48.7%	57.3%	65.0%	57.5%
Yi 1.5 34B	76.1%	68.5%	63.7%	68.7%	67.7%	72.9%	69.9%	69.5%
Gemma 2 2B	58.5%	46.5%	45.3%	48.5%	33.3%	35.2%	51.9%	40.9%
Gemma 2 9B	73.4%	64.3%	63.6%	64.0%	49.1%	51.6%	68.5%	57.3%
Gemma 2 27B	76.4%	68.4%	65.1%	68.1%	57.1%	60.9%	70.7%	63.6%
Llama 3.1 8B	69.0%	56.4%	51.4%	57.1%	45.6%	52.7%	60.2%	52.9%
Llama 3.1 70B	80.3%	74.9%	68.2%	70.0%	63.0%	64.4%	74.2%	68.1%
Llama 3.1 405B	**84.5%**	**78.4%**	70.9%	74.2%	67.9%	69.9%	77.7%	72.6%
Mistral Nemo 12B	68.8%	58.4%	54.6%	58.0%	40.1%	42.7%	61.7%	49.8%
Qwen2 7B	71.2%	64.8%	60.7%	65.4%	53.6%	54.8%	66.0%	59.6%
Qwen2 72B	82.9%	78.3%	**74.7%**	**76.3%**	**72.9%**	**77.7%**	**78.8%**	**76.3%**
Random	25.0%	25.5%	22.9%	24.6%	29.8%	28.1%	23.9%	27.0%

Table 1: Model performance on MMLU, Academic and Professional, and Cultural questions. Note that SenseChat refused to answer one subset of questions in Cultural Question 5-shot evaluation.

4.3 Hong Kong Cultural Questions

Proprietary models and Qwen 2 72B showed a good understanding of Hong Kong cultural knowledge, yet none of the models performed well across the subcategories. Looking into the sub-scores, models occasionally matched humans in most subtests (e.g. Food Culture and Life in HK). However, when inspecting the results, good performance by percentage *only reflects the size of existing Hong Kong knowledge represented in Wikipedia entries.* For example, only two models (Yi 1.5 6B and Qwen2 72B) correctly answered the origin of Demae Itcho noodles sold in Hong Kong, while 94% of humans did. The results for Language & Expressions also show that *most models did not have a nuanced understanding of Cantonese*. Compared to human performance at 85.8%, SenseChat scored the highest point out of all models in 5-shot (79.6%), but its performance dropped significantly in zero-shot (61.4%). In zero-shot evaluation, CLLM v0.5 34B delivered the best performance at 77.3%. Furthermore, model size affects the performance of geospatial tasks, with open-source models in the 6-9B parameter range achieving only about 50% of larger models' performance on Local Area Knowledge (e.g. Yi 1.5 34B 67.9%, 9B 35.7%). The overall results of this dataset suggest that Hong Kong cultural knowledge is underrepresented in LLM training. See Appendix C for details.

4.4 Linguistic and NLP Tasks

These two groups of tasks reveal the representation of Cantonese phonological, orthographic, lexical and grammatical knowledge in existing models. The overall results (Table 2) show a consistent trend where proprietary models outperformed open-weight models (but more pronounced in linguistic tasks). GPT-4o led with 76.7% and 89.6% in both *linguistic* and *NLP* tasks. Lower scores are often due to chance-level performance when

	Phonological Knowledge			Orthographic Knowledge			NLP
Model	Homophone	Rhyme	Misc.	Visual Sim.	Canton. Char.	Misc.	Avg.
Claude 3.5 Sonnet	28.0%	64.0%	16.0%	50.0%	76.9%	59.3%	89.2%
Doubao Pro	16.0%	44.0%	16.0%	70.0%	80.8%	48.1%	87.0%
Ernie 4.0	28.0%	60.0%	18.0%	70.0%	80.8%	53.7%	82.7%
Gemini 1.5 Flash	12.0%	20.0%	24.0%	40.0%	73.1%	31.5%	83.2%
Gemini 1.5 Pro	16.0%	40.0%	24.0%	50.0%	**88.5%**	46.3%	87.9%
GPT4o	**56.0%**	**96.0%**	**28.0%**	50.0%	65.4%	**63.0%**	**89.6%**
GPT4o-mini	20.0%	60.0%	20.0%	30.0%	57.7%	40.7%	86.1%
SenseChat	16.0%	36.0%	22.0%	**75.0%**	76.9%	42.6%	78.8%
Aya 23 8B	12.0%	40.0%	14.0%	15.0%	19.2%	31.5%	70.1%
CLLM v0.5 6B	24.0%	8.0%	18.0%	20.0%	50.0%	27.8%	71.9%
CLLM v0.5 34B	28.0%	28.0%	14.0%	35.0%	76.9%	37.0%	73.3%
Yi 1.5 6B	28.0%	12.0%	12.0%	10.0%	50.0%	20.4%	56.6%
Yi 1.5 9B	**36.0%**	40.0%	24.0%	30.0%	57.7%	18.5%	72.2%
Yi 1.5 34B	16.0%	32.0%	**26.0%**	30.0%	61.5%	33.3%	82.9%
Gemma 2 2B	8.0%	24.0%	18.0%	25.0%	53.8%	22.2%	73.4%
Gemma 2 9B	20.0%	28.0%	24.0%	25.0%	50.0%	33.3%	85.0%
Gemma 2 27B	20.0%	12.0%	16.0%	25.0%	65.4%	24.1%	83.2%
Llama 3.1 8B	12.0%	16.0%	18.0%	25.0%	42.3%	38.9%	60.3%
Llama 3.1 70B	28.0%	40.0%	12.0%	30.0%	61.5%	35.2%	**84.5%**
Llama 3.1 405B	20.0%	**44.0%**	18.0%	35.0%	65.4%	**50.0%**	64.4%
Mistral Nemo 12B	12.0%	28.0%	10.0%	25.0%	23.1%	37.0%	68.8%
Qwen2 7B	8.0%	40.0%	12.0%	35.0%	46.2%	33.3%	66.8%
Qwen2 72B	12.0%	28.0%	16.0%	**50.0%**	**76.9%**	48.1%	83.5%
Random/Control	16.0%	28.0%	24.0%	30.0%	11.5%	27.8%	76.8%

Table 2: Model performance on Linguistic Knowledge Dataset multiple-choice questions and NLP tasks. The bottom row indicates the expected correctness from random selection for the Phonological and Orthographic Knowledge tasks. For NLP, the reported figure is the average evaluation of professionally prepared translations for translation tasks serving as a control.

knowledge is absent, or below chance-level due to influence from Mandarin. Here are the key findings and observations:

Most LLMs understand Cantonese fine. Most models performed well in Sentiment Analysis (GPT4o 79.7%, Llama 3.1 405B 78.8%), Translation (3-shot: GPT4o 98.3%, Qwen2 72B 96.6%), and Summarisation (Claude 3.5 Sonnet 92.7%, Gemma 2 9B 91.3%). Models that obtained lower scores are often due to task completion problems, e.g. failure to handle long input and problems with low-frequency/mixed-language tokens.

Proprietary and large open-weight models have good Cantonese lexical knowledge. The performance in translation and sentiment analysis is closely tied to the ability to determine the meaning of Cantonese-specific words that are not found or used differently in Mandarin. Most models also performed well in the Cantoense Character Selection sub-task (Canton. Char. in Table 2) under Orthographic Knowledge. It is noteworthy that despite good performance with proprietary models (73.1% - 88.5%) and some open-weight models (CLLM v0.5 34B and Qwen2 72B, both 76.9%), GPT4o struggled with Cantonese orthography (65.4%).

LLMs in general lack knowledge about Cantonese pronunciation. In the Grapheme-to-Phoneme (G2P) conversion task, all models performed far worse than the rule-based control (Visual Fonts v3.3, CER 0.8%), with the closest being GPT-4o (5.4%) and Claude 3.5 Sonnet (7.9%) as shown in Table 3. The appalling results from all tested language models reveal how linguistic knowledge is seriously under-represented. While it is expected that the G2P tasks will be significantly improved in newer/future models, actual linguistic tasks that involve sounds require more ad-

Model	CER	Levenshtein
Claude 3.5 Sonnet	7.9%	0.018
Doubao Pro	20.9%	0.044
Ernie 4.0	34.4%	0.094
Gemini 1.5 Flash	34.7%	0.083
Gemini 1.5 Pro	15.3%	0.030
GPT4o	**5.4%**	**0.009**
GPT4o-mini	12.0%	0.023
SenseChat	54.4%	0.163
Aya 23 8B	96.6%	0.724
CLLM v0.5 6B	94.1%	0.859
CLLM v0.5 34B	**23.4%**	**0.058**
Yi 1.5 6B	99.0%	0.577
Yi 1.5 9B	97.2%	0.528
Yi 1.5 34B	79.6%	0.837
Gemma 2 2B	97.5%	0.524
Gemma 2 9B	73.0%	0.259
Gemma 2 27B	62.5%	0.201
Llama 3.1 8B	69.9%	0.270
Llama 3.1 70B	31.3%	0.086
Llama 3.1 405B	26.3%	0.074
Mistral Nemo 12B	59.8%	0.201
Qwen2 7B	97.3%	0.466
Qwen2 72B	74.0%	0.268
Rule Based	0.8%	0.001

Table 3: Model performance in the Grapheme-to-Phoneme (G2P) dataset. Scores calculated based on character error rates (CER) and Levenshtein distance. (Lower is better)

vanced knowledge about the language's sound system. Most models struggled with tasks like judging homophone or rhyme pairs in Table 2, with GPT-4o being a notable exception (Homophone: 56.0%; Rhyming: 96.0%). Poor (close to chance level) performance in other models is not only due to the lack of G2P ability, a prerequisite for phonological reasoning, but also due to how Mandarin homophones partially influence this task. This will continue to be challenging for Cantonese due to limited specialised data.

LLMs in general do not have meta-linguistic knowledge represented in Cantonese. Although certain models, especially the Chinese proprietary models, performed well in the visual similarity task (SenseChat 70%, Doubao 70%, Ernie 75%) or orthographic reasoning (GPT4o 63.0%), the knowledge seems to have come from Simplified Chinese, thus their good performance is not transferred to Cantonese-specific items. This seems to be caused by insufficient descriptive knowledge about the structure and properties associated with the individual glyphs.

5 Conclusion

This paper presents HKCanto-Eval, the first comprehensive evaluation benchmark focusing on Hong Kong Cantonese, by comparing the Cantonese language support of 6 proprietary and 7 open-weight model families. Our findings indicate that while these models can understand Cantonese in various contexts, retrieve knowledge about Hong Kong, and address problems written in or about Cantonese to some extent, there are notable limitations. Most models, especially open-weight models in the 6–9B range, lack sufficient linguistic, cultural and professional knowledge in Cantonese and Hong Kong. Performance was particularly poor for questions requiring knowledge not commonly found in major online sources.

One area that we paid close attention to is the presence of metalinguistic knowledge in these models. There is concern that models showed Cantonese proficiency in linguistic and NLP tasks primarily through Mandarin. If their linguistic understanding is based solely on Mandarin, they may perform well on simpler tasks but struggle significantly with "false friends" between languages, as Mandarin knowledge becomes a hindrance. This benchmark introduces a novel perspective, focusing on Cantonese processing abilities beyond superficial slang and expressions. By requiring reasoning about sounds and characters specific to Cantonese, our benchmark provides a fairer judgement that credits models accurately capturing Cantonese phonology and orthography, while exposing those that appear competent in Cantonese but are heavily reliant on Mandarin.

This challenge in processing Cantonese is shared by other low-resource languages. As training data increases, models tend to favour high-resource languages like Mandarin Chinese. The apparent similarity between Cantonese and Written Chinese further affects the ability of even proprietary models to distinguish between these linguistic contexts accurately. Addressing the segregation of regional and linguistic knowledge is crucial for developing culturally and linguistically adaptive LLMs. This issue extends beyond Cantonese to other under-represented language communities.

6 Limitations & Future Directions

The current benchmark exhibits several limitations.

Inaccuracies in machine-translated materials: First, the use of machine translation introduces potential inaccuracies. While Gemini 1.5 Flash balances cost and quality, human-translated questions could provide a more reliable benchmark, albeit at a higher resource cost. The reliance on multiple-choice and text-based questions does not fully capture the capabilities required for practical LLM applications such as code generation and mathematical problem-solving, which demand coherent and contextual text generation. The dataset also lacks multi-modal data like image and audio, which is now supported by proprietary models and should be evaluated.

Biases in topic selection: The newly and manually created questions might contain biases and a lack of scalability and comprehensiveness. The cultural questions, predominantly created by colleagues and relatives of the authors, may introduce bias in cultural references and wordings, leading to an over-representation of certain perspectives while under-representing others, such as traditional practices. Political topics were also specifically excluded, due to political complications, limiting cultural representation. This can also be considered a reasonable compromise since many models (e.g. those from Chinese companies) are configured to censor these topics, and there is a risk that our accounts or IP addresses will be banned before we complete all the benchmarking tasks for this paper.

Lack of Crosslingual Evaluation: English translations for cross-lingual ability evaluation were also not included due to resource limitations. An additional comparison should be added to compare whether the same set of questions will be answered less satisfactorily when presented in English or Standard Written Chinese instead of Cantonese, in line with the evaluation done for Basque (Etxaniz et al., 2024a) and Mongolian and Tibetan (Zhang et al., 2025). We will leave this for future research.

Reliance on human evaluation: Human evaluation, while insightful, is not scalable. Automated and objective evaluation methods, such as LLM-as-a-judge or rule-based approaches, are necessary for efficient evaluation, but this is challenging due to the low-resource nature of Cantonese.

Future directions include developing benchmarks incorporating audio, images, and tables, and addressing the aforementioned limitations to create more comprehensive and representative evaluations.

References

Alham Fikri Aji, Genta Indra Winata, Fajri Koto, Samuel Cahyawijaya, Ade Romadhony, Rahmad Mahendra, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Timothy Baldwin, Jey Han Lau, and Sebastian Ruder. 2022. One country, 700+ languages: NLP challenges for underrepresented languages and dialects in Indonesia. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7226–7249, Dublin, Ireland. Association for Computational Linguistics.

Anthropic. 2024. Claude 3 model card.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Baidu Inc. 2023. Baidu launches ernie 4.0 foundation model, leading a new wave of ai-native applications.

Robert S. Bauer. 2016. The hong kong cantonese language: Current features and future prospects. *Global Chinese*, 2(2):115–161.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Po-Heng Chen, Sijia Cheng, Wei-Lin Chen, Yen-Ting Lin, and Yun-Nung Chen. 2024. Measuring taiwanese mandarin language understanding. *arXiv preprint arXiv:2403.20180*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

ByteDance Doubao. 2024. Doubao models.

Jia Tina Du, Iris Xie, and Jenny Waycott. 2020. Marginalized communities, emerging technologies, and social innovation in the digital age: Introduction to the special issue. *Information Processing & Management*, 57(3):102235.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2024. *Ethnologue: Languages of the World*, 27 edition. SIL International, Dallas.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lacalle, and Mikel Artetxe. 2024a. Bertaqa: How much do language models know about local culture? *Advances in Neural Information Processing Systems*, 37:34077–34097.

Julen Etxaniz, Oscar Sainz, Naiara Miguel, Itziar Aldabe, German Rigau, Eneko Agirre, Aitor Ormazabal, Mikel Artetxe, and Aitor Soroa. 2024b. Latxa: An open language model and evaluation suite for basque. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14952–14972.

Ziru Fu, Yu Cheng Hsu, Christian S Chan, Chaak Ming Lau, Joyce Liu, and Paul Siu Fai Yip. 2024. Efficacy of chatgpt in cantonese sentiment analysis: comparative study. *Journal of Medical Internet Research*, 26:e51069.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. 2024. Are we done with mmlu? *arXiv preprint arXiv:2406.04127*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Vipul Gupta, David Pantoja, Candace Ross, Adina Williams, and Megan Ung. 2024. Changing answer order can decrease mmlu accuracy. *arXiv preprint arXiv:2406.19470*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Kung Yin Hong, Lifeng Han, Riza Theresa Batista-Navarro, and Goran Nenadic. 2024. Cantonmt: Cantonese-english neural machine translation looking into evaluations. In *Proceedings of the 16th Conference of the Association for Machine Translation in the Americas (Volume 2: User Track)*, pages 133–144.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. 2024. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in Neural Information Processing Systems*, 36.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card.

Jiyue Jiang, Liheng Chen, Pengan Chen, Sheng Wang, Qinghang Bao, Lingpeng Kong, Yu Li, and Chuan Wu. 2024. How far can cantonese nlp go? benchmarking cantonese capabilities of large language models. *arXiv e-prints*, pages arXiv–2408.

Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, et al. 2024. Arabicmmlu: Assessing massive multitask language understanding in arabic. *arXiv preprint arXiv:2402.12840*.

Chaak Ming Lau. 2024. Ideologically driven divergence in cantonese vernacular writing practices. In J.-F. Dupré, editor, *Politics of Language in Hong Kong*. Routledge.

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, Marseille, France. European Language Resources Association.

Genevieve Leung and Yuuko Uchikoshi. 2012. Relationships among language ideologies, family language policies, and children's language achievement: A look at cantonese-english bilinguals in the us. *Bilingual Research Journal*, 35(3):294–313.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. 2023. Cmmlu: Measuring massive multitask language understanding in chinese. *arXiv preprint arXiv:2306.09212*.

Mistral. 2024. Mistral nemo.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

Irene Plaza, Nina Melero, Cristina del Pozo, Javier Conde, Pedro Reviriego, Marina Mayor-Rocher, and María Grandury. 2024. Spanish and llm benchmarks: is mmlu lost in translation? *arXiv preprint arXiv:2406.17789*.

Itesh Sachdevl, Richard Bourhis, Sue-wen Phang, and John D'Eye. 1987. Language attitudes and vitality perceptions: Intergenerational effects amongst chinese canadian communities. *Journal of Language and Social Psychology*, 6(3-4):287–307.

SenseTime. 2024. Sensetime introduces sensechat (cantonese) to hong kong users, delivering localised ai experiences free-of-charge.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2024. Kmmlu: Measuring massive multitask language understanding in korean. *arXiv preprint arXiv:2402.11548*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Saeid Asgari Taghanaki, Aliasgahr Khani, and Amir Khasahmadi. 2024. Mmlu-pro+: Evaluating higher-order reasoning and shortcut learning in llms. *arXiv preprint arXiv:2409.02257*.

Zhi-Rui Tam, Ya-Ting Pai, Yen-Wei Lee, Jun-Da Chen, Wei-Min Chu, Sega Cheng, and Hong-Han Shuai. 2024. An improved traditional chinese evaluation suite for foundation model. *arXiv preprint arXiv:2403.01858*.

toastynews. 2020. openrice-senti.

Maria Tsapali and Hiu Ching Wong. 2023. The future of cantonese and traditional chinese among newly arrived hong kong immigrant children in the united kingdom–a study on parents'attitudes, challenges faced and support needed. *Cambridge Educational Research e-Journal*, 10:14–31.

Sshubam Verma, Mohammed Safi Ur Rahman Khan, Vishwajeet Kumar, Rudra Murthy, and Jaydeep Sen. 2024. Milu: A multi-task indic language understanding benchmark. *arXiv preprint arXiv:2411.02538*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Andrea W Wen-Yi, Unso Eun Seo Jo, and David Mimno. 2025. Do chinese models speak chinese languages? *arXiv preprint arXiv:2504.00289*.

Genta Winata, Shijie Wu, Mayank Kulkarni, Thamar Solorio, and Daniel Preotiuc-Pietro. 2022. Cross-lingual few-shot learning on unseen languages. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 777–791, Online only. Association for Computational Linguistics.

Rong Xiang, Emmanuele Chersoni, Yixia Li, Jing Li, Chu-Ren Huang, Yushan Pan, and Yushi Li. 2024. Cantonese natural language processing in the transformers era: a survey and current challenges. *Language Resources and Evaluation*, pages 1–27.

Yuemei Xu, Ling Hu, Jiayi Zhao, Zihan Qiu, Kevin Xu, Yuqi Ye, and Hanwen Gu. 2024. A survey on multilingual large language models: Corpora, alignment, and bias. *arXiv preprint arXiv:2404.00929*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

William Yiu. 2024. Hong kong government to adopt city's own chatgpt-style tool after openai further blocks access.

Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *arXiv preprint arXiv:2304.02015*.

Daoguang Zan, Bei Chen, Fengji Zhang, Dianjie Lu, Bingchao Wu, Bei Guan, Yongji Wang, and Jian-Guang Lou. 2022. Large language models meet nl2code: A survey. *arXiv preprint arXiv:2212.09420*.

Chen Zhang, Zhiyuan Liao, and Yansong Feng. 2025. Cross-lingual transfer of cultural knowledge: An asymmetric phenomenon. *arXiv preprint arXiv:2506.01675*.

Lubei Zhang, Linda Tsung, and Xian Qi. 2023. Home language use and shift in australia: Trends in the new millennium. *Frontiers in Psychology*, 14:1096147.

Quasi-symbolic Semantic Geometry over Transformer-based Variational AutoEncoder

Yingji Zhang[1†], **Danilo S. Carvalho**[1,3], **André Freitas**[1,2,3]
[1] Department of Computer Science, University of Manchester, United Kingdom
[2] Idiap Research Institute, Switzerland
[3] National Biomarker Centre, CRUK-MI, Univ. of Manchester, United Kingdom
`{firstname.lastname}@[postgrad.]†manchester.ac.uk`

Abstract

Formal/symbolic semantics can provide canonical, rigid controllability and interpretability to sentence representations due to their *localisation* or *composition* property. How can we deliver such property to the current distributional sentence representations to better control and interpret the generation of language models (LMs)? In this work, we theoretically frame the sentence semantics as the composition of *semantic role - word content* features and propose the formal semantic geometrical framework. To inject such geometry into Transformer-based LMs (i.e. GPT2), we deploy a supervised Transformer-based Variational AutoEncoder, where the sentence generation can be manipulated and explained over low-dimensional latent Gaussian space. In addition, we propose a new probing algorithm to guide the movement of sentence vectors over such geometry. Experimental results reveal that the formal semantic geometry can potentially deliver better control and interpretation to sentence generation.

1 Introduction

Language Models (LMs) have provided a flexible scaling-up foundation for addressing a diverse spectrum of tasks (Touvron et al., 2023). Nonetheless, the question remains: can we develop language representations/models that offer more granular levels of control and interpretation from the perspective of "formal/structural" semantics? Addressing this question will enable us to enhance the controllability, interpretability, and safety of LMs.

Formal semantics, which provides a canonical, granular, and rigid representation, have been investigated for thousands of years with well established theoretical frameworks, such as Montague Semantics (Dowty et al., 2012), Davidsonian Semantics (Davidson, 1967), Semantic Role Labelling (SRL, Palmer et al. (2010)), and Argument Structure Theory (AST, Jackendoff (1992)). One typical characteristic of such formal semantics is the *localisation* or *composition* property. For example, in the sentence: *animals require oxygen for survival*, the words are functionally combined into sentence semantics: $\lambda x(\text{animals}(x) \rightarrow \text{require}(x, \text{oxygen}))$ where x is the variable of any entity within a logical structure. In this case, we can localise the sentence semantics by replacing x with *birds*, etc. This localised process indicates the interpretation in Cognitive Science (Lees, 1957; Smolensky, 2006). However, such localisation is precisely what current distributional semantics lack, thereby limiting their controllability and interpretability.

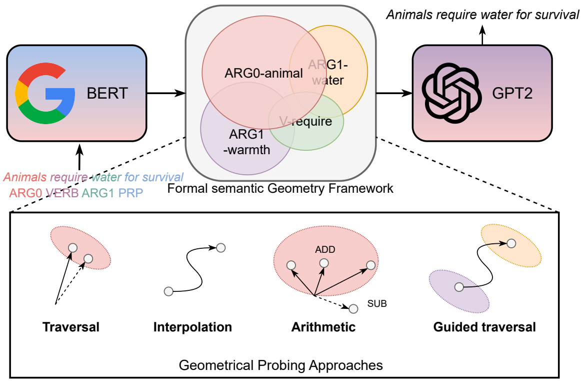

Animals require water for survival
BERT
ARG0-animal
ARG1-water
V-require
ARG1-warmth
GPT2
Animals require water for survival
ARG0 VERB ARG1 PRP
Formal semantic Geometry Framework
Traversal
Interpolation
ADD
SUB
Arithmetic
Guided traversal
Geometrical Probing Approaches

Figure 1: Overview: latent sentence semantics can be decomposed into *semantic role- word content* features.

Disentanglement (Bengio, 2013), which refers to the feature-dimension alignment, can potentially provide such localisation, which has been widely investigated to localise image features, such as *nose* in facial images (Esser et al., 2020; Jeon et al., 2019; Liu et al., 2021). In Transformers (Vaswani et al., 2017), however, token embeddings, residual stream, and attention have the *polysemanticity* phenomenon (Elhage et al., 2022), meaning that multiple dimensions contribute to a feature. Although some prior studies explored the possibility of language disentanglement, most are focused on coarse-grained/task-specific semantic features, such as sentiment, within the context of style-transfer tasks

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 12–29
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

(John et al., 2019; Bao et al., 2019; Hu and Li, 2021; Vasilakes et al., 2022; Gu et al., 2022; Liu et al., 2023a; Gu et al., 2023).

In this work, we focus on the localisation of *general* semantic features of sentences over distributional space to shorten the gap between deep latent semantics and formal linguistic representations (Gildea and Jurafsky, 2000; Banarescu et al., 2013; Mitchell, 2023), integrating the flexibility of distributional-neural models with the properties of linguistically grounded representations, facilitating both interpretability and generative control from the perspective of formal semantics. We specifically choose the conceptual dense explanatory sentences from WorldTree (Jansen et al., 2018) due to their clear formal semantic representation designed in the explanatory, cognitive reasoning task.

In the NLP domain, Variational AutoEncoders (VAEs, Kingma and Welling (2013)) have been recognized as a prominent foundation for investigating generation control and interpretation through the observable low-dimensional smooth and regular latent spaces (e.g., std Gaussian space) (John et al., 2019; Li et al., 2022b; Bao et al., 2019; Mercatali and Freitas, 2021; Felhi et al., 2022; Vasilakes et al., 2022). Therefore, we probe the localisation property of formal semantics over latent sentence spaces under VAE architecture. Specifically:

(1) We first propose a geometrical framework to present the formal semantic features of sentences as *semantic role - word content* pairs (denoted as role-content) from the perspective of AST (Jackendoff, 1992) within the compositional distributional model (Clark et al., 2008). Subsequently, **(2)** we introduce a supervised approach for learning the role-content features of explanatory sentences in latent spaces. **(3)** Additionally, we contribute to a method to control sentence generation by navigating the sentence vectors across different role-content features within our geometric framework. **(4)** Our findings reveal that the role-content features are encoded as a convex cone in the latent sentence space (Figure 1). This semantic geometry facilitates the localisation of sentence generation by enabling the manipulation of sentence vectors through traversal and arithmetic operations within the latent space.

2 Related work

Formal-distributional semantics. Integrating distributional semantics with formal / symbolic semantics is challenging due to the difficulty of optimisation over discrete space (van Krieken et al., 2023). In the Reasoning domain, existing approaches usually perform symbolic behaviour via explicitly symbolic representation injection, including graph (Khashabi et al., 2018; Khot et al., 2017; Jansen et al., 2017; Thayaparan et al., 2021), linear programming (Valentino et al., 2022b; Thayaparan et al., 2024), adopting iterative methods, using sparse or dense encoding mechanisms (Valentino et al., 2020; Lin et al., 2020; Valentino et al., 2022a; Bostrom et al., 2021), or synthetic natural language expression (Clark et al., 2020; Yanaka et al., 2021; Fu and Frank, 2024), among others. Comparatively, we explore the formal semantic property over distributional semantics via latent sentence geometry, which can potentially deliver better interpretation and control to current LMs.

Language geometry. There is a line of work that studies the geometry of word and sentence representations (Arora et al., 2016; Mimno and Thompson, 2017; Ethayarajh, 2019; Reif et al., 2019; Li et al., 2020a; Chang et al., 2022; Jiang et al., 2024a). E.g., $king - man + woman = queen$, which the word vectors can be manipulated with geometric algebra. This phenomenon indicates the linear subspaces in language representations, similar features are encoded as a close direction in latent space, which has been widely explored ranging from word (Mikolov et al., 2013a) to sentences (Ushio et al., 2021), Transformer-based LMs (Merullo et al., 2023; Hernandez et al., 2023), and multi-modal models (Trager et al., 2023; Huh et al., 2024). Under the linear subspace hypotheses, a significant work explored the interpretability (Li et al., 2022a; Geva et al., 2022; Nanda et al., 2023) and controllability (Trager et al., 2023; Merullo et al., 2023; Turner et al., 2023) of neural networks. In this work, we emphasise the formal semantic geometry for bridging the distributional and formal semantics, which is currently under-explored.

Language disentanglement. Disentanglement, refers to separating features along dimensions (Bengio, 2013), leading to clear geometric and linear representations. In the NLP domain, prior studies explored the disentanglement between specific linguistic perspectives, such as sentiment-content (John et al., 2019), semantic-syntax (Bao et al., 2019), and negation-uncertainty (Vasilakes et al., 2022), or syntactic-level disentanglement (Mercatali and Freitas, 2021; Felhi et al., 2022). However, those approaches focused on disentangling coarse-grained/task-specific semantic features. In this work, we contribute to a new lens on the disentanglement (separation) of "general" sentence features from the perspective of formal semantics.

3 Formal Semantic Geometry

In this section, we first define the sentence semantic features as *semantic role - word content* from the perspective of formal semantics. Then, we link the semantic features with distributional vector spaces in which each *semantic role - word content* is encoded as a convex cone, as shown in Figure 1.

Formal semantic features. For formal / structural semantics, *Argument Structure Theory (AST)* (Jackendoff, 1992; Levin, 1993; Rappaport Hovav and Levin, 2008) provides a model for representing sentence structure and meaning of sentences in terms of the interface between the their syntactic structure and the associated semantic roles of the arguments within those sentences. It delineates how verbs define the organisation of their associated arguments and the reflection of this organisation in a sentence's syntactic realisation. AST abstracts sentences as predicate-argument structures, where the predicate p (associated with the verb) has a set of associated arguments arg_i, where each argument has an associated positional component i and a thematic/semantic roles r_i, the latter categorising the semantic functions of arguments in relation to the verb (e.g. agent, patient, theme, instrument). In the context of this work, the AST predicate-argument representation is associated with a lexical-semantic representation of the content c_i of the term t_i.

In this work, we simplify and particularise the relationship between the argument structure and the distributional lexical semantic representation as a *role-content* relation, where the structural syntactic/semantic relationship is defined by its shallow semantics, i.e. as the composition of the content of the terms, their position in the predicate-argument (PArg) structure (arg_i) and their semantic roles (SRs) (r_i: $pred$, arg), as described below:

$$\underbrace{animals}_{ARG0}\ \underbrace{require}_{PRED}\ \underbrace{oxygen}_{ARG1}\ \underbrace{for\ survival}_{ARGM-PRP}$$

Therefore, we define the semantics of sentences, $sem(s)$, as the compositions between *role-content*, which can be described as follows:

$$sem(s) = \underbrace{t_1(c_1,r_1)}_{i.e., ARG0-animals} \oplus \cdots \oplus \underbrace{t_i(c_i,r_i)}_{PRP-survival}$$

Where $t_i(c_i,r_i) = c_i \otimes r_i$ represents the semantics of term t_i with content c_i (i.e., *animals*) and SRL r_i (i.e., *ARG0*) in context s. $\otimes$: connects the meanings of words with their roles, using the compositional-distributional semantics notation of (Smolensky and Legendre, 2006; Clark and Pulman, 2007; Clark et al., 2008). $\oplus$: connects the lexical semantics (word content + structural role) to form the sentence semantics. To deliver the localisation or composition property, the sentence semantics should be able to present separation or disentanglement under connector $\oplus$. E.g., replacing *ARG0-animals* with *ARG0-fishes*.

Formal semantic features in vector space. After defining the semantic features of sentences, we propose the concept of a *convex cone of semantic feature*. In linear algebra, a *cone* refers to a subset of a vector space that is convex if any $\alpha\vec{v_i} + \beta\vec{v_j}$ if any $\vec{v_i}$ and $\vec{v_j}$ belong to it. α and β are positive scalars. Formally, the definition of convex cone, C, is described as a set of vectors: $C = \{x \in V | x = \sum_{i=1}^{n} \alpha_i v_i, \alpha_i \geq 0, v_i \in R\}$ where x is an element vector in vector space $\mathbb{R}$, v_i are the basis vectors. α_i are non-negative scalars. In this context, we consider each *role-content* feature as a convex cone, C, corresponding to a hypersolid in high-dimensional vector space: $C_{c_i,r_i} = \{t(c_i,r_i) | t(c_i,r_i) \in sem(s), s \in corpus\}$ where $t(c_i,r_i)$ represents the basis vector in C_{c_i,r_i} (Figure 2). According to set theory, we can define the formal semantic space as follows:

Assumption1: *The sentence semantic space is the union of all unique C_{c_i,r_i} convex cones:*

$$C_{c_1,r_1} \cup C_{c_2,r_2} \cup \cdots \cup C_{c_{V(c)},r_{V(r)}}$$

V is the vocabulary of a corpus. Based on Assumption1, we can establish:

Proposition1: *The geometrical location of sentence semantic vectors, $sem(s)$, can be determined by the intersection of different C_{c_i,r_i}:*

$$\begin{aligned} sem(s) &= t_1(c_1,r_1) \oplus \cdots \oplus t_i(c_i,r_i) \\ &= \{t_1(c_1,r_1)\} \oplus \cdots \oplus \{t_i(c_i,r_i)\} \\ &= C_{c_1,r_1} \cap C_{c_2,r_2} \cap \cdots \cap C_{c_i,r_i} \end{aligned}$$

4 Geometrical Formal Semantic Control

In this section, we first show that our formal semantic geometry can interpret sentence generation, such as arithmetic (Shen et al., 2020), and extend the "Linear Representation Hypothesis". Then, we propose a new semantic control approach, which recursively traverses the latent dimensions to probe the semantic geometry over latent spaces.

Geometrical algebra interpretability. Arithmetic has been considered a common way to control word or sentence semantics over latent spaces (Mikolov et al., 2013b). E.g., the addition operation can steer the sentence semantics (Shen et al., 2020; Mercatali and Freitas, 2021; Liu et al., 2023b), or linear interpolation can generate smooth intermediate sentences (Hu et al., 2022). However, they lack an explanation for these phenomena. We show that our geometrical framework can provide an intuitive explanation for these phenomena.

For linear interpolation, for example, it takes two sentences x_1 and x_2 and obtains latent vectors z_1 and z_2, respectively. It interpolates a path $z_k = z_1 \cdot (1-k) + z_2 \cdot k$ with k increased from 0 to 1 by a step size of 0.1. Given two sentences with one role-content set overlap, C_{c_j,r_j}. We can describe:

$$\begin{aligned} &sem(s_1) \cap sem(s_2) \\ &= \{C^{s_1}_{c_1,r_1} \cap \cdots \cap C^{s_1}_{c_i,r_i}\} \cap \{C^{s_2}_{c_1,r_1} \cap \cdots \cap C^{s_2}_{c_i,r_i}\} \\ &= \{C^{s_1}_{c_1,r_1} \cap \cdots \cap C^{s_2}_{c_i,r_i}\} \cap C^{s_{1(2)}}_{c_j,r_j} \end{aligned}$$

According to the definition of convex cone, if z_1 and z_2 are left in $C^{s_{1(2)}}_{c_j,r_j}$, the weighted sum vector, z_t, is also in $C^{s_{1(2)}}_{c_j,r_j}$. Therefore, the intermediate sentence semantics can be described as:

$$\begin{aligned} &sem(s^t_{1\to 2}) \\ &= (1-k) \times sem(s_1) + k \times sem(s_2) \\ &= \{\{z_1 \cdot (1-k) + z_2 \cdot k\}, \ldots \{\ldots\}\} \cap C^{s_{1(2)}}_{c_j,r_j} \end{aligned}$$

That is, the intermediate sentences will hold the $\{c_j, r_j\}$ information during interpolation.

Linear representation hypothesis. "Linear representation hypothesis" refers to high-level concepts being represented linearly as directions in representation space, which has been widely evaluated to interpret Large LMs' mechanism (Marks and Tegmark, 2023; Xie et al., 2021; Wang et al., 2024; Jiang et al., 2024b; Park et al., 2023, 2024). However, a main challenge for this hypothesis is that it's not clear what constitutes a high-level concept.

Our geometrical framework can further support and extend this hypothesis by answering the questions: What and how are they "linearly" encoded?

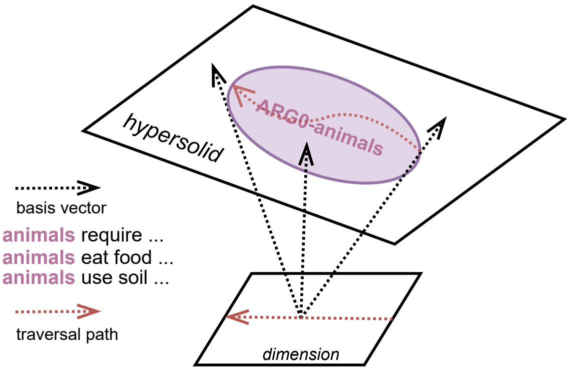


Figure 2: Algorithm 1: by modifying the latent dimensions, we can control the movement of latent vectors over latent space.

For example, given a set of N atomic sentences: s_i: *bird is a kind of living thing* varying the content of arg1. Their semantics can be described below:

$$sem(s) = \{C^{s_i}_{c_i,arg1}, \ldots\} \cap \cdots \cap C_{living\ thing,arg2}$$
$$\text{, where } c_i \in \{\text{tiger}, \text{bird}, \ldots\}$$

In this case, the concept *living thing* is encoded as a convex cone where all different $C^{s_i}_{c_i,arg1}$ contribute to its boundary, leading to a direction. The hierarchical relations between *living thing* and *bird, etc.* are determined by the convex cones *is a kind of.*

Guided traversal. Since we describe different sentence semantic features, $\{c_i, r_i\}$, as distinct convex cones, C_{c_i,r_i}, within a N-dimensional vector space, $V \in \mathbb{R}^N$, we can linearly divide each basis dimension, $i \in \{1, \ldots, N\}$, into different value regions, $[a,b]^{(i)}$, based on minimal information entropy. Consequently, there is a sequence of dimensional subspaces for each semantic feature. Thus, movement between different C_{c_i,r_i} regions can be achieved by moving out the dimensional regions within this sequence. This process can be implemented via a decision tree. In figure 3, for example, we can move the sentence from $C_{pred,causes}$ to $C_{pred,means}$ by modifying the values started from *dim 21* ≤ -0.035, ..., ending at *dim 10* ≤ -1.11. By traversing the tree path, we can control the sentence generation by moving between convex cones, detailed in Algorithm 1.

Based on our algorithm, we can use classification metrics as proxy metrics to evaluate latent space geometry. E.g., accuracy and recall for measuring feature *separability* and *density*.

Algorithm 1 Guided latent space traversal

```
Datasets: $D = \{s_1, \dots, s_n\}$
Labels: $Y = \{y_1, \dots, y_n\}$, $y_i \in \{0, 1\}$
# 0:pred-causes, 1:pred-means
Seed: $s$ = fire causes chemical change
for $s_i \in D$ do
    $z_i \leftarrow$ Encoder($s_i$)
end for
$X \leftarrow \{z_1, \dots, z_n\}$
tree ← DecisionTreeClassifier($X, Y$)
path ← filter(tree) # choose the shortest path between $C_0$ and $C_1$
$z \leftarrow$ Encoder($s$)
for node ∈ path do
    (dim, range, yes/no) ← node
    if in current branch do
        z[dim] ← $v \notin$ range if yes else $v \in$ range
    else do
        z[dim] ← $v \in$ range if yes else $v \notin$ range
end for
$s \leftarrow$ Decoder(z) # fire means chemical change
```

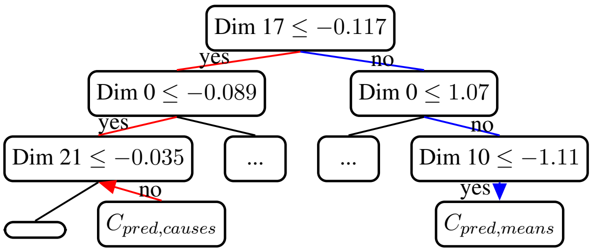


Figure 3: Traversal between different role-content sets by moving along the tree path.

5 SRL-Conditional VAE

In this section, we investigate the architecture of VAE to integrate the latent sentence space with LMs and propose a supervision approach to learn formal semantic geometry (i.e., role-content).

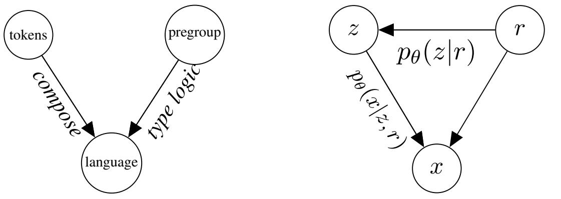
Model architecture. We consider Optimus (Li et al., 2020b) as the foundation which used BERT and GPT2 as Encoder and Decoder, respectively. In detail, the sentence representation, Embed(x), encoded from BERT[cls] will first transform into a Gaussian space by learning the parameters μ and σ through multilayer perceptions W_μ, W_σ. The final latent sentence representations can be obtained via: $z = W_\mu \times \text{Embed(x)} + W_\sigma$, which, as an additional Key and Value, is concatenated into the original Key and Value weights of GPT2, which can be described as: $\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q[z;K]^T}{\sqrt{d}})[z;V]$ where Q has the shape $\mathbb{R}^{\text{seq}\times 64}$, K, V has the shape $\mathbb{R}^{(\text{seq}+1)\times 64}$ (64 is dimension of GPT2 attention, seq is sequence length). Since Q represents the target, K and V represent the latent representations. By intervening the KV with z, we can learn the transformation between latent space and observation distribution.

Optimisation. It can be trained via the evidence lower bound (ELBO) on the log-likelihood of the data x (Kingma and Welling, 2014). To bind the word content and semantic role information in latent space, we conditionally inject the semantic role sequence into latent spaces where the latent space z and semantic role r are dependent. The joint distribution can be described as:

$$P_\theta(x, r, z) = \underbrace{P_\theta(x|z, r)}_{likelihood} \times \underbrace{P_\theta(z|r)}_{prior} \times P(r)$$

Specifically, we first model the categorical structures by encoding the semantic roles sequence to learn the prior distribution with parameters $\mu^{(srl)}$ and $\sigma^{(srl)}$. Then, we jointly encode semantic roles and lexical tokens to learn the approximate posterior parameterised by μ and σ. By minimising the Kullback-Leibler (KL) divergence between prior and approximate posterior, the semantic features can be encoded in the latent sentence space. Moreover, to avoid the KL vanishing problem, which refers to the KL term in the ELBO becomes very small or approaching zero, we select the cyclical schedule to increase weights of KL β from 0 to 1 (Fu et al., 2019) and a KL thresholding scheme (Li et al., 2019) that chooses the maximum between KL and threshold λ. The final objective function can be described as follows:

$$\mathcal{L}_{\text{CVAE}} = -\mathbb{E}_{q_\phi(z|r,x)}\Big[\log p_\theta(x|z, r)\Big] + \beta \sum_i \max\left[\lambda, \text{KL}q_\phi(z_i|x, r)||p(z_i|r)\right]$$



(a) CDM (Clark et al., 2008)
(b) SRL-Conditional VAE

Figure 4: Comparison between Compositional Distributional Model (CDM) (left) and SRL-Conditional VAE (right).

where q_ϕ represents the approximated posterior (i.e., encoder). i is the i-th latent dimension.

6 Empirical analysis

In the experiment, we quantitatively and qualitatively evaluate the latent space geometry via geometrical probing approaches: (1) traversal, (2) arithmetic, and (3) guided traversal. All experimental details are provided in Appendix A.

6.1 Latent Traversal

Qualitative evaluation. Traversal refers to the random walk over latent space. It can be done by decoding the latent vector in which each dimension is resampled and other dimensions are fixed (Higgins et al., 2017; Kim and Mnih, 2018; Carvalho et al., 2023). Given a latent vector from a "seed" sentence, we can traverse its neighbours to evaluate the geometry. As illustrated in Table 1, those traversed sentences can hold the same content under different semantic roles as the input, such as *automobile* in *ARG1*, indicating *role-content* feature separation in latent spaces.

an automobile is a kind of vehicle

an automobile is a kind of moving object
an automobile is a kind of object

an airplane is a kind of vehicle
a car is a kind of vehicle

Table 1: Traversal showing held semantic factors in explanations corpus.

Quantitative evaluation. Next, we employ t-SNE (Van der Maaten and Hinton, 2008) to examine *role-content* features cluster and separation over latent space (i.e., *natural clustering property* (Bengio, 2013)). In the corpus, however, due to the small number of data points within each role-content cluster, t-SNE cannot capture the differences between clusters well, resulting in the visualized latent space not displaying good role-content separability (top in figure 5). Therefore, we increase the number of data points in different role-content clusters by traversing each and keeping those resulting data points with the same role-content. Then, we visualise the role-content cluster at the bottom of figure 5. We can find that the features are clustered and separated over the latent space. If this was not the case, after traversing the resulting vectors from the same role-content cluster, the visualization should show the same entanglement as the original datapoints distribution.

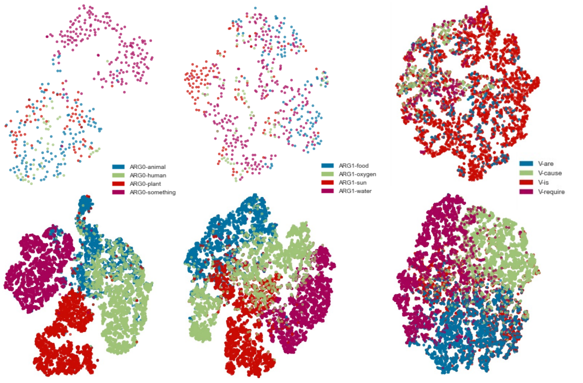


Figure 5: t-SNE plot of role-content distribution before and after traversal. From left to right are ARG0-(animal, human, plant, and something), ARG1-(food, oxygen, sun, and water), and predicate-(are, cause, is, require) (top: original role-cluster distribution, bottom: distribution after traversal). PCA plots are in Figure 9.

6.2 Latent Arithmetic

Qualitative evaluation. In addition, we demonstrate the geometric properties via interpolation in Table 2. For the top-most one, we can observe that sentences are smoothly moved from source to target (e.g., from *beach ball* to *atom* connected

a beach ball is a kind of container
1. a pool table is a kind of object
2. a balloon is a kind of object
3. a magnet is a kind of object
4. a neutron is a kind of particle
5. a proton is a kind of particle
an atom is a kind of particle

protons are found in the nucleus of an atom
1. protons are found in the nucleus of an atom
2. 1 atom is positive 1 in electric charge
3. 1 in 6000 is equal to 27 in 10 years
4. if protons and neutrons have the same number of neutrons then those two particles are physically closer than one another
5. if a neutron has a negative -10 electric charge then the atom will not be able to move
6. if a neutron has a negative -10 electric charge then the neutron will not have a positive electric charge
if a neutral atom loses an electron then an atom with a positive charge will be formed

Table 2: Interpolation examples (top: interpolation between sentences with similar semantic information, bottom: interpolation between sentences with different semantic information). Only unique sentences shown.

by *ballon*, *magnet*, *neutron*, and *proton*) where the same role-content (i.e., *pred-is*) unchanged. In contrast, the second case doesn't display the smooth interpolation path. E.g., the third sentence connecting different semantic structures is unrelated to both source and target due to a discontinuous space gap between different clusters. Both indicate that the explanatory sentences might be clustered according to different semantic role structures.

s_1: animals require food for survival
s_2: animals require warmth for survival
animals eat plants
animals produce milk
animals usually eat plants
animals eat berries ; plants
animals require food to survive
animals require shelter to survive

---

s_1: water vapor is invisible
s_2: the water is warm
igneous rocks are found under the soil
quartz is usually very small in size
quartz is formed by magma cooling
quartz is made of iron and zinc
silica is made of argon and argon
sedimentary is formed by lithosphere collapsing

Table 3: $s_1 \pm s_2$ (top: addition, bottom: subtraction).

Following the definition of convex cone, we next traverse the resulting sentence after adding or subtracting two sentences with the same role-content feature. As illustrated in Table 3, the adding operation tends to hold the same role-content (e.g., *ARG0-Animals*) as inputs. In contrast, the subtraction loses such control, e.g., from *ARG1-water* to *ARG1-quartz*. More similar observations are in Table 11. These results corroborate our geometry.

Quantitative evaluation. Next, we quantitatively assess our geometry framework by calculating the ratio of the same role-content results from the vector addition and subtraction for all sentence pairs with a matching role. As illustrated in Figure 6, the ADDed results (dark blue) can greatly hold the same token-level semantics (role-content) as inputs, indicating our geometrical framework. In contrast, the SUBed results (shallow blue) suffer from semantic shift. Similar observations for VERB and ARG1 can be found in Figure 11 and 12. Besides, we can quantify each role-content cluster's geometrical area by calculating the cosine similarity between randomly selected sentence pairs in this cluster. We report the maximal and minimal distance in Figure 7. Similar observations for VERB and ARG1 can be found in Figure 13 and 14.

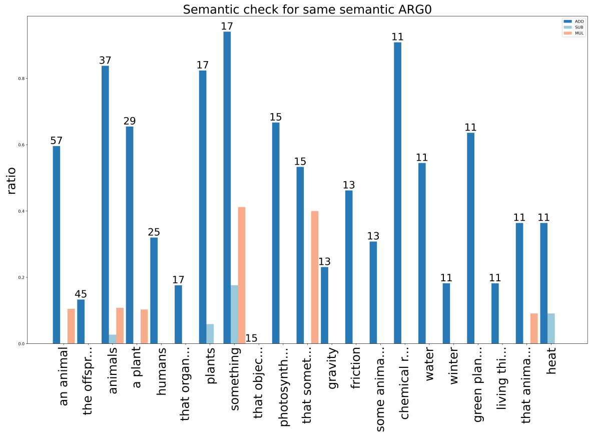


Figure 6: Arithmetic, $s_1 \pm s_2$, for ARG0 with contents (dark blue: addition, shallow blue: subtraction, orange: element-wise production).

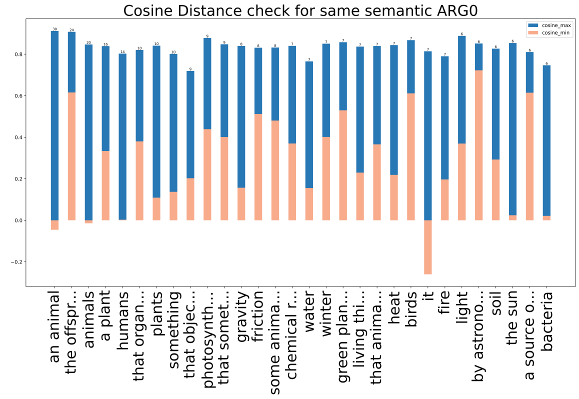


Figure 7: Evaluating the geometrical size of role-content clusters (blue: max, orange: min).

6.3 Guided Latent Traversal

Finally, we examine the semantic geometry via algorithm1. The categories selected below are chosen based on their frequencies, ensuring a balanced distribution during the classifier's training process.

Qualitative evaluation. Firstly, we evaluate the traversal between different semantic role structures, e.g, conditional and atomic sentences. Table 4 shows that the cluster of the generated sentence changes as the values of different dimensions change sequentially (e.g., the first three sentences hold the same characteristic *if ... then ...* as the input. The remaining sentences gradually move closer to the target characteristics, such as *is*). Meanwhile, the sentences can hold the subject, *something*, during the movement, corroborating our geometry framework. Next, we evaluate the traversal between predicates. Table 5 shows the movement between verbs (*cause* and *mean*). We can observe that the predicate is modified from *causes* to *mean*. In the traversal process, some sentences fall into the *V-is* region. The reason is that the *V-is* cluster is widely scattered in latent space (shown in Figure 5), which leads to a big overlap between *V-is* and *V-mean*. Moreover, we calculate the ratio of the generated sentences that hold the expected predicate, *mean*, from 100 sentences with predicate *cause*. The ratio is 0.71, which indicates that the decision tree is a reliable way to navigate the movement of sentences. Finally, we evaluate the traversal between arguments. Table 6 shows the movement from argument *water* to *something*. Similarly, the ARG1 can be modified from *water* to *something* following its path. Besides, the final generated explanation still holds a similar semantic structure, *is a kind of*, compared with input.

if something receives sunlight it will absorb the sunlight
Dim27: if a thing absorbs sunlight then that thing is warmer
Dim12: if something is eaten then that something produces heat
Dim08: if something gets too hot in sunlight then that something is less able to survive
Dim03: something contains physical and chemical energy
Dim21: something contains sunlight
Dim10: some things are made of matter
Dim00: something is made of atoms
Dim17: a forest contains life
Dim00: something that is cold has a lower temperature
Dim21: something rises in temperature
Dim00: something is formed from things dissolved in water
Dim30: something that is cold has fewer nutrients
Dim21: something that is not moved is dead

Table 4: Movement from *conditional* to *atomic* sentences.

Quantitative evaluation. Finally, we use classification metrics, including accuracy (*separability*) and recall (*density*), as proxy metrics to assess latent space geometry. As shown in Table 7, all features show higher separation where argument1 leads to the highest separation, indicating latent space geometry.

fire causes chemical change
Dim06: fire causes chemical changes
Dim22: fire causes chemical reactions
Dim02: fire can cause harm to plants
Dim27: smoke can cause harm to organisms
Dim14: fire causes physical harm to objects
Dim24: fire can cause chemical changes
Dim08: fire destroys material
Dim01: fire means chemical change
Dim14: waste means igneous metal
Dim06: combustion means burning
Dim00: combustion means chemical changes
Dim21: combustion means burning
Dim00: fire is formed by thermal expansion
Dim18: fire chemical means chemical energy
Dim03: fire is corrosive

winter means cold environmental temperature
Dim03: winter means cold - weather
Dim18: winter means cold weather
Dim00: winter means weathering
Dim21: drought means high temperatures / low precipitation
Dim00: winter means high amounts of precipitation
Dim06: drought causes natural disasters
Dim14: drought has a negative impact on crops
Dim01: drought has a negative impact on animals
Dim08: drought causes animal populations to decrease
Dim24: drought causes ecosystem loss
Dim14: drought causes animals to have lower natural temperature
Dim27: cold climates causes wildfires
Dim02: climate change can cause low rainfall
Dim22: global warming causes droughts
Dim06: winter causes weather patterns

Table 5: Movement between *cause* and *mean*.

water is a kind of substance
Dim12: water is a kind of substance
Dim00: water is a kind of liquid
Dim23: liquid is a kind of material
Dim29: water has a positive impact on a process
Dim17: absorbing water is similar to settling
Dim06: absorbing is similar to reducing
Dim21: absorbing something is similar to absorbing something
Dim04: storing something means being protected
Dim06: producing something is a kind of process
Dim04: storing something is similar to recycling
Dim21: absorbing something is a kind of process
Dim01: absorbing something can mean having that something
Dim22: folding something is similar to combining something
Dim07: improving something is a kind of transformation
Dim11: absorbing something is a kind of method
Dim07: absorbing something is a kind of process

Table 6: Movement from *water* to *something*.

Formal semantic features	**separation↑**	**density↑**
predicate (causes, means)	0.87	0.92
argument1 (water, something)	0.95	0.48
structure (condition, atomic)	0.58	0.55

Table 7: Proxy metrics for latent space geometry.

7 Conclusion and Future Work

In this study, we investigate the localisation of general semantic features to enhance the controllability and explainability of distributional space from the perspective of formal semantics, which is currently under-explored in the NLP domain. We first propose the formal semantic features as *role-content* and define the corresponding geometrical framework. Then, we propose a supervision approach to bind the semantic role and word content. In addition, we propose a novel traversal probing approach to assess the latent space geometry based on information set and entropy. We extensively evaluate the latent space geometry through geometrical operations, such as traversal, arithmetic, and our guided traversal. Experimental results indicate the existence of formal semantic geometry.

Since recent theoretical works reveal that the LLMs can encode linear symbolic concepts (Jiang et al., 2024b), in the future, we will explore their in-context learning of compositional semantics based on our formal semantic geometry framework.

8 Limitations

1. Limitation of data source: this work only focused on explanatory sentences. Whether the semantic separability of other corpora emerges over latent space is not explored. **2.** Role-content clusters overlapping: the geometric analysis indicates that the role-content regions still have significant overlapping over distributional spaces. Therefore, a new potential task can be how we can better separate/disentangle the semantic features (role-content) to provide better localisation or composition behaviour over distributional semantic spaces in the Computational Linguistics domain, further assisting downstream tasks, such as Natural Language Reasoning, Compositional Generalisation, etc. **3.** Large Language Models: this paper only investigates the BERT-GPT2 architecture based on the current state-of-the-art language VAE (Optimus). The larger decoder is out of the scope of this work and needs to be investigated in the future.

Acknowledgements

We appreciate the reviewers for their insightful comments and suggestions. This work was partially funded by the EPSRC grant EP/T026995/1 entitled “EnnCore: End-to-End Conceptual Guarding of Neural Architectures” under Security for all in an AI enabled society, by the Swiss National Science Foundation (SNSF) project NeuMath (200021_204617), by the CRUK National Biomarker Centre, and supported by the Manchester Experimental Cancer Medicine Centre and the NIHR Manchester Biomedical Research Centre.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. 2019. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019.

Yoshua Bengio. 2013. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer.

Kaj Bostrom, Xinyu Zhao, Swarat Chaudhuri, and Greg Durrett. 2021. Flexible generation of natural language deductions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6266–6278, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Danilo S. Carvalho, Yingji Zhang, Giangiacomo Mercatali, and Andre Freitas. 2023. Learning disentangled representations for natural language definitions. *Findings of the European chapter of Association for Computational Linguistics (Findings of EACL)*.

Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. 2022. The geometry of multilingual language model representations. *arXiv preprint arXiv:2205.10964*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *arXiv preprint arXiv:2002.05867*.

Stephen Clark, Bob Coecke, and Mehrnoosh Sadrzadeh. 2008. A compositional distributional model of meaning. In *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*, pages 133–140. Oxford.

Stephen Clark and Stephen G. Pulman. 2007. Combining symbolic and distributional models of meaning. In *Quantum Interaction*.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees.

Donald Davidson. 1967. The logical form of action sentences.

David R Dowty, Robert Wall, and Stanley Peters. 2012. *Introduction to Montague semantics*, volume 11. Springer Science & Business Media.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2020. A disentangling invertible interpretation network for explaining latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Ghazi Felhi, Joseph Le Roux, and Djamé Seddah. 2022. Towards unsupervised content disentanglement in sentence representations via syntactic roles. *arXiv preprint arXiv:2206.11184*.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. 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 240–250, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiyan Fu and Anette Frank. 2024. Exploring continual learning of compositional generalization in nli. *arXiv preprint arXiv:2403.04400*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson H S Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2017. A deep semantic natural language processing platform.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. *arXiv preprint arXiv:2203.14680*.

Daniel Gildea and Daniel Jurafsky. 2000. Automatic labeling of semantic roles. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, page 512–520, USA. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022. A distributional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, Weihong Zhong, and Bing Qin. 2023. Controllable text generation via probability density estimation in the latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12590–12616, Toronto, Canada. Association for Computational Linguistics.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2023. Linearity of relation decoding in transformer language models. *arXiv preprint arXiv:2308.09124*.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. 2022. Fuse it more deeply! a variational transformer with layer-wise latent variable inference for text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 697–716, Seattle, United States. Association for Computational Linguistics.

Zhiting Hu and Li Erran Li. 2021. A causal lens for controllable text generation. *Advances in Neural Information Processing Systems*, 34:24941–24955.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. The platonic representation hypothesis. *arXiv preprint arXiv:2405.07987*.

Ray S Jackendoff. 1992. *Semantic structures*, volume 18. MIT press.

Peter Jansen, Rebecca Sharp, Mihai Surdeanu, and Peter Clark. 2017. Framing qa as building and ranking intersentence answer justifications. *Computational Linguistics*, 43(2):407–449.

Peter A Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton T Morrison. 2018. Worldtree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. *arXiv preprint arXiv:1802.03052*.

Giyoung Jeon, Haedong Jeong, and Jaesik Choi. 2019. An efficient explorative sampling considering the generative boundaries of deep generative neural networks.

Yibo Jiang, Bryon Aragam, and Victor Veitch. 2024a. Uncovering meanings of embeddings via partial orthogonality. *Advances in Neural Information Processing Systems*, 36.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. 2024b. On the origins of linear representations in large language models. *arXiv preprint arXiv:2403.03867*.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2019. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2018. Question answering as global reasoning over semantic abstractions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *arXiv preprint arXiv:1704.05572*.

Hyunjik Kim and Andriy Mnih. 2018. Disentangling by factorising. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2649–2658. PMLR.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes.

Robert B Lees. 1957. Syntactic structures.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Bohan Li, Junxian He, Graham Neubig, Taylor Berg-Kirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3603–3614, Hong Kong, China. Association for Computational Linguistics.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020b. Optimus: Organizing sentences via pre-trained modeling of a latent space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2022a. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*.

Zhuang Li, Lizhen Qu, Qiongkai Xu, Tongtong Wu, Tianyang Zhan, and Gholamreza Haffari. 2022b. Variational autoencoder with disentanglement priors for low-resource task-specific natural language generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10335–10356, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William W Cohen. 2020. Differentiable open-ended commonsense reasoning. *arXiv preprint arXiv:2010.14439*.

Guangyi Liu, Zeyu Feng, Yuan Gao, Zichao Yang, Xiaodan Liang, Junwei Bao, Xiaodong He, Shuguang Cui, Zhen Li, and Zhiting Hu. 2023a. Composable text controls in latent space with ODEs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16543–16570, Singapore. Association for Computational Linguistics.

Sheng Liu, Lei Xing, and James Zou. 2023b. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Yahui Liu, Enver Sangineto, Yajing Chen, Linchao Bao, Haoxian Zhang, Nicu Sebe, Bruno Lepri, Wei Wang, and Marco De Nadai. 2021. Smoothing the disentangled latent style space for unsupervised image-to-image translation.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

Giangiacomo Mercatali and André Freitas. 2021. Disentangling generative factors in natural language with discrete variational autoencoders. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3547–3556.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv:2305.16130*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

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

David Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878, Copenhagen, Denmark. Association for Computational Linguistics.

Melanie Mitchell. 2023. How do we know how smart ai systems are?

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *arXiv preprint arXiv:2309.00941*.

Martha Stone Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic role labeling*, volume 6. Morgan & Claypool Publishers.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2024. The geometry of categorical and hierarchical concepts in large language models.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Malka Rappaport Hovav and Beth Levin. 2008. The english dative alternation: The case for verb sensitivityl. *Journal of linguistics*, 44(1):129–167.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of bert. *Advances in Neural Information Processing Systems*, 32.

Tianxiao Shen, Jonas Mueller, Regina Barzilay, and Tommi Jaakkola. 2020. Educating text autoencoders: Latent representation guidance via denoising. In *International conference on machine learning*, pages 8719–8729. PMLR.

Paul Smolensky. 2006. Harmony in linguistic cognition. *Cognitive science*, 30(5):779–801.

Paul Smolensky and Géraldine Legendre. 2006. *The harmonic mind: From neural computation to optimality-theoretic grammar. Vol. 1, Cognitive architecture*. MIT.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Explainable inference over grounding-abstract chains for science questions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1–12.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2024. A differentiable integer linear programming solver for explanation-based natural language inference. *arXiv preprint arXiv:2404.02625*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, and Stefano Soatto. 2023. Linear spaces of meanings: compositional structures in vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15395–15404.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*.

Asahi Ushio, Luis Espinosa-Anke, Steven Schockaert, and Jose Camacho-Collados. 2021. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? *arXiv preprint arXiv:2105.04949*.

Marco Valentino, Mokanarangan Thayaparan, Deborah Ferreira, and André Freitas. 2022a. Hybrid autoregressive inference for scalable multi-hop explanation regeneration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11403–11411.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2020. Explainable natural language reasoning via conceptual unification. *arXiv preprint arXiv:2009.14539*.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2022b. Case-based abductive natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1556–1568.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Emile van Krieken, Thiviyan Thanapalasingam, Jakub Tomczak, Frank Van Harmelen, and Annette Ten Teije. 2023. A-nesi: A scalable approximate method for probabilistic neurosymbolic inference. *Advances in Neural Information Processing Systems*, 36:24586–24609.

Jake Vasilakes, Chrysoula Zerva, Makoto Miwa, and Sophia Ananiadou. 2022. Learning disentangled representations of negation and uncertainty. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8380–8397, Dublin, Ireland. Association for Computational Linguistics.

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

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Hitomi Yanaka, Koji Mineshima, and Kentaro Inui. 2021. SyGNS: A systematic generalization testbed based on natural language semantics. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 103–119, Online. Association for Computational Linguistics.

A Experiment Setting

Dataset. Table 8 displays the statistical information of the datasets used in the experiment. The data of the two datasets partially overlap, so only the unique explanations are selected as the experimental data. The rationale for choosing explanatory sentences is that they are designed for formal/localised/symbolic semantic inference task in natural language form, which provides a semantically complex and yet controlled experimental setting, containing a both well-scoped and diverse set of target "concepts" and sentence structures, providing a semantically challenging yet sufficiently well-scoped scenario to evaluate the syntactic and semantic organisation of the space. Besides, those concepts mentioned in the corpus, such as *animal is a kind of living thing*, are fundamental to human semantic understanding.

Corpus	Num data.	Avg. length
WorldTree (Jansen et al., 2018)	11430	8.65
EntailmentBank (Dalvi et al., 2021)	5134	10.35

Table 8: Statistics from explanations datasets.

Table 9 illustrates the semantic, structure, and topic information of explanatory sentences over the latent space. The explanatory sentences are automatically annotated using the semantic role labelling (SRL) tool, which can be implemented via AllenNLP library (Gardner et al., 2017). We report in Table 10 the semantic roles from the explanations corpus.

Architecture. Figure 8 provides a visual representation of the connection between BERT and GPT2 within the AutoEncoder architecture.

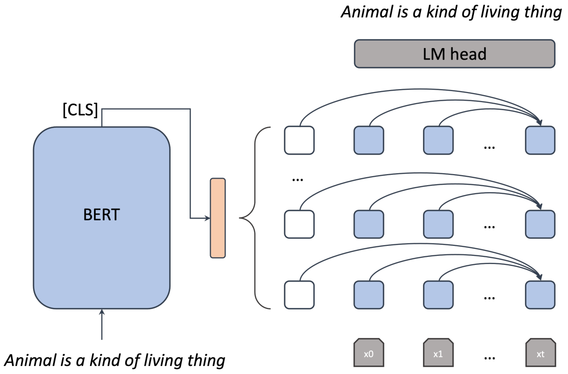

Animal is a kind of living thing
LM head
[CLS]
BERT
...
x0
x1
xt
Animal is a kind of living thing

Figure 8: Latent sentence injection.

To train the CVAE, we use a new embedding layer for semantic roles and separate MLP layers W_{μ}^{srl} and W_{σ}^{srl} to learn prior distribution.

Hyperparameters. The training process of the decision tree binary classifier can be implemented via scikit-learn packages with default hyperparameters. As for Optimus, the latent space size is 32 in the experiment. The training details are following the original experiment from Optimus (Li et al., 2020b).

B Further Experimental Results

Traversal visualisation. PCA plots for ARG0, ARG1, and PRED are provided in Figure 9.

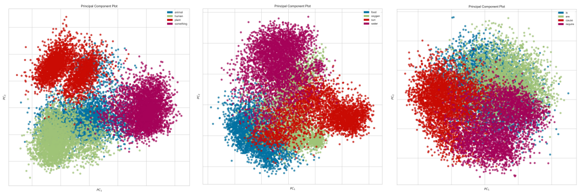


Figure 9: PCA visualisation.

In addition, we also provide the visualisation of word content *animal* with different semantic roles: ARG0, ARG1, ARG2, in Figure 10. From it, we can observe that the same content with different semantic roles can also be clustered and separated in latent space.

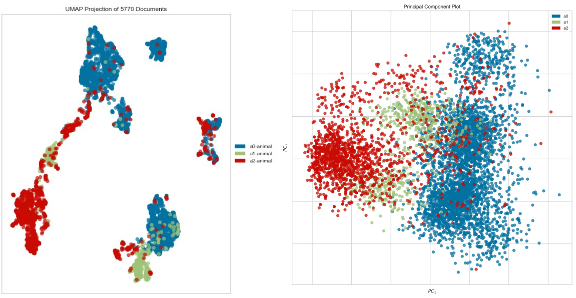


Figure 10: Visualisation for *animal-ARG0,1,2*.

Qualitative evaluation for arithmetic. Table 11 lists the traversed explanations after addition (blue) and subtraction (red) on different semantic role information. We can observe that the resulting sentences after addition can hold the same role-content as inputs, revealing latent space geometry.

Quantitative evaluation for arithmetic. Quantitative evaluation for our hypotheses via latent arithmetic. Both VERB and Object can perform high

Cluster	Theme and Pattern
0	Theme: physics and chemistry. Pattern: *if then* and *as*. E.g., if a substance is mixed with another substance then those substances will undergo physical change.
1	Theme: country, astronomy, and weather. E.g., new york state is on earth
2	Theme: physics and chemistry. Pattern: *is a kind of*. E.g., light is a kind of wave.
3	Theme: biology. E.g., a mother births offspring.
4	Theme: synonym for verb. Pattern: *means* and *is similar to*. E.g., to report means to show.
5	Theme: astronomy. E.g., the solar system contains asteroids.
6	Theme: animal/plant. Pattern: *is a kind of*. E.g., a seed is a part of a plant.
7	Theme: item. E.g., a telephone is a kind of electrical device for communication.
8	Theme: synonym for life. Pattern: *means* and *is similar to*. E.g., shape is a kind of characteristic.
9	Theme: geography. Pattern: *is a kind of*. E.g., a mountain is a kind of environment.
10	Theme: animal and plant. Pattern: *if then* and *as*. E.g., if a habitat is removed then that habitat is destroyed.
11	Theme: scientific knowledge. Pattern: *(;)*, *number* and */*. E.g., freezing point is a property of a (substance ; material).
12	Theme: item. Pattern: *is a kind of object*. E.g., a paper is a kind of object.
13	Theme: chemistry and astronomy. E.g., oxygen gas is made of only oxygen element.
14	Theme: general about science. Pattern: *(;)*. E.g., seed dispersal has a positive impact on (a plant ; a plant 's reproduction).
15	Theme: item. Pattern: *is a kind of*. E.g., fertilizer is a kind of substance.
16	Theme: physics and chemistry. Pattern: *(;)*. E.g., the melting point of oxygen is -3618f ; -2188c ; 544k.
17	Theme: animal. E.g., squirrels live in forests.
18	Theme: nature. E.g., warm ocean currents move to cooler ocean regions by convection.
19	Theme: life. E.g., pond water contains microscopic living organisms.

Table 9: Cluster Information for explanatory sentences, we use a k-means classifier to classify the sentence representations and manually evaluate each class.

Semantic Tags	Prop. %	Description and Example
ARGM-DIR	0.80	Directionals. E.g. all waves transmit energy **from one place to another**
ARGM-PNC	0.08	Purpose. E.g. many animals blend in with their environment **to not be seen by predators**
ARGM-CAU	0.05	Cause. E.g. cold environments sometimes are white in color **from being covered in snow**
ARGM-PRP	1.30	Purpose. E.g. a pot is made of metal **for cooking**
ARGM-EXT	0.04	Extent. E.g. as the amount of oxygen exposed to a fire increases the fire will burn **longer**
ARGM-LOC	4.50	Location. E.g. a solute can be dissolved **in a solvent** when they are combined
ARGM-MNR	2.00	Manner. E.g. fast means **quickly**
ARGM-MOD	9.80	Modal verbs. E.g. atom **can** not be divided into smaller substances
ARGM-DIS	0.07	Discourse. E.g. if something required by an organism is depleted **then** that organism must replenish that something
ARGM-GOL	0.20	Goal. E.g. We flew **to Chicago**
ARGM-NEG	1.20	Negation. E.g. cactus wrens building nests in cholla cacti does **not** harm the cholla cacti
ARGM-ADV	6.70	Adverbials
ARGM-PRD	0.20	Markers of secondary predication. E.g.
ARGM-TMP	7.00	Temporals. E.g. a predator **usually** kills its prey to eat it
O	-	Empty tag.
V	100	Verb.
ARG0	32.0	Agent or Causer. E.g. **rabbits** eat plants
ARG1	98.5	Patient or Theme. E.g. rabbits eat **plants**
ARG2	60.9	indirect object / beneficiary / instrument / attribute / end state. E.g. animals are **organisms**
ARG3	0.60	start point / beneficiary / instrument / attribute. E.g. sleeping bags are designed **to keep people warm**
ARG4	0.10	end point. E.g. when water falls from the sky that water usually returns **to the soil**

Table 10: Semantic Role Labels that appears in explanations corpus.

ratio after addition, indicating role-content separability.

ADD and SUB arithmetic

ARGUMENT1:
a needle is a kind of object
a tire is a kind of object

a wire is a kind of object
a stick is a kind of object
a ball is a kind of object

a serotype is similar to intersex egg
a zygote contains many cell types
an xylem is made of two clumps

VERB:
chromosomes are located in the cells
Australia is located in the southern hemisphere

stars are located in the solar system
Jupiter is located in the milky way galaxy
aurora is located in the constellation of Leo

a crystal is made of metal
an alloy is made of iron and zinc
an aluminum plug is nonmagnetic

LOCATION:
volcanoes are often found under oceans
mosquitos can sense carbon dioxide in the air

polar ice sheets are located along rivers
hurricanes occur frequently along the coast in Africa
tide waves cause flooding in coastal waters

valley is a kind of location
shape is a property of rocks
desert is a kind of place

TEMPORAL:
as the population of prey decreases competition between predators will increase
as competition for resources decreases the ability to compete for resources will increase

as the population of an environment decreases ecosystem function will decrease
as the spread of available air mass increases the population will increase
as the number of heavy traffic required increases the traffic cycle will decrease

some types of lizards live in water
a rose is rich in potassium
a fern grass roots foot trait means a fern grass

NEGATION:
pluto has not cleared its orbit
sound can not travel through a vacuum

radio waves don't have electric charge
electromagnetic radiation does not have a neutral electric charge
electromagnetic radiation contains no electric charge

Mars is a kind of moon / planet
Anothermic rock is a kind of metamorphic rock
Anal Cetus's skeleton is a kind of fossil

Table 11: Latent sapce arithmetic for five semantic tags (blue: addition, red: subtraction).

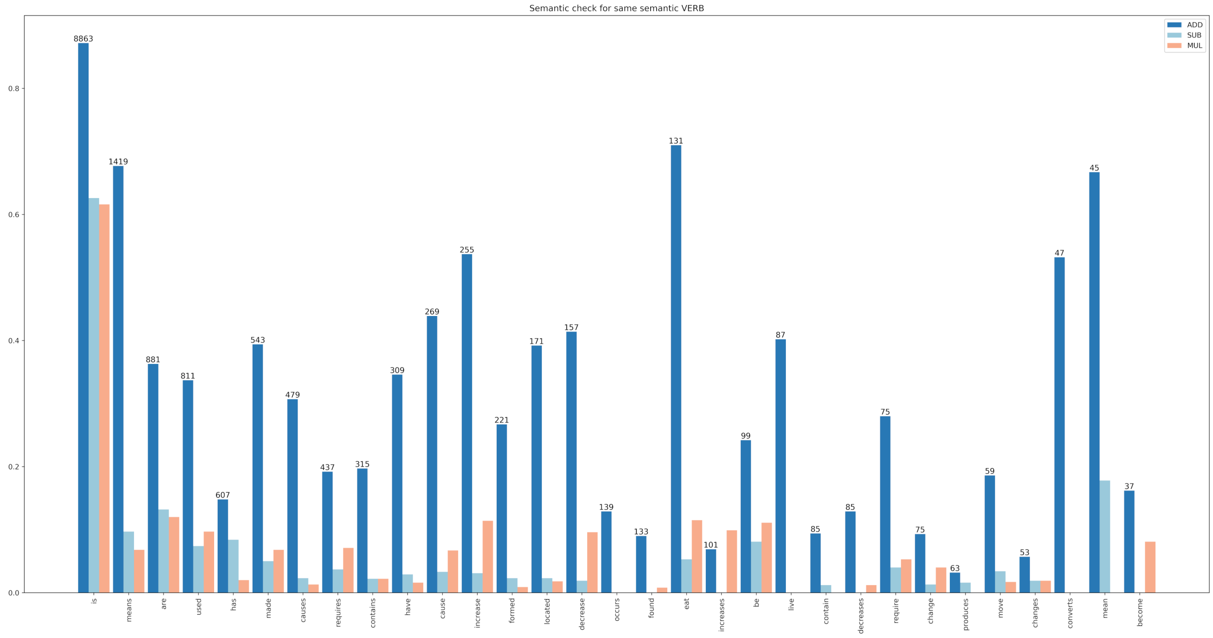


Figure 11: Predicate (VERB). The content *is* shows the high ratio after subtraction, indicating that the *V-is* is widely distributed over the latent space.

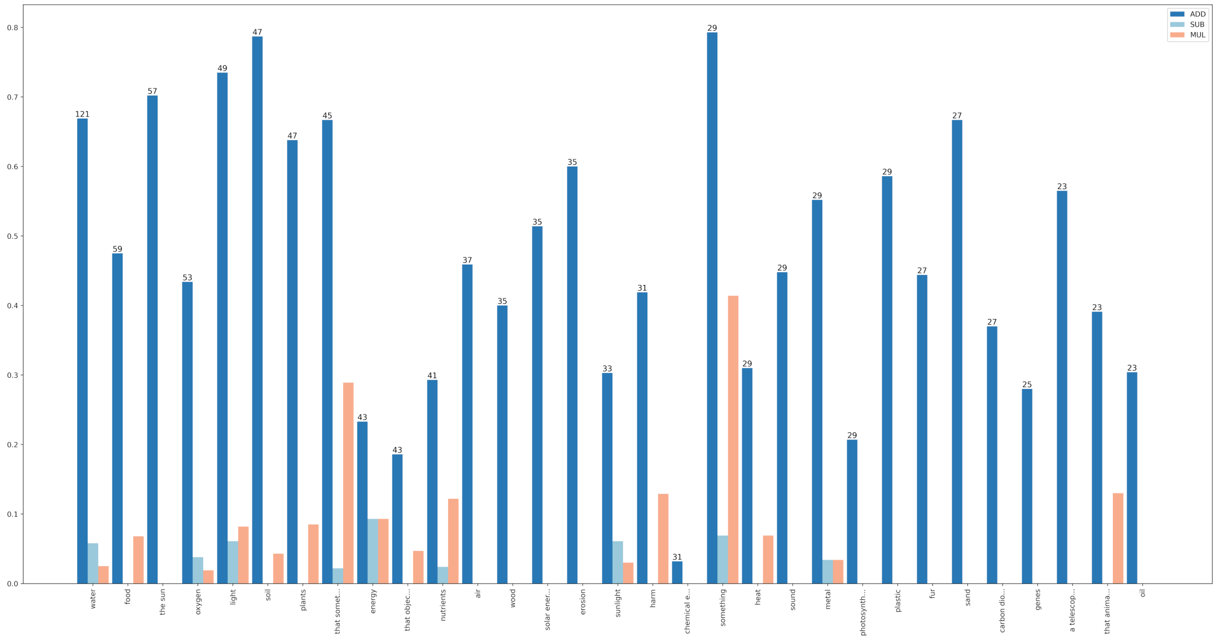


Figure 12: Object (ARG1).

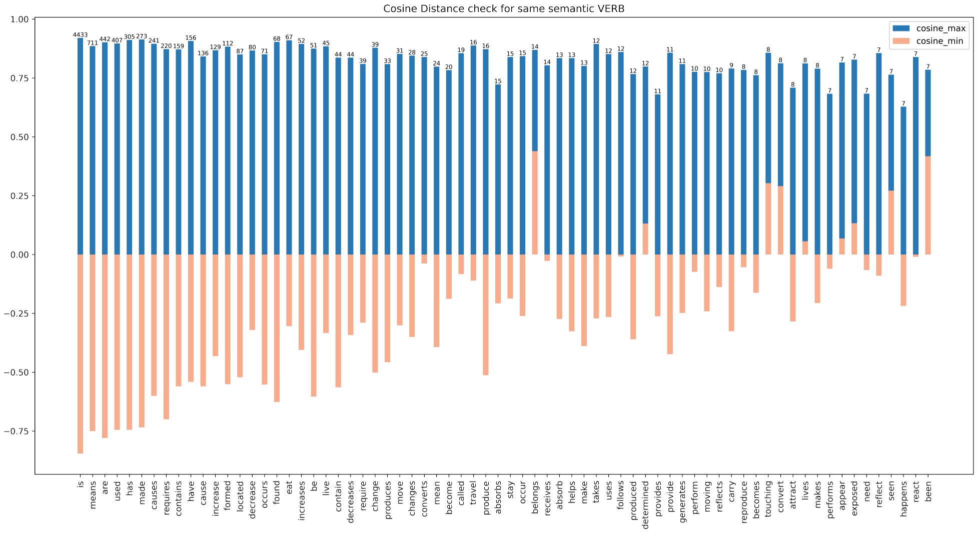


Figure 13: Cosine distance of sentence pairs in VERB-content clusters.

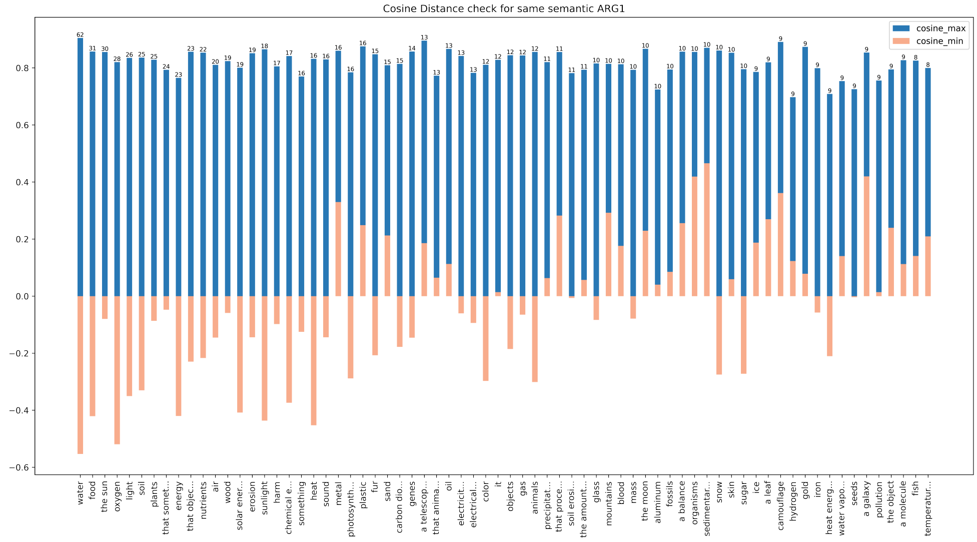


Figure 14: Cosine distance of sentence pairs in ARG1-content clusters.

LawToken: a single token worth more than its constituents

Yu-Hsiang Tseng [1], Hsin-Yu Chou [2], Shu-Kai Hsieh [2]
[1]Department of Linguistics, University of Tübingen
[2] Graduate Institute of Linguistics, National Taiwan University

Abstract

Legal citations require correctly recalling the law references of complex law article names and article numbering, which large language models typically treat as multi-token sequences. Motivated by the form-meaning pair of constructionist approaches, we explore treating these multi-token law references as a single holistic law token and examining the implications for legal citation accuracy and differences in model interpretability. We train and compare two types of models: LawToken models, which encode the legal citations as a single law token, and LawBase models, which treat them as multi-token compounds. The results show that LawToken models outperform LawBase models on legal citation tasks, primarily due to fewer errors in the article numbering components. Further model representation analysis reveals that, while both models achieve comparable semantic representation quality, the multi-token-based LawBase suffers from degraded representations in multistep decoding, leading to more errors. Taken together, these findings suggest that form-meaning pairing can operate in a larger context, and this larger unit may offer advantages in future modeling of legal reasoning. In practice, this approach can significantly reduce the likelihood of hallucinations by anchoring legal citations as discrete, holistic tokens, thereby minimizing the risk of generating nonexistent or incorrect legal references.

1 Introduction

Recalling the correct legal citations, e.g., the law articles, regulations, or precedents, poses a great challenge to the large language models and raises an interesting question to computational linguistics (Guha et al., 2024; Dahl et al., 2024). While the autoregressive models are so adept at working with legal texts in certain, but not all, scenarios and tasks (Katz et al., 2024; Rodgers et al., 2023), generating the correct without producing non-existent articles or hallucinating remains a challenge to the modern models (Weiser, 2023; Henderson et al., 2023). While finding efficient ways to train LLMs adept at legal citations may potentially be addressed in future models, the linguistic intrigues nevertheless persist regarding how models encode the explicit textual forms and their impacts on the model's representations.

In current large language models (Dubey et al., 2024; Yang et al., 2024; Achiam et al., 2023), these legal citations are treated as normal texts: processed by the tokenizer, they are chunked into a sequence of tokens. For example, the legal citation form in Taiwan generally is the article name followed by the article and paragraph numbers, such as "Road traffic safety regulations, Article 94, Paragraph 3." The model needs to learn how the multi-token sequence is related to the intended meanings in context.

The intended meaning of a cited law reference may entail the following three layers, in the order of their context-dependence: (1) the compositional meaning from the tokenized components, which, for instance, are the composite meanings of road traffic, safety, and others (Bell and Schäfer, 2016; Cordeiro et al., 2019; Alipoor and Schulte im Walde, 2020); (2) the semantic extensions of the legal text content, specifying the legal obligation of the driver (Tseng et al., 2023; Noraset et al., 2017; Mickus et al., 2019); and (3) the pragmatical usage of the law in the court verdict when determining the liability (Ruis et al., 2023; Louis et al., 2020; Parrish et al., 2021). In practice, the large language models might be good at deriving pragmatics and resolving the intended sense of the ambiguous words (tokens) from the constituting lexical semantics; but, in contrast, the hallucination (Guha et al., 2024; Bommasani et al., 2023; Dahl et al., 2024) suggests the model may struggle with decoding back from the context-specific pragmatic to the underlying constituent tokens.

An alternative approach is to map between the

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 30–46
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

layers *as direct as possible*; that is, treating law references as a single holistic form-meaning pair, where the entire law citation – including the law names and article or paragraph numbering – is recognized as one *law token*. These additional law tokens are motivated by the constructionist approach (Goldberg, 2024; Lakoff, 1987; Bybee, 2010). As linguistic units, from single words to multi-word idioms, function as form-meaning pairs, there is no theoretical limit on their scope except for cognitive constraints. However, computationally, large language models may already have enough capacity to capture the complex form-meaning mapping, provided they have clear cue-meaning mappings from tokenization.

This paper aims to empirically study the effect of tokenization on legal citations, focusing on both task performance and how tokenization affects the model's prediction probabilities and representation. Using the court verdicts of Taiwan, we compile a LawToken dataset containing 675M tokens. The dataset is used to fine-tune two types of models: LawBase models, which use the unmodified tokenizer, and LawToken models, which use an augmented tokenizer that includes frequently-used *law references* as new *law tokens*. When referring generically to using law tokens or references in the texts, we use the term *legal citation*. We first establish that LawToken models outperform LawBase models in legal citation tasks, and we next further analyze model representations, revealing that the performance difference may stem from the degraded contextualized representation during the multistep decoding in LawBase models.

This paper is organized as follows. After briefly summarizing the related works in Section 2, Section 3 describes the preprocessing steps, dataset, training, and evaluation of LawToken and LawBase models. Section 4 examines the model representations and explores how they differ in the two models. Section 5 concludes the paper.

2 Related Works

A legal reference, consisting of law or act names and article numbers, is composed of multiple tokens, which the language model has to learn to determine the intended meaning of the multi-token compound. However, past literature suggests that the compound meaning is not always transparent in terms of its constituents. Some are semantically transparent, such as "swimming pool," where the compound meaning is directly composite of the constituents; some are opaque, such as "hot dog." However, even a seemingly transparent compound may be challenging to pinpoint the relationships between its constituents; for instance, "airport" and "airplane" (compounds written without spaces), the role of "air" may be unexpectedly complicated (Bell and Schäfer, 2016; Reddy et al., 2011; Zwitserlood, 2014). Modeling the semantic transparency of compounds remains difficult, even when using static or contextualized semantic vectors (Shwartz and Dagan, 2019; Miletić and im Walde, 2023).

Some multi-token(word) expressions are not usually considered compounds but nevertheless convey meanings more than their parts. For example, "hazard a guess," or more idiom-like expression, "I hope this mail finds you well." These expressions, gaining their meaning through repeated uses by the language community and, therefore, form a static form-meaning pair, are *constructions* (Goldberg, 2013).

Along this line of reasoning, the law references can act as a construction. However, if the law reference is an opaque multi-token expression, the LLMs should already handle them to some extent (Goldberg, 2024). Yet, a previous study argued that the LLM's task performances are form-dependent (Ohmer et al., 2024), indicating that the models rely more on the surface form rather than the underlying meaning to complete the task. Consequently, even though the law reference is a construction, the way they are tokenized can significantly influence the model's task behavior.

Tokenizing law reference as a single law token has implications beyond linguistic theory. Using law tokens implies the model operates with a fixed set of "law vocabulary," which prevents the model from producing nonexistent law articles (Guha et al., 2024; Dahl et al., 2024). Although specialized legal-domain LLMs have become more prevalent, they are fine-tuned or continuously pretrained on legal texts or using retrieval-augmented generation without changing tokenization specifically for legal references (Colombo et al., 2024; Wiratunga et al., 2024; Lee, 2023; Cui et al., 2023). Furthermore, from an information-theoretic perspective, tokenization is the pre-compression in the LLM (Deletang et al., 2024). It is therefore interesting to observe how using a law token will change the compression behavior.

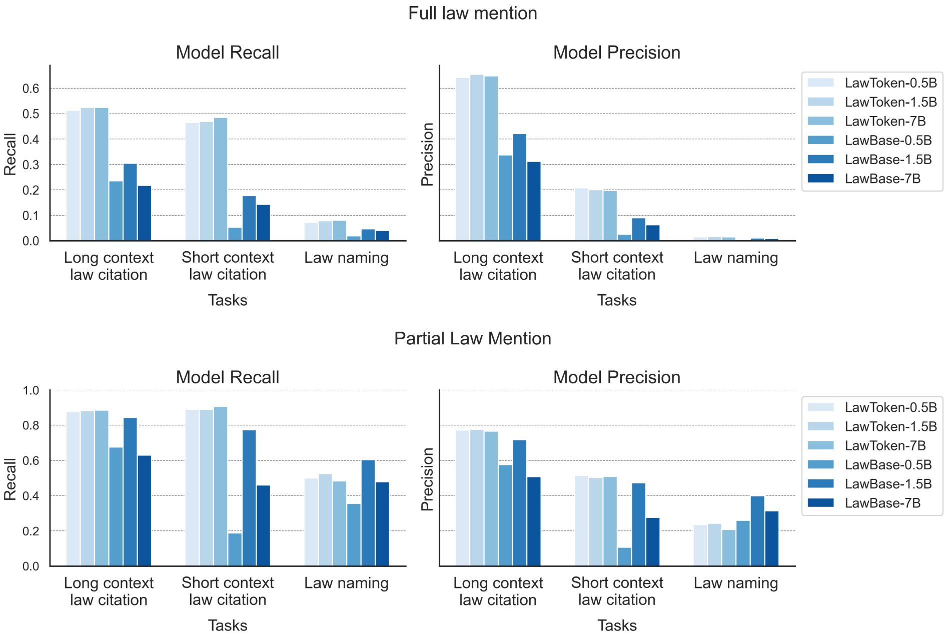


Figure 1: Evaluation results using full law mentions (upper panel) and partial law mentions (lower panel). The performances are evaluated using recall and precision, where recall is the proportion of correctly predicted law tokens among all true tokens, and precision is the proportion of correctly predicted tokens among all predicted ones.

3 LawToken & LawBase models

3.1 Dataset

The LawToken Datasets[1] consist of legal documents publicly available in Taiwan, encompassing both law articles and court verdicts. The dataset has three parts. The first and second parts, composed of court verdicts and law articles, respectively, standardize law references in natural language by representing them in the following format: <*LAW_NAME*|*ARTICLE_NUMBER*>. For instance, a reference to 道路交通安全規則第94條第3項“road traffic safety regulations, Article 94, Paragraph 3.” is transformed into the format <道路交通安全規則|94|3>. Conversely, the third group, derived solely from the court verdicts, employs a different transformation: legal references are removed from their original positions in the main text and then appended at the end of each court verdict, enclosed between a start-of-citation marker “<cite>” and an end-of-citation marker “</cite>.” Examples of each group are provided in the Appendix. The three groups are combined and randomly shuffled. Subsequently, a train and test split is generated at a ratio of 9:1, resulting in a training set with 545.4k instances and a testing set with 60.6k instances.

3.2 Model Training

The three base models employed in this paper are Qwen2 of sizes 0.5B, 1.5B, and 7B [2]. We select the frequently occurred law references, namely, the total frequencies of the law references in the court verdicts need to be higher than 100 times, resulting in 13,083 law tokens. Subsequently, we train LawToken models with the high-frequency law tokens added into the tokenizer. The integration of the law tokens into the tokenizer enables the models to recognize the law references as single tokens and learn the contexts in which they are referenced. On the other hand, the LawBase uses the unmodified tokenizer. In other words, the mentions of law references in natural languages are represented as single tokens in LawToken models, whereas in the baseline LawBase models, they are interpreted as multi-token sequences.

Overall, six models are trained [3]. The fine-tuning uses 4 nVIDIA H100s and takes around 30 hours for all models. The evaluation cross-entropy losses of the LawToken models are .86, .79, and .69 for 0.5B, 1.5B, and 7.0B model sizes, respectively, and they are .82, .76, and .65 for the LawBase models. The evaluation loss decreases as the model

---

[1] https://huggingface.co/datasets/amy011872/LawToken.

[2] Models obtained from https://huggingface.co/Qwen

[3] All six models are available on HuggingFace, for instance, the 7B finetuned model could be found at https://huggingface.co/amy011872/LawToken-7B-a2.

size increases, whereas LawBase model losses are consistently lower than those of LawToken models.

3.3 Evaluation

The evaluation tasks include a long-context law citation task, a short-context law citation task, and a law naming task. These tasks, derived from the testing set, involve the same objective: predicting relevant LawTokens based on the provided context, with "<cite>" serving as the special token for prediction.

In the long-context law citation task, the model is provided with the full context of court verdicts, with law references removed, and is asked to predict the relevant legal citations. Conversely, the short-context law citation focuses on a more localized context, where sentences containing legal citations are identified, and the model is provided with only the preceding sentence as context to predict the relevant citations. The law naming task, on the other hand, is derived from law articles. Here, the model is presented solely with the content from a certain law article and is required to predict the correct law name and article number in the standardized format. Examples of each evaluation task are included in Appendix.

Figure 1 presents the recall and precision of the six models across three different tasks. The upper panel indicates the measures evaluated using full law mentions, where a prediction is counted as correct only if both the law name and article number match the ground truth. The results show that the LawToken models consistently outperform the LawBase ones, regardless of the tasks and model sizes. These patterns may suggest that LawToken encodes better representations of law mentions, or simply sidesteps the challenge of predicting article number, which the LawBase model often struggles with. To investigate, we re-evaluate using partial law mentions, where the predictions are considered correct when the law name alone matches with the true ones. The results are shown in the lower panel of Figure 1. Again, LawToken still outperforms LawBase in most cases, although the performance gap narrows, especially with LawToken 1.5B, and in the law naming task. In addition, out of all the unique law reference predictions produced by the LawBase models, 6.6% of them do not exist in those generated by the 0.5B model, 8.2% by the 1.5B model, and 7.6% by the 7B models. That is, the LawBase models still experience hallucinations after being fine-tuned explicitly in the current dataset.

The patterns in Figure 1 further reveal three notable observations: (1) Task difficulties vary with the richness of pragmatic context: the more context a task provides, the better the model performs. This effect is particularly evident in precision scores, where both models achieve the highest precisions in the long context task and the worst in the law naming task, where only the legal text content is available, with no additional pragmatic context. (2) Pragmatic context helps the LawBase model predict law names but not article numbers. This is shown in the partial law mention evaluations, where the LawBase's performance closes in on that of LawToken. This pattern is consistent with the fact that both LawBase and LawToken are trained on the same data, and the law names are lexical tokens that LawBase can learn their contextual usages during fine-tuning. By contrast, article numbers are highly ambiguous tokens reused across different law mentions and LawBase, having no specialized tokenization, struggles to disambiguate them. This is where LawToken has an advantage. (3) We also observed that 1.5B model size in the LawBase family is the best-performing one in both full and partial law mentions across the board. This suggests that, given the moderate size of our fine-tuning data (675M tokens), 1.5B may represent the optimal model size under data constraints, assuming no changes to the tokenization.

Finally, to further compare the task performance of LawToken to other models, we randomly sampled 1,000 instances from each evaluation task to assess the performance of one of the commercial models (Achiam et al., 2023) (GPT-4o-mini). The generation method employs the batch API, with greedy decoding (temperature set to 0) and model specified to "GPT-4o-mini-2024-07-18". We use one-shot prompt design for GPT-4o-mini to understand the task better and produce the answer in the same format of LawTokens. The prompt example is provided in the Appendix.

The results are presented in Table 1. Overall, GPT-4o-mini does not perform at a level comparable to LawToken models. While we find that GPT-4o-mini is quite competitive when provided with ample contextual information, for example, in the long-context law citation task, nearly matching the performance of the fine-tuned LawBase models, its effectiveness diminishes significantly in tasks with limited context, such as the short-context law citation task and the law naming task. The compar-

Model	Long		Short		Naming	
	R	P	R	P	R	P
LawTok-0.5B	0.54	0.65	0.46	**0.25**	0.08	0.02
LawTok-1.5B	**0.55**	**0.67**	0.44	0.22	0.08	0.02
LawTok-7.0B	0.53	0.65	**0.46**	0.22	**0.09**	**0.02**
LawBas-0.5B	0.23	0.33	0.06	0.03	0.02	0.01
LawBas-1.5B	0.31	0.42	0.20	0.11	0.05	0.01
LawBas-7.0B	0.21	0.30	0.18	0.09	0.05	0.01
GPT-4o-mini	0.28	0.41	0.03	0.02	0.01	0.01

Table 1: Comparison of recalls and precisions in different models in the 1000-dataset.

ison crucially demonstrates that the tasks cannot be solved solely by superficial textual cues included in the context, which the GPT-4o models will take advantage of.

Taken together, these results show LawToken models consistently outperform LawBase models. Moreover, the comparison between full and partial law mention evaluations suggests the crucial differences stem from how the model handles law names versus article numbers. To better understand the model representations of the law tokens and their law names and article number constituents, we next examine the representational differences between LawToken and LawBase models.

4 Examining model representations

While both model types show competitive results across the three legal tasks, LawToken consistently outperforms the LawBase models, with the only difference between the two being tokenization. This raises the question of what underlies this difference. On the one hand, the better performance of LawToken seems counterintuitive, as it uses fewer tokens to represent the legal mentions, thus fewer "buffering tokens" when decoding (Goyal et al., 2024; Herel and Mikolov, 2024). On the other hand, retrieving a legal mention is arguably distinct from reasoning; thus, LawToken may benefit from using an explicit, holistic token, allowing it to escape the complex structure within the legal mention comprising long compounds of act names and highly ambiguous article numbers.

In what follows, we investigate why the LawToken and LawBase models behave differently in the task. First, we demonstrate that the input embeddings learned by LawToken models reflect a general structure. Next, we examine the type-level representation similarities by comparing the model (hidden) states at different layer depths to the embeddings of the law's textual content. Finally, we analyze the token-level prediction probability as an index of how difficult the model finds certain tokens. These analyses provide further insight into the underpinnings of the models' performance differences.

4.1 Input embeddings

Figure 2 shows the visualization of the law tokens' input embeddings of the top 3 common laws extracted from the LawToken model. Each point in the panel represents a law token; for example, articles number 330 and 107 in the Code of Criminal Procedure are coded as two green dots. We use linear discriminant analysis to show how law tokens of different laws can be separable by a linear hyperplane. The underlying rationale is that law tokens coming from different laws should already reflect different usage patterns. Indeed, all classification accuracies are above the random chance level, while the 7B model is the worst of the three.

However, while classifying for law names is a simple and intuitive method to explore the embedding structure, it is not ideal. Law tokens of the same law may not necessarily be more similar than those of different ones. To better gauge the semantic representation of the law tokens and the law references, we next examine the text embeddings of the legal text content.

4.2 Type-level representation similarity

To better independently assess the quality of semantic representation encoded by the LawToken and LawBase models, we obtain the text embeddings of legal text content [4] with the commercial embedding models [5]. These embeddings are compared to the model's hidden states in various layer depths when encoding the selected sentences in the test split. A total of 13,215 sentences were selected, which included 2,211 unique legal citations. These sentences were selected to better evaluate the effect on the surrounding contexts, where there is only one law token or reference occurring before or after the 100-character window. We compute the centered kernel alignment scores (CKA; Kornblith et al., 2019) to measure the similarity between the model-encoded representation and the embedding of legal text content, where a higher score indicates

---

[4] For example, the text embedding for law token `<Labor Standards Act|43>` is the vector representation of the legal text content: "Workers may request leave for reasons such as marriage, [...]" (texts were in Taiwanese Mandarin.)

[5] Open AI's text-embedding-3-large

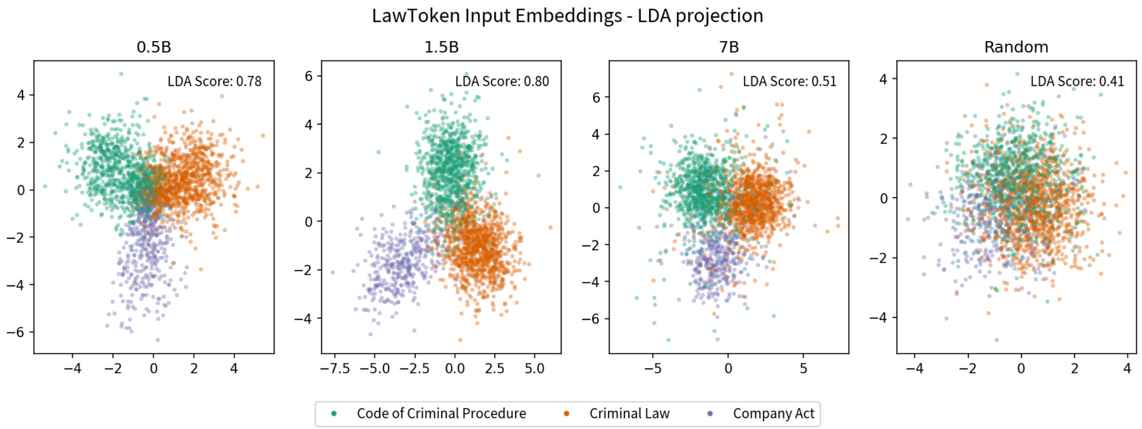


Figure 2: The input embeddings of the LawToken models, color-coded with the law article names: Code of Criminal Procedure (刑事訴訟法), Criminal Law (中華民國刑法), and Company Act (公司法). Only three laws are included for better visualization. The random Gaussian embeddings (Random) are shown as a baseline.

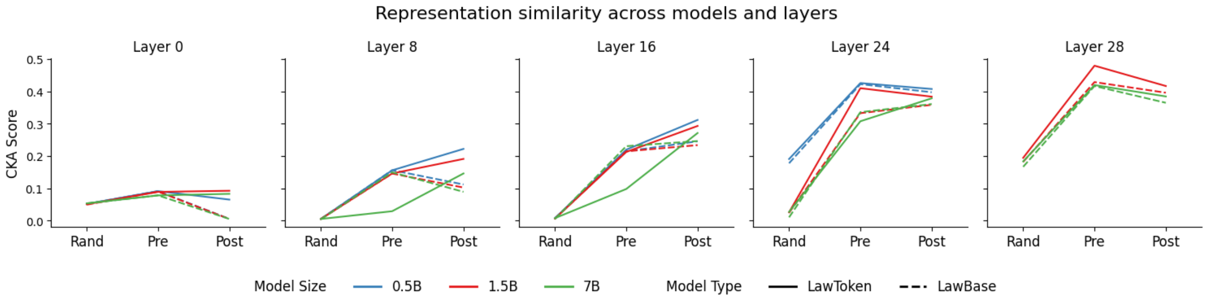
a better correspondence between two representations.

However, caveats remain when using such text embeddings. The legal text content is the semantic extension of a legal citation – what it normatively refers to – whereas the model encodes how a legal token or reference is functionally used in the legal texts. They are inevitably different. In addition, LawToken and LawBase both encode the usage in the context, meaning that each law token occurrence induces a different model state, while the legal text embedding stays the same. Therefore, although we use legal text embeddings as a reference for semantic representation, they are only an operationalization of the law token's meaning.

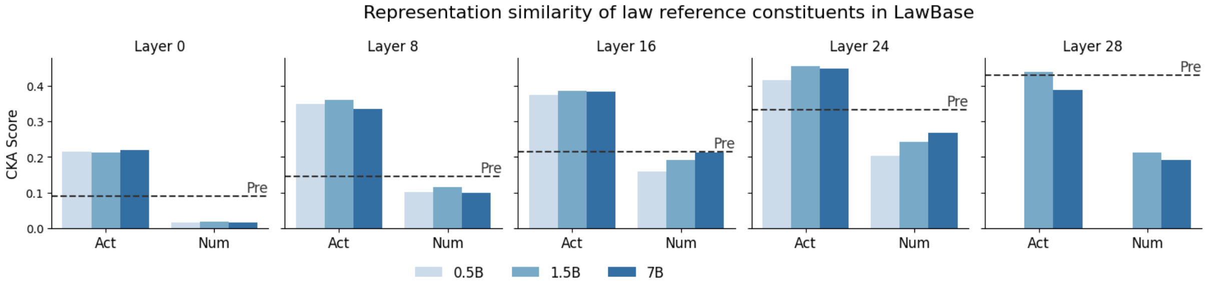
Figure 3a shows in each panel the results of representational similarities from the input layer (Layer 0) to the last layer of 0.5B model (Layer 24) or of 1.5B and 7B model (Layer 28). Each panel also shows three sites of interest. The `Rand` site denotes a random location before the target law token or reference, the `Pre` site is one token just before the target law token or the law reference, and the `Post` site is the token at the end of the target, which is the law token itself in the LawToken model and the last token of the law reference. Put in a more functional perspective, the `Rand` site provides a baseline estimate of the similarity possible to achieve only with the preceding context; the `Pre` site sheds light on the model states at which the model is about to predict the target law token or the first token in the law reference; and the `Post` site is when the models take into account of the law token or the law reference itself.

As shown in Figure 3a, the representation similarities increase throughout the layers and deeper into the sentence context. At the early layers of 0, 8, and 16, the `Rand` site scores are close to zero, reflecting that there is only very local information at this stage, and they do not correlate well with the law semantics. In contrast, the `Pre` sites are more indicative of the law content, potentially because the immediate pre-context of the target law token and reference is already informative enough about the legal mentions. Interestingly, the `Post` sites start to show diverging patterns between the representation of LawToken and LawBase, where the scores from LawToken are consistently higher than those from LawBase. The pattern effectively demonstrates the effects of tokenizing legal mentions as a whole in the LawToken model, showing that the embeddings of the law tokens carry rich lexical information.

However, this advantage is not irreplaceable. As we move into the deeper layers of 24 and 28, the contextual effect is more pronounced. The diverging trends observed in the earlier layers are closing in on Layer 24, especially for the 0.5B model, which is the last layer, and on Layer 28, where all models' scores are similar. Nevertheless, in the last layers, the `Pre` sites have higher scores than the `Post` sites, which hints at three potential explanations: (1) the model's hidden states at `Pre` site should be the most indicative for the legal references, as they are ones used to generate final token logits. (2) The scores may inevitably decrease after the `Pre` site, as the models shift from focusing on the legal reference to predicting the subsequent context. (3) Alternatively, the drop may potentially be a consequence of the internal structure of the



(a) Representation similarity scores across different sites. **Rand**: random location before the target law token or law reference; **Pre**: the token before the target; **Post**: the last token of the target, which is the law token itself and the last token of the law reference. Higher CKA scores indicate better alignment of the vectors with the law's semantics extensions.



(b) The representation similarities of the two constituents. **Act** refers to the name of the law article, and **Num** refers to the article number. As a visual reference, the dashed lines indicate the values of the Pre site of 1.5B LawBase model.

Figure 3: Representation similarities in different layers and different sites across model type and sizes.

legal references.

To instantiate the impact of the internal structure of the legal reference, we compute their representation similarity scores on `Act` and `Num` sites. The `Act` and `Num` sites, applying only to the LawBase models, are two constituents in the law references: the former being the last token of the act name and the latter the last token of article numbering. Each panel clearly shows that while act name representations contribute more as we move from `Pre` site to `Act` site, especially in the early layers, the `Num` sites consistently reduce the scores. This suggests the numbering constituents of the law references are less informative than the article numbering or even the preceding context. In fact, incorporating the article numbering seems to negatively impact the representation of the law references.

Representation similarities show the (mis-)alignments with law content semantics across different model layers and different sites, but they nevertheless only offer a coarse-grained view of the individual context each law token or reference is embedded. Being a context-independent measure of semantic extension, law content semantics is only based on the law content and has no access to the context information encoded by the LawBase or LawToken model. It is very well possible that the misalignment we observed, for example, the reduced similarity scores of the Article numbering site, is because the model has captured the context information that is not encoded in the static law content semantics. Therefore, we move to token-level probabilities to investigate the model's behavior further.

4.3 Token-level probability

The token-level probability provides complementary information for evaluating model behaviors. Distinct from the representation similarities where the token-based model states are compared to a type-based law content semantics, the prediction probabilities (of the true targets) are computed and evaluated in their context. There are two advantages of such a measure. (1) The prediction probabilities come directly from the model states of the hidden layer after accounting for all the other possible candidates. It effectively measures how good or close the last hidden states are to the true embeddings in that context. (2) The prediction probabilities also have explicit interpretations, which are surprisals as used in psycholinguistics studies (Goodkind and Bicknell, 2018; Wilcox et al., 2020), and information content or the compressed message length in bits if the law token or reference

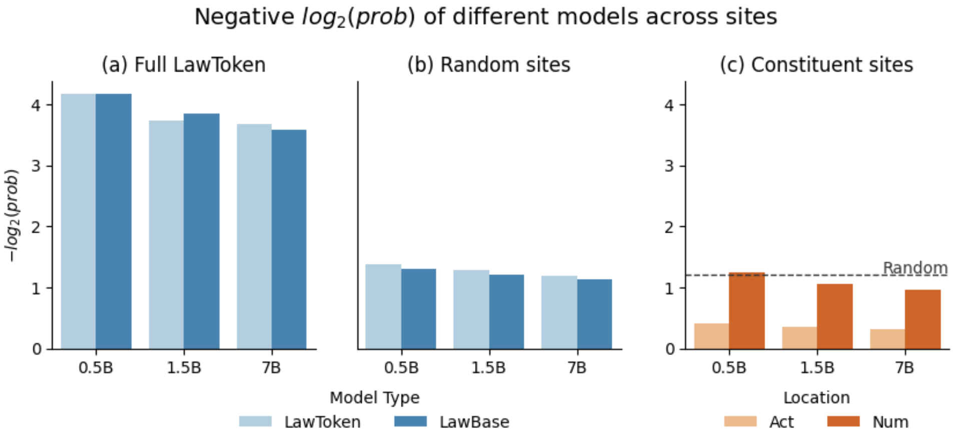


Figure 4: Negative $log_2(prob)$ of next-token predictions of different sites across models. (a) **Full LawToken** refers to the true law token and the multi-token sequences of the law reference. (b) The **Random sites** are the random locations before the target. (c) **Constituent sites** are the Act and Num sites. The dashed line is added as a visual reference, which are the values of 1.5B LawBase in the Random sites.

were to be compressed with an optimal compressor (Deletang et al., 2024; Tseng et al., 2024). That is to say, the prediction probabilities, particularly when transformed with a 2-based logarithm, signify the degree of difficulty the model has in predicting the law tokens or the law references based on the context it has encountered so far.

Figure 4 presents the results of prediction probabilities. Interestingly, despite the drastically different tokenization – where the law reference in LawBase has 11.90 tokens and only one in LawToken – their information contents (the $log_2(prob)$, summed over all tokens in law references) are largely the same across model sizes. However, this does not suggest intrinsic differences in decoding capacities between model types. As shown in Figure 4(b), LawBase models are not generally more efficient than the LawToken ones as the information contents remain comparable in the random sites where the predicted tokens occur before the law token. The findings are consistent with the previous representation similarities results, where the model states of the last hidden layers are almost the same in the *Pre* sites (except for the 1.5B model size, Figure 3a). Furthermore, this makes sense when considering the law token or reference conceptually: they are only two realizations of the same concept in input tokens, so both model types are expected to encode the law token or reference with similar information contents.

However, the similar information contents of the law tokens and references do not fully account for the observed differences in law citation tasks. As suggested by the previous model states findings, both LawBase and LawToken models achieve similar qualities of model states, as indicated by the CKA scores. It is only when LawBase models begin decoding token by token that the representation similarities decrease, especially at the article numbering sites. This pattern is consistently reflected in Figure 4(c). When comparing the Act name (`Act`) and article numbering sites (`Num`), the `Act` sites show very low information contents, significantly lower than the Random sites. In contrast, the `Num` site has higher values comparable to the Random ones. These token-level prediction probability results align with the type-level representation similarity findings: although LawToken models exhibit better lexical representation in the early layers, both models ultimately encode a similar amount of information through context. The key difference is that the LawBase models decode the law reference in multiple steps, and the best decoding representations are already achieved before the first token of law reference. Afterward, the LawBase models struggle with the highly ambiguous tokens from article numbers (`Num` sites), as evidenced by the reduced type-level representation similarities and the lower token-level information content.

Finally, Figure 5 presents the prediction probabilities following the law tokens and references. Neither the LawToken nor LawBase models show significant effects after the legal mentions, except that the 0.5B and 1.5B models do show small but significant differences in the immediate token fol-

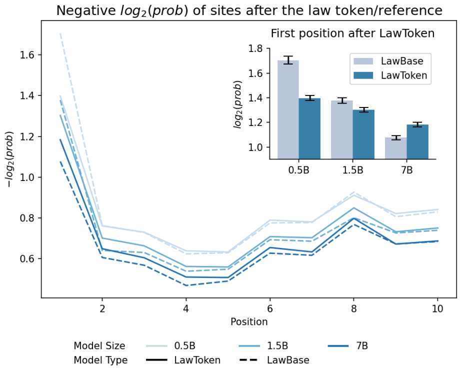


Figure 5: The negative $log_2(prob)$ of the sites after the target law token (LawToken) or reference (LawBase). The horizontal axis shows how many tokens are after the target. The inset highlights the first token after the target, where the LawToken models show higher predictability than the LawBase ones, except for the 7B model.

lowing position. This result is not surprising; as shown earlier, both model types encode comparable information content of legal mentions and can eventually compensate for the lexical information carried by the law token using context. Therefore, the holistic tokenization of law tokens only has a very limited effect on the following tokens.

5 Conclusion

Motivated by the form-meaning pairs of cognitive linguistics, we propose that the legal citations involving multi-word constituents can be processed not only as multi-token compounds but as holistic tokens. This paper empirically tests and investigates how different tokenizations affect model behaviors and representations. We train two model types: LawToken models, which consider the whole legal citation as one law token, and LawBase models, where the same citation is treated as multiple tokens. Our results show that LawToken models outperform LawBase models in legal citation tasks, particularly due to the article numbering component. We further analyze the model representations and find that both LawToken and LawBase models achieve comparable semantic representation quality. However, the LawBase model suffers from degraded representation in the multi-step decoding process, potentially increasing errors and hallucinations.

It may seem counterintuitive that treating an entire legal mention as a holistic law token improves task performance instead of leading to overfitting. However, this becomes understandable when we consider the compositionality problem inherent in the legal mention. In the mention, the article number component is the least informative constituent in a compound: it is constantly reused, lacks intrinsic connection to the intended meaning, and can only be resolved by context. The fact that LawBase models can achieve higher performance through fine-tuning, yet still fall short of LawToken models, suggests an upper bound to what contextualization alone can achieve. Beyond that, the model may need a more efficient or more discriminative cue, i.e., a law token in this case, to link with the intended semantics. In this sense, the model either considers the legal mention as a single "word" or compress a compound as a token, depending on one's definition of "word." Regardless, this line of reasoning align with linguistic models that do not assume the compositionality processing of compounds or a fixed and static notion of words (Baayen et al., 2019; Libben, 2022). Moreover, while this study shows that a manually defined law token is beneficial, whether such tokens can be learned dynamically (Pagnoni et al., 2025) remains an open question.

The implications of the present findings extend beyond linguistic theory. Indeed, the ability of LawToken models to encode what requires multiple tokens in LawBase ones already highlights that the form-meaning mappings can operate in a larger scope. Furthermore, treating legal citations as law tokens has significant implications for future legal reasoning studies, particularly when examining potential circuits (Tigges et al., 2024; Prakash et al., 2024). Linguistic theories may not directly inform the development of LLMs. Instead, the growing use of LLMs now makes it possible for linguists to empirically test theoretical claims that were previously out of reach. When combined with such implementations, linguistic theories can begin to move toward "an integrated model that generates precise quantitative predictions for vast arrays of empirical findings" (Baayen, 2024), opening new pathways that connect LLMs with our existing knowledge of language.

Acknowledgments

This study is supported by the ERC Horizon Europe Programme, ERC-2021-ADG, No. 101054902 (SUBLIMINAL). We also thank NVIDIA-NTU Artificial Intelligence Joint Research Center for providing computational resources.

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*.

Pegah Alipoor and Sabine Schulte im Walde. 2020. Variants of vector space reductions for predicting the compositionality of English noun compounds. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4379–4387, Marseille, France. European Language Resources Association.

R Harald Baayen. 2024. The wompom. *Corpus Linguistics and Linguistic Theory*, 20(3):615–648.

R Harald Baayen, Yu-Ying Chuang, Elnaz Shafaei-Bajestan, and James P Blevins. 2019. The discriminative lexicon: A unified computational model for the lexicon and lexical processing in comprehension and production grounded not in (de) composition but in linear discriminative learning. *Complexity*, 2019(1):4895891.

Melanie J Bell and Martin Schäfer. 2016. Modelling semantic transparency. *Morphology*, 26:157–199.

Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146.

Joan Bybee. 2010. *Language, usage and cognition*. Cambridge University Press.

Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre FT Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and 1 others. 2024. Saullm-7b: A pioneering large language model for law. *arXiv preprint arXiv:2403.03883*.

Silvio Cordeiro, Aline Villavicencio, Marco Idiart, and Carlos Ramisch. 2019. Unsupervised compositionality prediction of nominal compounds. *Computational Linguistics*, 45(1):1–57.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Gregoire Deletang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language modeling is compression. In *The Twelfth International Conference on Learning Representations*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Adele E. Goldberg. 2013. Constructionist Approaches. In *The Oxford Handbook of Construction Grammar*. Oxford University Press.

Adele E Goldberg. 2024. Usage-based constructionist approaches and large language models. *Constructions and Frames*.

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, Salt Lake City, Utah. Association for Computational Linguistics.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2024. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36.

Peter Henderson, Tatsunori Hashimoto, and Mark Lemley. 2023. Where's the liability in harmful ai speech? *J. Free Speech L.*, 3:589.

David Herel and Tomas Mikolov. 2024. Thinking tokens for language modeling. *arXiv preprint arXiv:2405.08644*.

Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. 2024. Gpt-4 passes the bar exam. *Philosophical Transactions of the Royal Society A*, 382(2270):20230254.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

G. Lakoff. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*. University of Chicago Press.

Jieh-Sheng Lee. 2023. Lexgpt 0.1: pre-trained gpt-j models with pile of law. In *Proceedings of the Seventeenth International Workshop on Juris-Informatics 2023 (JURISIN 2023)*, pages 15–24.

Gary Libben. 2022. From lexicon to flexicon: The principles of morphological transcendence and lexical superstates in the characterization of words in the mind. *Frontiers in Artificial Intelligence*, 4:788430.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7411–7425, Online. Association for Computational Linguistics.

Timothee Mickus, Denis Paperno, and Matthieu Constant. 2019. Mark my word: A sequence-to-sequence approach to definition modeling. In *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, pages 1–11, Turku, Finland. Linköping University Electronic Press.

Filip Miletić and Sabine Schulte im Walde. 2023. A systematic search for compound semantics in pretrained bert architectures. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1499–1512.

Thanapon Noraset, Chen Liang, Larry Birnbaum, and Doug Downey. 2017. Definition modeling: Learning to define word embeddings in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Xenia Ohmer, Elia Bruni, and Dieuwke Hupkes. 2024. From form (s) to meaning: Probing the semantic depths of language models using multisense consistency. *Computational Linguistics*, pages 1–51.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, and 1 others. 2025. Byte latent transformer: Patches scale better than tokens. In *ICML 2025 Workshop on Tokenization and Beyond (TokShop)*, Vancouver Convention Center, Vancouver, BC, Canada. Extended version available at arXiv:2412.09871.

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, Online. Association for Computational Linguistics.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *The Twelfth International Conference on Learning Representations*.

Siva Reddy, Diana McCarthy, and Suresh Manandhar. 2011. An empirical study on compositionality in compound nouns. In *Proceedings of 5th international joint conference on natural language processing*, pages 210–218.

Ian Rodgers, John Armour, and Mari Sako. 2023. How technology is (or is not) transforming law firms. *Annual Review of Law and Social Science*, 19(1):299–317.

Laura Eline 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 *Thirty-seventh Conference on Neural Information Processing Systems*.

Vered Shwartz and Ido Dagan. 2019. Still a pain in the neck: Evaluating text representations on lexical composition. *Transactions of the Association for Computational Linguistics*, 7:403–419.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. LLM circuit analyses are consistent across training and scale. In *Proceedings of the 9th Workshop on Representation Learning for NLP (RepL4NLP-2024)*, pages 290–303, Bangkok, Thailand. Association for Computational Linguistics.

Yu-Hsiang Tseng, Pin-Er Chen, Da-Chen Lian, and Shu-Kai Hsieh. 2024. The semantic relations in LLMs: An information-theoretic compression approach. In *Proceedings of the Workshop: Bridging Neurons and Symbols for Natural Language Processing and Knowledge Graphs Reasoning (NeusymBridge) @ LREC-COLING-2024*, pages 8–21, Torino, Italia. ELRA and ICCL.

Yu-Hsiang Tseng, Mao-Chang Ku, Wei-Ling Chen, Yu-Lin Chang, and Shu-Kai Hsieh. 2023. Vec2Gloss: definition modeling leveraging contextualized vectors with Wordnet gloss. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 679–690, Hong Kong, China. Association for Computational Linguistics.

Benjamin Weiser. 2023. Here's what happens when your lawyer uses chatgpt. *The New York Times*, 27.

Ethan Gotlieb Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the predictive power of neural language models for human real-time comprehension behavior. In *Proceedings of the 42nd Annual Meeting of the Cognitive Science Society*, page 1707–1713.

Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. 2024. Cbr-rag: case-based reasoning for retrieval augmented generation in llms for legal question answering. In *International Conference on Case-Based Reasoning*, pages 445–460. Springer.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Pienie Zwitserlood. 2014. The role of semantic transparency in the processing and representation of dutch compounds. In *Morphological Structure, Lexical Representation and Lexical Access (RLE Linguistics C: Applied Linguistics)*, pages 341–368. Routledge.

Appendix

A Limitation

This paper examines how tokenization impacts model performance in predicting legal citations and shaping semantic representations, using Taiwan's legal citation system as our dataset. However, we acknowledge that the citation formats vary across different countries, especially considering that Taiwan follows the civil law system, in contrast to the common law system, as in the British or the United States. Although we believe the findings are relevant to other surface forms, this work remains constrained by the dataset upon which it is trained and tested. On the theoretic side, our results indicate that the multi-token models (i.e., LawBase) suffer from degraded representations during multi-step decoding. Yet, it remains unclear whether the degradation stems from the nature of legal citation or can be generalized to a more general form-meaning mapping problem, such as those found in compounds or multi-word expressions. Addressing these questions requires more experiments and analyses in future studies.

B Examples of training data

The dataset comprises laws and verdicts in Taiwan. The examples of training data shown below, other than the "Question", "Answer" and the law citation tokens, are all in traditional Chinese. Personal names are anonymized, although they appear in the original verdicts. English translations are provided for clarity but are never seen by the model.

B.1 Example 1

理 由 一、本件原裁定以抗告人陳○○因不服臺灣新北地方檢察署 101年度執更丑字第4313號執行指揮書而聲明異議，經原審以10 7年度聲字第544號裁定駁回，並囑託法務部矯正署宜蘭監獄長官於民國107 年4月2日向抗告人合法送達，此有送達證書附卷可查。其抗告期間之末日為同年月7 日星期六，翌日為星期日，均為休息日。其提起抗告，僅可於休息日次日即同年月9 日星期一為之。乃竟遲至同年月11日始向法務部矯正署宜蘭監獄長官提起抗告，有抗告人所提刑事抗告狀在卷可證。已逾5日抗告期間，因依<刑事訴訟法|411>前段規定駁回其抗告。經核尚無不合。 二、抗告意旨徒以107年4月4日至同年月8日為休假期日，依社會通念休假期日不計算期日，同年月12日才是抗告終止日等語，係憑己見指摘原裁定不當。其抗告為無理由，應予駁回。 據上論結，應依<刑事訴訟法|412>，裁定如主文。

English Translation
Reasoning

1. The original ruling was based on the fact that the appellant, Chen xx-xxx, objected to the execution order No. 4313 (Year 101, Re-Execution Chou Character) issued by the Taiwan New Taipei District Prosecutors Office, and filed an objection accordingly. The original court dismissed the objection in Ruling No. 544 (Year 107, Objection Character), and entrusted the Yilan Prison Warden of the Agency of Corrections, Ministry of Justice to serve the ruling lawfully to the appellant on April 2, 2018 (Year 107 of the Republic of China calendar). This is confirmed by the certificate of service included in the case file.

The last day of the appeal period was Saturday, April 7 of the same year, and the next day, Sunday, was also a rest day. Therefore, an appeal could only be filed on the next business day, which was Monday, April 9 of the same year. However, the appellant did not file the appeal until April 11 of the same year, submitting it to the Yilan Prison Warden. This is proven by the criminal appeal document submitted by the appellant on record. Since this was beyond the 5-day appeal period, the appeal is dismissed according to the first part of Article 411 of the Code of Criminal Procedure. Upon review, this decision is deemed proper.

2. The grounds for appeal merely argue that the period from April 4 to April 8, 2018, was a holiday, and that under common social understanding, holidays are not counted toward deadlines, hence April 12 should be considered the last day to appeal. This is a subjective interpretation and an unfounded criticism of the original ruling. The appeal lacks merit and should be dismissed.

In conclusion, pursuant to Article 412 of the Code of Criminal Procedure, the ruling is made as stated in the main text.

B.2 Example 2

<土地法|46-2>重新實施地籍測量時，土地所有權人應於地政機關通知之限期內，自行設立界標，並到場指界。逾期不設立界標或到場指界者，得依左列順序逕行施測： 一、鄰地界址。 二、現使用人之指界。 三、參照舊地籍圖。 四、地方習慣。 土地所有權人因設立界標或到場指界發生界址爭議時，準用第五十九

條第二項規定處理之。

English Translation
<Land Act|46-2> When a cadastral resurvey is being conducted, the landowner shall, within the deadline specified in the notice issued by the land administration authority, install boundary markers and appear on-site to indicate the boundaries. If the landowner fails to install boundary markers or appear on-site within the prescribed period, the survey may proceed directly according to the following order of priority: (1) The boundaries of adjacent parcels. (2) The boundary indications provided by the current user of the land. (3) Reference to the old cadastral maps. (4) Local customs. If a boundary dispute arises due to the installation of boundary markers or the on-site boundary indication by the landowner, the provisions of Paragraph 2, Article 59 shall apply mutatis mutandis.

B.3 Example 3

原審以：被上訴人主張上訴人為系爭支票發票人，伊為執票人等情，為上訴人所不爭執，且有系爭支票影本可稽，堪信為真實。 上訴人抗辯：鄭○○詐稱呂○○對其負有債務，且將來會負責支付系爭支票票款，要求開立系爭支票與被上訴人，但實際上呂○○未積欠鄭○○錢，呂○○被鄭○○及被上訴人詐欺，陷於錯誤交付系爭支票等語，可知上訴人因認呂○○對鄭○○負有債務始簽發系爭支票，嗣因呂○○與鄭○○間發生債務糾葛，呂○○始否認對鄭○○負有債務，此由鄭○○於上訴人簽發系爭支票後，嗣後另案起訴請求呂○○返還投資款即明，復有民事起訴狀影本可參，況上訴人自始未提出任何證據佐證其被詐欺或脅迫而簽發系爭支票，上訴人此部分抗辯，不足為採。上訴人另抗辯系爭支票之原因關係不存在，惟票據係文義證券及無因證券，屬不要因行為，故執票人祇須就該票據作成之真實負證明之責，關於票據給付之原因，並不負證明之責任，票據債務人仍應就其抗辯之原因事由，先負舉證責任。然上訴人未就其抗辯事由負舉證責任，則被上訴人請求上訴人給付系爭支票款1100萬元，及自105年8月1日起至清償日，按週年利率6%計算之利息，為有理由，應予准許等詞，因而維持第一審所為上訴人敗訴之判決，駁回其上訴，經核於法並無不合。 按票據乃文義證券及無因證券，票據上之權利義務悉依票上所載文義定之，與其基礎之原因關係各自獨立，票據上權利之行使不以其原因關係存在為前提。是執票人行使票據上權利時，就其基礎之原因關係確係有效存在不負舉證責任。僅於票據債務人以自己與執票人間所存抗辯事由對抗執票人，而該票據基礎之原因關係經確立者，法院就此項原因關係進行實體審理時，當事人於該原因關係是否有效成立或已否消滅等事項有所爭執，始應適用各該法律關係之舉證責任分配原則。查上訴人為系爭支票發票人，被上訴人為執票人，為原審所確定。被上訴人主張上訴人係為返還伊投資款而簽發系爭支票，上訴人則抗辯係鄭○○詐稱呂○○對其負有債務，且將來會負責支付系爭支票票款，而簽發交付系爭支票予被上訴人，就被上訴人取得系爭支票之原因關係，各執一詞，並未確立，依上說明，仍應由上訴人就其抗辯之原因關係，負舉證之責。原審因上訴人未舉證證明系爭支票之原因關係，而為其不利之認定，自不違背舉證責任分配原則。至上訴人援引之本院判決，或係就該票據基礎之原因關係經確立情形所為之闡述，或與本件事實有所差異，均無從比附援引。上訴論旨，指摘原判決不當，聲明廢棄，非有理由。 據上論結，本件上訴為無理由。依、、、，判決如主文。<cite><民事訴訟法|436-2|2>,<民事訴訟法|78>,<民事訴訟法|449|1>,<民事訴訟法|481></cite>

English Translation

The court of first instance found that: the appellee asserted that the appellant was the issuer of the check in dispute, and that the appellee was the holder of said check—facts not contested by the appellant and supported by a copy of the disputed check, which is deemed credible and authentic.

The appellant contended that Cheng xxx-xxx falsely claimed that Lu xxx-xxx was indebted to him and would be responsible for the payment of the disputed check, and thus requested the issuance of the check jointly with the appellee. However, in fact, Lu xxx-xxx owed no debt to Cheng xxx-xxx, and the check was delivered under a mistake caused by the fraud committed by Cheng xxx-xxx and the appellee. From this, it is clear that the appellant issued the check under the belief that Lu xxx-xxx owed Cheng xxx-xxx a debt. After a dispute arose between Lu xxx-xxx and Cheng xxx-xxx regarding said debt, Lu xxx-xxx denied owing any such debt. This is evident from the fact that, after the appellant issued the check, Cheng xxx-xxx filed a separate lawsuit seeking return of his investment from Lu xxx-xxx; a copy of that civil complaint is also on record. Moreover, the appellant never submitted any evidence to support the claim of having been defrauded or coerced into issuing the check. Therefore, this part of the appellant's defense lacks merit.

The appellant further argued that there was no underlying transaction or cause for the issuance of the check in dispute. However, as a negotiable instrument, a check is a documentary and abstract security—its legal force derives from the wording on the instrument itself and is independent of the underlying cause. Accordingly, the holder of the check only bears the burden of proof with respect to the authenticity of the check itself, and not regarding the underlying cause of payment. On the contrary, it is the debtor on the check who must bear the burden of proof for any defenses raised against payment. Since the appellant failed to provide proof supporting the grounds for their defense, the appellee's claim for payment of NT$11 million, along with interest calculated at an annual rate of 6% from August 1, 2016 until the date of repayment, is well-founded and should be granted. Therefore, the judgment of the court of first instance, which ruled against the appellant, is upheld, and the appeal is dismissed. Upon review, this judgment is in accordance with the law.

According to law, negotiable instruments are documentary and abstract in nature. The rights and obligations indicated on the face of the instrument govern, independently of any underlying transaction. The exercise of rights under a negotiable instrument does not require proof of the existence of the underlying relationship. Thus, when a holder of an instrument seeks to exercise such rights, they bear no burden of proof regarding the validity of the underlying relationship. Only when the debtor on the instrument raises a defense based on their own relationship with the holder—and the underlying cause of the instrument is thereby established—does the court proceed to substantively examine that cause. In such cases, the burden of proof is allocated according to the relevant substantive legal relationships.

The appellant is confirmed to be the issuer of the check in dispute, and the appellee its holder, as determined by the lower court. The appellee claims the check was issued by the appellant to repay an investment, while the appellant claims the check was issued under the false impression—due to misrepresentation by Cheng xxx-xxx—that Lu xxx-xxx owed Cheng a debt and would pay the amount. Each party presents a different version of the reason behind the check's issuance, and no cause has been established. According to the principles stated above, it remains the appellant's responsibility to prove their asserted cause. Since the appellant failed to meet that burden, the lower court's unfavorable ruling does not violate the principle of burden of proof allocation.

As for the judgments cited by the appellant, those either concern cases where the underlying cause of the negotiable instrument was established, or differ in facts from the present case, and are therefore inapplicable. The grounds of appeal, which challenge the lower judgment as improper and request its reversal, are without merit.

In conclusion, the appeal in this case is groundless. Pursuant to,,, judgment is rendered as stated in the main text. <cite><Civil Procedure Code|436-2|2>,<Civil Procedure Code|78>,<Civil Procedure Code|449|1>,<Civil Procedure Code|481></cite>

C Examples of evaluation tasks

C.1 Example of long-context citation task

Question: ``四、原審已依吳○○就醫之相關病歷資料、診斷證明書、臺中榮民總醫院函文、勞動部勞工保險局函文等資料，載敘吳○○傷勢及結果甚詳，上訴人及其辯護人於審理中並未爭執有何記載錯誤、不實之處，則原審綜合全案證據資料，依其所採取之證據及得心證理由之說明，已足以認定吳○○受有右眼創傷性黃斑部裂孔造成僅能辨識眼前指數10公分，且右眼視野缺損、最佳矯正視力為0.01，達一目視能嚴重減損之重傷害，而未再為其他無益之調查，自無上訴意旨所指適用法則不當、調查未盡之違法情形可言。又本院為法律審，不為事實之調查，上訴人上訴於本院，始提出其蒐得吳○○工作之照片作為新證據資料，執以指摘原判決違誤，亦非上訴第三審之合法理由。 五、綜合前旨及其他上訴意旨，無非係置原判決所為明白論斷於不顧，仍持已為原判決指駁之陳詞再事爭辯，或對於事實審法院取捨證據與自由判斷證據證明力之職權行使，徒以自己之說詞，為相異評價，任意指為違法，或單純為事實上枝節性之爭辯，要與法律規定得為第三審上訴理由之違法情形，不相適合。本件上訴違背法律上之程式，應予駁回。 據上論結，應依刑事訴訟法前段，判決如主文。<cite>''

Answer: ``<刑事訴訟法|395>,<刑事訴訟法|377>''

English translation

Question: 4. The original trial court had already reviewed relevant medical records, diagnostic certificates, correspondence from Taichung Veterans General Hospital, and documents from the Bureau of Labor Insurance of the Ministry of Labor. These materials provided a detailed description of Wu xxx-xxx's injuries and medical outcomes. During the proceedings, neither the appellant nor their defense counsel disputed any inaccuracies or falsehoods in those records. Therefore, the trial court, based on the totality of the evidence and its reasoning for the credibility of the accepted proof, was fully justified in concluding that Wu xxx-xxx sustained a traumatic macular hole in his right eye, rendering him able to perceive only hand motion at 10 cm in front of the eye. He also suffers from a loss of visual field and a best-corrected visual acuity of 0.01 in that eye—constituting a serious injury causing severe impairment to monocular vision. As such, the court did not engage in further unnecessary investigation, and there is no indication of improper application of the law or failure to investigate, as alleged in the appeal.

Furthermore, this Court serves as a court of law, not of fact. The appellant's submission of photographs allegedly showing Wu xxx-xxx at work—presented for the first time on appeal to this Court as new evidence and cited as grounds to challenge the lower court's decision—does not constitute a legitimate reason for a third-instance appeal.

In sum, the foregoing and the rest of the appeal merely disregard the clear reasoning of the original judgment, reasserting arguments already addressed and rejected by the lower court, or challenge the trial court's discretion in evaluating and weighing evidence by offering alternative interpretations based on the appellant's own narrative. Such arguments are factual disputes over minor points and do not qualify as legal grounds for a third-instance appeal under the law. This appeal thus violates procedural requirements and shall be dismissed.

Based on the above reasoning, and pursuant to the first part of the Code of Criminal Procedure, judgment is rendered as stated in the main text. <cite>

Answer: <Code of Criminal Procedure|395>, <Code of Criminal Procedure|377>

C.2 Example of short-context citation task

`Question: ``按當事人因無資力支出訴訟費用而聲請訴訟救助者，關於無資力支出訴訟費用之事由，應提出可使法院信其主張為真實並能即時調查之證據，以釋明之。此觀之規定自明。<cite>''`

`Answer: ``<民事訴訟法|109|2>''`

English translation According to the law, when a party applies for litigation aid on the grounds of inability to afford litigation costs, they must provide evidence that is sufficient to convince the court of the truthfulness of their claim and that can be promptly verified by the court, in order to clarify the grounds for their financial inability. This is clearly stipulated by law. <cite>

Answer: <Civil Procedure Code|109|2>

C.3 Example of law-naming task

`Question: ``物之發明之實施，指製造、為販賣之要約、販賣、使用或為上述目的而進口該物之行為。<cite>''`

`Answer: ``<專利法|58|2>''`

English translation

Question: The implementation of an invention of a product refers to the acts of manufacturing, offering for sale, selling, using, or importing the product for the above purposes. <cite>

Answer: "<Patent Act|58|2>"

D Prompt design for GPT-4o-mini

D.1 System prompt

`"你是一名熟悉中華民國法條的法律專業人士，在任何情境下，你都能援引最適切的法條予以回應。"`

English translation

You are a legal professional well-versed in the laws of the Republic of China (Taiwan), and in any situation, you are able to cite the most appropriate legal provisions in your response.

D.2 Prompt template

```
---

<判決書>
[....]
<cite>
<法條>
<刑事訴訟法|449|1>,<刑事訴訟法|449|3>,<毒品危害防制條例|20>,<毒品危害防制條例|23|2>,<刑事訴訟法|454|1>,<毒品危害防制條例|23>,<毒品危害防制條例|10|2>
```

```
---
<判決書>
{Question}
<cite>
<法條>
"
```

English translation

Your task is to identify the most relevant legal provisions. First, refer to the judgments and their associated legal articles in the examples below. Then, a second judgment will be presented—this is your task. Based on the content and subject of that judgment, please provide the applicable legal provisions in the specified format and return them as a JSON file.

```
---

<verdict>
[....]
<cite>
<laws>
<Code of Criminal Procedure|449|1>,<Code of
Criminal Procedure|449|3>,<Narcotics Hazard
Prevention Act|20>,<Narcotics Hazard Prevention
Act|23|2>,<Code of Criminal Procedure|454|1>,
<Narcotics Hazard Prevention Act|23>,<Narcotics
Hazard Prevention Act|10|2>

---

<verdict>

{Question}
<cite>
<laws>
```

Interpersonal Memory Matters: A New Task for Proactive Dialogue Utilizing Conversational History

Bowen Wu[1,2], **Wenqing Wang**[1], **Haoran Li**[2], **Yunhan Deng**[2],
Ying Li[1*], **Jingsong Yu**[1], **Baoxun Wang**[2]
[1]School of Software & Microelectronics, Peking University, Beijing, China
[2]Platform and Content Group, Tencent
`{jason_wbw,li.ying,yujingsong}@pku.edu.cn`
`{wangwenqing}@stu.pku.edu.cn`
`{heyfonli,yunhandeng,asulewang}@tencent.com`

Abstract

Proactive dialogue systems aim to empower chatbots with the capability of leading conversations towards specific targets, thereby enhancing user engagement and service autonomy. Existing systems typically target predefined keywords or entities, neglecting user attributes and preferences implicit in dialogue history, hindering the development of long-term user intimacy. To address these challenges, we take a radical step towards building a more human-like conversational agent by integrating proactive dialogue systems with long-term memory into a unified framework. Specifically, we define a novel task named **M**emory-**a**ware **P**roactive **Dia**logue (**MapDia**). By decomposing the task, we then propose an automatic data construction method and create the first **Ch**inese **M**emory-**a**ware **P**roactive **Data**set **(ChMapData)**. Furthermore, we introduce a joint framework based on Retrieval Augmented Generation (RAG), featuring three modules: Topic Summarization, Topic Retrieval, and Proactive Topic-shifting Detection and Generation, designed to steer dialogues towards relevant historical topics at the right time. The effectiveness of our dataset and models is validated through both automatic and human evaluations. We release the open-source framework and dataset at `https://github.com/FrontierLabs/MapDia`.

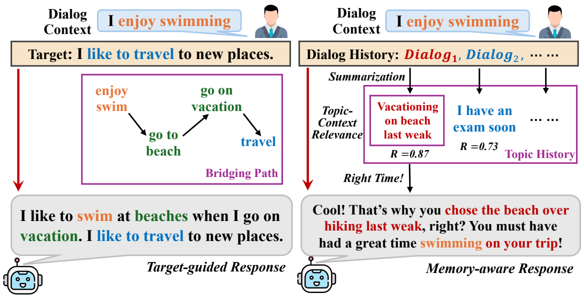

Dialog Context
I enjoy swimming
Target: I like to travel to new places.
enjoy swim
go to beach
go on vacation
travel
Bridging Path
I like to swim at beaches when I go on vacation. I like to travel to new places.
Target-guided Response
Dialog Context
I enjoy swimming
Dialog History: $Dialog_1$, $Dialog_2$, ……
Summarization
Topic-Context Relevance
Vacationing on beach last weak
R =0.87
I have an exam soon
R =0.73
……
Topic History
Right Time!
Cool! That's why you chose the beach over hiking last weak, right? You must have had a great time swimming on your trip!
Memory-aware Response

Figure 1: Comparison of previous proactive dialogue systems (Left) that extracted from Gupta et al. (2022) and our system (Right) on the same sample: The left system transitions the context to a pre-designed target through a bridging path, whereas our system involving summarization, retrieval, and timing detection to generate the memory-aware response.

1 Introduction

Recent years have witnessed significant advancements in the design of conversational agents, with various methods proposed to generate engaging responses, e.g., external knowledge (Xu et al., 2023; Yang et al., 2024), personality traits (Madaan et al., 2020; Ju et al., 2022), and the utilization of large-scale models (Fan et al., 2023; Liu et al., 2024). Among these, proactive behavior in agents—where the agent takes control of the conversation instead of merely responding passively to users—has been identified as a crucial advancement for the next generation of conversational AI (Deng et al., 2023).

Nevertheless, a more lifelike dialogue system must go beyond generating contextually appropriate responses; it should also employ more intelligent mechanisms to maintain a coherent social relationship over time (Campos et al., 2018). Memory, in particular, has already been acknowledged as essential for driving conversations, developing rapport, and maintaining long-term connections (Zhong et al., 2024; Maharana et al., 2024). However, existing proactive dialogue systems insufficiently exploit memory mechanisms, whose targets are pre-defined ones, such as topical keywords (Tang et al., 2019), knowledge entities (Wu et al., 2019a), conversational goals (Liu et al., 2020), while overlooking the contextual richness of dialogue history. Even advanced ChatGPT (Achiam et al., 2023) faces constraints, yielding responses to inquiries regarding the context, reflecting a passive engagement with historical data. On the one hand, predefined topics may not align with user interests, which is further discussed in the Appendix A based on previous research. On the other hand, as users' personal information accumulates over time, ig-

---

[*]Corresponding author

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 47–67
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

noring historically interpersonal interactions with agents (i.e., the memory mentioned in this paper) causes a failure to capture their attributes and preferences. All of these contradict the proactive dialogue motivation to improve user engagement.

To bridge this gap, we integrate proactive dialogue systems with memory mechanisms, moving closer to creating more intelligent and human-like conversational agents. Specifically, we propose the Memory-aware Proactive Dialogue task, depicted in Figure 1. Contrary to traditional proactive systems that respond based solely on a pre-designed target, our approach extracts topics from past dialogues, identifies the most relevant topic as target, assesses the appropriateness of topic transitions, and finally integrates memory into the response.

To minimize extensive human annotation, we introduce an automated four-step data construction method, proven effective in validation. This method uses GPT-4 (Achiam et al., 2023) to develop ChMapData, the first Chinese Memory-aware Proactive Dataset encompassing all the information to perform MapDia including dialogue histories, corresponding topics, current contexts, topic transition timings, and history-informed responses. Specifically, we guide the generation of certain historical dialogues using memorable subjects, such as events in which the user has participated. These dialogues serve as references for the subsequent generation of proactive chats and shape the current context, ensuring continuity and facilitating memory recall. The final segment of the data determines the appropriateness of topic transitions during conversations and formulates responses accordingly, concentrating on either shifting to a historical topic or maintaining the ongoing context.

With ChMapData, we propose a new proactive dialogue framework containing three components: 1) Topic Summarization, condensing historical dialogues into topics for simplified retrieval; 2) Topic Retrieval, identifying the most relevant historical topic with a RAG mechanism; and 3) Proactive Topic-shifting Detection and Generation, timing and executing topic transitions at optimal moments.

The main contributions are as follows: 1) We are the first to integrate memory technique into proactive dialogue systems and introduce a novel task of **MapDia**, where the system navigates current dialogue towards relevant historical topics at an appropriate opportunity; 2) We propose an effective automated data construction methodology and, based on this, construct **ChMapData**, the first memory-aware proactive dialogue dataset in Chinese; 3) We present a RAG-based proactive dialogue framework that combines summarization, retrieval, timing detection, and response generation mechanisms. Both automatic and human evaluations demonstrate the effectiveness of our method.

2 ChMapData Dataset

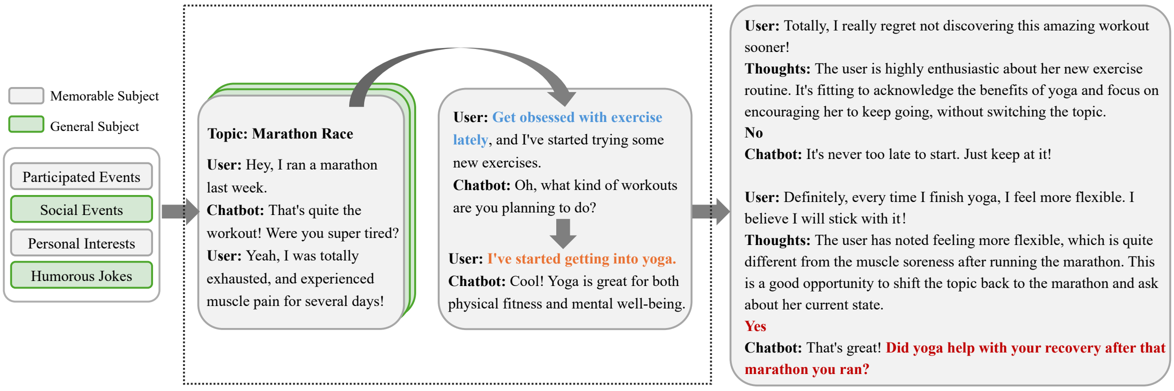
Despite the existence of Chinese datasets for proactive dialogue systems (Wu et al., 2019b; Zhou et al., 2020), they lack the ability to engage with the dialogue history while either steering the conversation towards a new topic or continuing with the current one. To fill this gap, we automatically generate the first multi-turn dataset designed for proactive conversations that leverage historical memory utilizing GPT-4 with a range of prompts (detailed in Appendix B). This process is further validated by annotators, thereby eliminating the high costs and lengthy procedures associated with human annotation. Note that we construct the dataset through GPT-4 because LLMs have been proven as powerful tools for synthetic data generation (Agrawal et al., 2022; Liu et al., 2022; Bitton et al., 2023).

2.1 Data Construction

Figure 2 gives an overview of the ChMapData construction pipeline, involving a four-step process.

1) Subject Selection. The initial phase involves GPT-4 brainstorming to generate a pool of potential subjects. Out of these, 11 subjects are manually selected and categorized into two groups: Memorable Subjects, intended to evoke recollections related to the user's own experiences , including personal interests, feelings, skills, traits, participating events, and events' progression; as well as General Subjects, which have no direct connection to the user's life and are not typically brought up again, encompassing social events, opinion debates, humorous jokes, audience stories, and knowledge sharing.

2) Topic and Dialogue Generation. On the basis of 6 Memorable and 5 General Subjects selected above, a fine-grained topic along with corresponding dialogues are generated serving as a bank of dialogue histories. To emulate the flow of real-life conversations, we crafted more dialogues for General Subjects than Memorable ones, at a ratio of 2:1, yielding 500 and 250 dialogues for each respectively, culminating in 4,000 topic-dialogue pairs. Each dialogue is limited to 5-8 turns to maintain brevity and focus. Among these, 1500 dialogues



Topic: Marathon Race
User: Hey, I ran a marathon last week.
Chatbot: That's quite the workout! Were you super tired?
User: Yeah, I was totally exhausted, and experienced muscle pain for several days!

User: Get obsessed with exercise lately, and I've started trying some new exercises.
Chatbot: Oh, what kind of workouts are you planning to do?

User: I've started getting into yoga.
Chatbot: Cool! Yoga is great for both physical fitness and mental well-being.

User: Totally, I really regret not discovering this amazing workout sooner!
Thoughts: The user is highly enthusiastic about her new exercise routine. It's fitting to acknowledge the benefits of yoga and focus on encouraging her to keep going, without switching the topic.
No
Chatbot: It's never too late to start. Just keep at it!

User: Definitely, every time I finish yoga, I feel more flexible. I believe I will stick with it!
Thoughts: The user has noted feeling more flexible, which is quite different from the muscle soreness after running the marathon. This is a good opportunity to shift the topic back to the marathon and ask about her current state.
Yes
Chatbot: That's great! **Did yoga help with your recovery after that marathon you ran?**

(a) Subject Selection (b) Topic & Dialogue Generation (c) Dialogue Continuation (d) Topic-shift Detection & Response Generation

Figure 2: The pipeline of dataset construction. Not derived from the actual dataset.

originate from Memorable Subjects, designed to potentially trigger memory in subsequent conversations. For every one of these 1500 dialogues, 1 to 10 additional dialogues are selected from candidate pools and manually sequenced to construct a coherent, conflict-free multi-segment chat history.

3) Dialogue Continuation. Subsequently, we initiate a current dialogue session by generating two beginning turns, which is a continuation of each prior dialogue generated in the second step after a lapse of several days. Specifically, since step 2 ensures that each conversation history has a Memorable Subject-driven dialogue, we extend the dialogue to facilitate memory recollection in the following step. The two turns are generated separately: The first turn is derived from the topic and dialogue content (generated in Step 2), maintaining consistency with the prior dialogue, as indicated by the blue sentence in Figure 2. The second turn is then developed with a focus solely on the content of the first one (generated in Step 3) to prevent shifting the current conversation directly back to the dialogue history, as highlighted in orange. In total, 1,500 beginnings for current dialogues have been created, corresponding to the number of dialogue histories produced under Memorable Subject.

4) Topic-shift Detection and Response Generation. Ultimately, we continued the conversation based on the beginning of each current dialogue (generated in Step 3), and tried to naturally introduce new topics related to the preset memorable conversation history at appropriate moments. Inspired by the Chain of Thought (CoT) (Wang et al., 2022) technique, each turn additionally incorporates a "Thoughts" feature, aiming to enhance the accuracy and interpretability of the detection process, together with a decision-making mechanism to identify whether it's an appropriate opportunity to switch topics, as highlighted in red in Figure 2. It should be noted that ending the dialogue without switching to the historical topic is also permissible.

	Hist. Dlg.	**Curr. Dlg.**
# Dialogues	3,989[1]	1,464
# Utterances	40,619	16,373
# Unique Tokens	21,822	12,503
# Thoughts	-	5,081
# Topic-shift Sess.	-	1,254
Avg. # Utts. Length	33.23	38.97
Avg. # Utts. per Sess.	10.14	11.18

Table 1: Statistics of both historical (Hist. Dlg.) and current dialogue (Curr. Dlg.) dataset. **# Thoughts** represents the chatbot's considerations on whether to switch the topic at each turn. **# Topic-shift Sess.** refers to conversations that successfully revert to the historical topic. The calculation of **# Utterances** excludes the **# Thoughts**, considering only the dialogue segments.

2.2 Overall Statistics

After data construction, we enhanced the dataset's quality by checking and manually removing 36 dialogues from the current dialogues due to format inconsistencies or illogical "Thoughts," resulting in a total of 1,464 entries. Statistics of the ChMapData are presented in Table 1, which is reported from two aspects: historical dialogue, generated in Step 2, and current dialogue, initiated with two turns in Step 3 and extended to the end in Step 4. Out of these, 1,254 dialogues successfully recalled the historical topic, as indicated by a "Yes" output during the detection process. The remaining 210 dialogues, which consistently output "No" throughout the session, are also retained for training.

---

[1] Dialogue irrelevant to the subject has been filtered out.

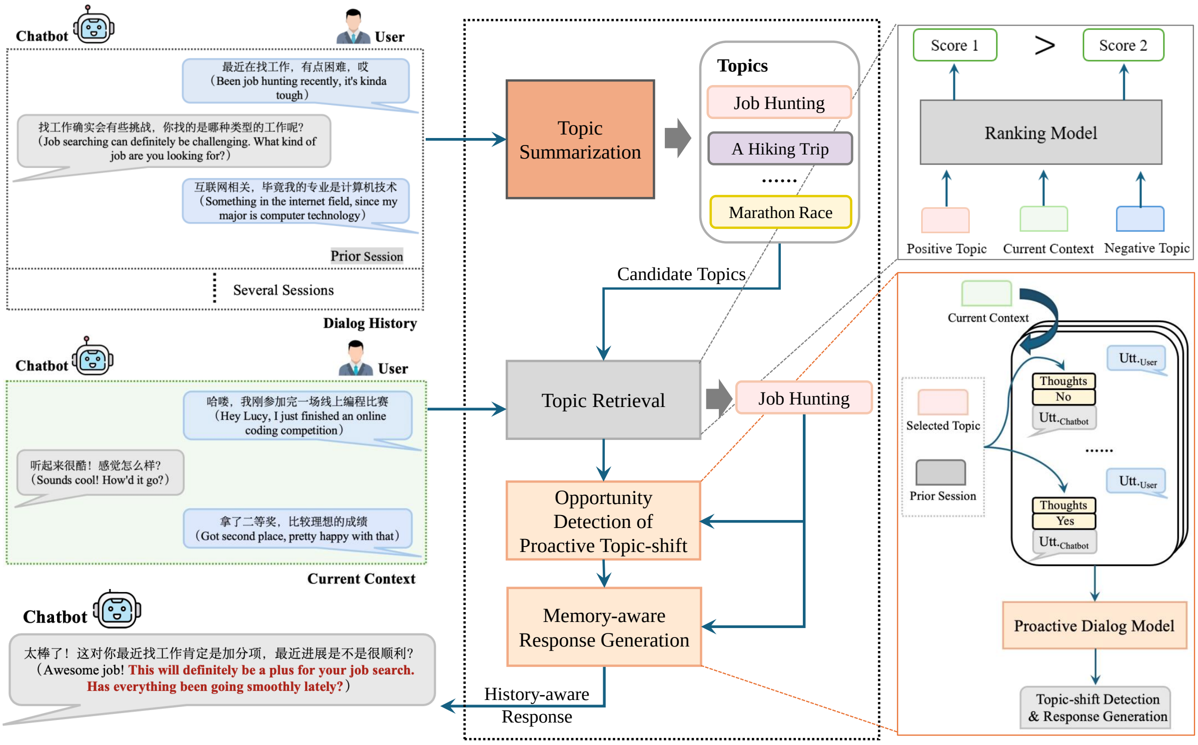

Chatbot
User
最近在找工作，有点困难，哎（Been job hunting recently, it's kinda tough）
找工作确实会有些挑战，你找的是哪种类型的工作呢?（Job searching can definitely be challenging. What kind of job are you looking for?）
互联网相关，毕竟我的专业是计算机技术（Something in the internet field, since my major is computer technology）
Prior Session
Several Sessions
Dialog History
Chatbot
User
哈喽，我刚参加完一场线上编程比赛（Hey Lucy, I just finished an online coding competition）
听起来很酷！感觉怎么样?（Sounds cool! How'd it go?）
拿了二等奖，比较理想的成绩（Got second place, pretty happy with that）
Current Context
Chatbot
太棒了！这对你最近找工作肯定是加分项，最近进展是不是很顺利?（Awesome job! This will definitely be a plus for your job search. Has everything been going smoothly lately?）
Topic Summarization
Topics
Job Hunting
A Hiking Trip
……
Marathon Race
Candidate Topics
Topic Retrieval
Job Hunting
Opportunity Detection of Proactive Topic-shift
Memory-aware Response Generation
History-aware Response
Score 1
>
Score 2
Ranking Model
Positive Topic
Current Context
Negative Topic
Current Context
Selected Topic
Prior Session
Utt.User
Thoughts
No
Utt.Chatbot
……
Utt.User
Thoughts
Yes
Utt.Chatbot
Proactive Dialog Model
Topic-shift Detection & Response Generation

Figure 3: An overview of our system. Left showcases an example of proactive dialogue with memory awareness. Middle outlines the pipeline, featuring a summarization model for topic extraction, a ranking model to identify relevant historical topics, and a proactive dialogue model for topic shifts and reintroducing past information at the appropriate moments. Right is a breakdown detailing how these models operate.

3 Approach

Task Definition. Given Given a set of dialogue history $H = \{d_1, d_2, \ldots, d_n\}$ consisting of n dialogues, where d_i denotes the i-th dialogue and t_i represents its topic, and the current dialogue context c, the system is tasked with generating a topic-shift response R that proactively guide the conversation c to a related historical topic t_r at an appropriate opportunity — specifically at turn τ.

Up until now, we have obtained the ChMapData, containing historical dialogues with the corresponding topics, current context, thoughts on topic shifting, and response content. With these supports, we propose partitioning MapDia into three distinct modules as follows and integrating them through a RAG framework as shown in Figure 3.

Topic Summarization. Xu et al. (2022a) noted that dense retrieval of past conversations has two drawbacks: it requires storing large amounts of context and places a heavy workload on the model to extract and retrieve information. To address this, we start by condensing dialogue history into topics using a summary model. The training data for this step is derived from historical dialogues along with their corresponding topics (referred to as *ChMapData-Sum*), and the model is trained to summarize a topic t_i for each dialogue history d_i.

Topic Retrieval. We then developed a ranking model to identify the most pertinent summarized topic t_r for the current context c, facilitating continuous memory updates and the integration of historical information within the dialogue system. This model utilizes context c, along with its dependent historical topic t, as outlined in ChMapData construction's Step 3 (denoted c-t pairs as *ChMapData-Ret*) [2]. Given that the ranking model trains a classifier to predict preference probabilities between pairs of responses, as modeled by the Bradley-Terry model (Bradley and Terry, 1952). To prepare the dataset, we use GPT-4 to evaluate the relevance of the target topic t and 29 other randomly chosen topics from the pool to c, generating positive T^+ and negative T^- samples. The highest-ranked topic and t form T^+; if they coincide, only one positive example is constructed. Topics ranked lower than t become T^-, enhancing the dataset while ensuring the top-ranked topic is never a negative example.

For each dialogue context c, a training sample is

---

[2] Please note that c comprises two beginning turns of dialogue generated in Section 2.1 Step 3 and the first utterance user-generated in Step 4, making a total of 5 utterances.

formed by pairing a topic t^+ from T^+ with a corresponding negative topic t^-, which is randomly selected from T^-. The ranking model is implemented by appending a randomly initialized linear head to predict a scalar value. We then estimate the parameters of the ranking model by optimizing the maximum likelihood loss, defined as follows:

$$\mathcal{L}(\theta, \mathcal{D}) = \mathbb{E}_{(c,t^+,t^-)\sim\mathcal{D}}[\log(1 + e^{r_\theta(c,t^-)-r_\theta(c,t^+)})]$$

where $r_\theta(c, t)$ is the scalar output of the ranking model with parameters θ, and $\mathcal{D}$ is the preprocessed dataset of pairwise judgments. During inference, the ranking model outputs a scalar value, such that $P(t^+ \succ t^- \mid c) \propto e^{r_\theta(c,t^+)}$, which is learned through pairwise loss that topic t^+ is preferred over t^- given context c. Thus, topic t^+ is considered superior to t^- when $r_\theta(c, t^+) > r_\theta(c, t^-)$.

Proactive Topic-shifting Detection and Generation. Ultimately, we trained a memory-aware proactive response generation model to proactively lead the current conversation c towards the identified topic t_r through multiple turns of responses $R = \{r_1, r_2, \ldots, r_m\}$ at an appropriate moment τ. The training data for this step is called *ChMapData-Mem*, which comprises historical dialogues with their corresponding topics and the current dialogue as inputs, along with Thoughts and detection for topic shifting, and response content as learning objectives. As previously mentioned, the bot initially assesses whether it is an appropriate time to transition to a historical topic based on the current context c, and provides the reasoning behind this decision as a form of CoT. Subsequently, it generates the response content, with "Yes" or "No" indicating whether the response incorporates memory or is based solely on the current context.

4 Experiments

We design comparative experiments from two perspectives (both individual modules and the entire framework), assess two approaches (RAG-based alongside end-to-end) and utilize different test sets (our new ChMapData-test and an existing dataset).

4.1 Dataset

Our evaluation involved creating a new test set **ChMapData-test**, following the method outlined in Section 2.1. Please refer to Appendix C for the detailed construction process. Additionally, we incorporated the existing Chinese dataset **NaturalConv** (Wang et al., 2021) as conversation history to construct test data, so as to evaluate the method's generalization to unseen topics.

4.2 Compared approaches

In our exploration of the overall framework, we conduct a series of experiments from both RAG-based and end-to-end perspectives. Given that RAG-based methods comprise three components—namely, a module for processing dialogue history, the retriever, and the generator—we have designed four progressive permutations.

- **BGE w/ Qwen:** Widely-used BGE-M3 retrieval model (Chen et al., 2024) retrieves relevant memories from raw dialogue history, with Qwen2.5 generating proactive dialogue responses as a baseline.
- **QSum w/ BGE w/ Qwen:** Compared to BGE w/ Qwen, BGE-M3 retrieves memories from historical topics condensed by our fine-tuned Qwen on the *ChMapData-Ret* dataset, named QSum.
- **QSum w/ QRet w/ Qwen:** Compared to QSum w/ BGE w/ Qwen, the retrieval model is replaced with our fine-tuned QRet.
- **QSum w/ QRet w/ QMem (Ours):** Fine-tuned QMem that has topic-shifting capability represents the dialogue model while using Qsum and QRet.
- **Qwen-E2E:** Fine-tuned Qwen on ChMapData in an end-to-end (E2E) manner, utilizing all original dialogues as references without any intermediate steps such as summarization or retrieved results.
- **GPT4-E2E:** GPT-4, via prompt engineering, generates memory-aware responses.

To compare methods for proactively introducing topics using dialogue history, we use Qwen2.5-7B[3] as the base LLM unless otherwise specified. Implementation details are in AppendixE, and full prompts are in AppendixF. Observations from our ChMapData dataset show that user responses significantly influence the model's ability to transition topics. To prevent subconscious topic steering by human annotators and ensure objectivity while reducing costs, we trained a **User-role Dialogue Model**. For more information, see Appendix G.

4.3 Evaluation Metrics

Following previous works (Yuan et al., 2019; Han et al., 2021), we utilized Recall ($R_{10}@k$) to evaluate topic retrieval module, where the correct topic is among the top k out of ten candidates, specifically using $\boldsymbol{R_{10}@1}$, $\boldsymbol{R_{10}@2}$, and $\boldsymbol{R_{10}@3}$. We also used **MRR** and **NDCG** as additional retrieval

---

[3] https://huggingface.co/Qwen/Qwen2.5-7B

Models	**Arch.**	**Retrieval**	**Achievement**	**Overall Quality**	**Engagement**		**Avg.**	**#Turn**
					Utts.-level	*Sess.-level*		
				ChMapData-test				
BGE w/ Qwen	RAG	per Sess.	0.02	0.89	0.02	0.02	0.34	4.70
		per Utt.	0.01	0.88	0.04	0.02	0.30	5.30
QSum w/ BGE w/ Qwen	RAG	per Sess.	0.04	0.92	0.05	0.05	0.38	4.52
		per Utt.	0.00	0.88	0.05	0.02	0.34	6.02
QSum w/ QRet w/ Qwen	RAG	per Sess.	0.14	0.99	0.04	0.02	0.44	3.34
		per Utt.	0.06	1.00	0.05	0.05	0.44	4.34
Ours	RAG	per Sess.	0.82	1.23	0.34	0.57	**1.18**	3.23
		per Utt.	**0.89**	**1.36**	0.34	**0.60**	**1.18**	3.51
Qwen-E2E	E2E	-	0.39	0.97	0.20	0.37	0.74	2.70
GPT4-E2E	E2E	-	0.80	1.04	**0.50**	0.55	1.11	2.23
				NaturalConv-test				
BGE w/ Qwen	RAG	per Utt.	0.01	0.98	0.02	0.01	0.32	4.32
QSum w/ BGE w/ Qwen	RAG	per Utt.	0.05	1.04	0.05	0.01	0.36	4.03
QSum w/ QRet w/ Qwen	RAG	per Utt.	0.08	1.07	0.07	0.05	0.38	3.98
Ours	RAG	per Utt.	**0.78**	**1.29**	**0.28**	**0.31**	**1.16**	3.83
Qwen-E2E	E2E	-	0.34	0.94	0.18	0.22	0.71	4.23
GPT4-E2E	E2E	-	0.50	1.11	0.22	0.17	0.83	4.47
Kappa			0.76	0.69	0.63	-	0.70	0.70

Table 2: Human evaluation of the proactive dialogue systems on both test sets. We further explored the effectiveness of retrieval once per session and once per utterance in the ChMapData-test. **Achievement** is calculated as the proportion of sessions that successfully shift topics (Score 2). **Overall Quality** is calculated as the average of the total scores for each utterance. **Engagement** at the *utterance-level* is calculated as the average of all scores, while the *session-level* is measured by the proportion of the score of "2" within the session. **Avg.** represents the average of the scores for the first three evaluation metrics. **Bold** indicate the best performance, while underlined rank second. Annotator agreement is measured by Cohen's kappa (Cohen, 1960), with κ >0.6 denoting high agreement.

metrics referring Zhao et al. (2024).

Since existing automatic metrics like BLEU and METEOR can't authentically reflect the quality of responses (Cai et al., 2019; Yang et al., 2022), we evaluate overall performance by human annotators. Specifically, we assess the quality of generated responses from each system using a total of 200 entries, with each of the two test sets containing 100. To avoid infinite conversations that never reach the target, we set a maximum of 10 turns per session. Three annotators score the generated dialogues on a scale of {0, 1, 2} with higher scores indicating better quality, based on three evaluation criteria at both the utterance and session levels. Annotation details are given in Appendix H.

- **Engagingness:** An utterance-level metric measuring chatbots' ability to capture and respond to user attributes and preferences, thereby fostering ongoing engagement and long-term connections.
- **Overall Quality:** An utterance-level metric assessing the effectiveness of the response, focusing on smooth topic transitions, coherence, and consistency with the given context.
- **Achievement:** A session-level metric determining whether the chatbot has successfully transitioned the conversation to a relevant history. "1" indicates the topic is merely mentioned, and "2" signifies a successful transition. **#Turn**[4] is the turn number when mention or transition occurred.

5 Result and Analysis

5.1 Quantitative Analysis

RAG-based surpass end-to-end methods. As demonstrated in Table 2, our RAG-based framework significantly outperforms the end-to-end trained Qwen and even exceeds the powerful GPT-4 across almost all metrics on both test sets, which effectively highlights the necessity of our framework. Although GPT4-E2E excels in utterance-level engagement, our review of its responses reveals that

---

[4] Value 0 if non-achievement, otherwise 1 to 10.

Current Context	Models	Response
User: I've been reading a lot about computer programming and game design these past few days. I'm really interested in both, but I can't decide which one to focus on. **Chatbot:** You could start by doing some short-term projects or taking a few courses in both areas. You can get a feel for what working in each field is like. **User:** Can you recommend some beginner-friendly projects or courses for computer programming and game design? **Chatbot:** Sure! For programming, you might want to try Codecademy's Python course or Coursera's "Computer Science 101." As for game design, you could check out Coursera's "Introduction to Game Design" or Udemy's "Complete C# Unity Developer 2D" course. These are all great for beginners. **Historical Topic** 1. User's bicycle was stolen. 2. User supports environmental protection. 3. Chatbot shares anecdotes and jokes with User. 4. User is interested in the recent rocket launch. 5. User is learning piano. 6. User is job hunting. 7. User supports environmental protection. 8. Chatbot shares anecdotes and jokes with User. 9. User is interested in the recent Olympics.	**QSum w/ BGE w/ Qwen**	**User:** You're right, I'll keep your advice in mind. **Chatbot:** I'm glad I could help. Having clear career interests is a very important factor. Choosing a field you're passionate about will make you more dedicated to learning and exploring, leading to better results. Before making a decision, it might be helpful to gather more information and learn some related knowledge to make a more informed choice.
	QSum w/ QRet w/ Qwen	**User:** Thank you. Your advice is very useful, and I've decided to find my true interest through practical experience. **Chatbot:** You're welcome. Keep going, and don't let difficulties stop you from pursuing your interests and dreams. If you encounter any problems during your practice, feel free to reach out to me anytime. I wish you the best in finding your ideal job and achieving your goals.
	Ours	**User:** These courses sound really interesting, I'll give them a try. **Chatbot:** Great, this will also be very helpful for your job search, as many companies in programming and game design require these skills. When you're job hunting, you can apply to companies in both fields and see which one suits you better. You can also specify in your resume which area you are more interested in, making your application more targeted.
	GPT4-E2E	**User:** Okay, I'll start learning tomorrow. I'll first get to know the specifics of computer programming and game design, and then decide which direction to pursue. **Chatbot:** That sounds like a solid plan. Remember, learning is a lifelong journey, not just for finding a job but also for enriching yourself and pursuing your interests. Just like the job search process, it's important to try to build new connections. They might offer valuable advice on skill development or deepen your understanding of these fields.

Table 3: Case study on the test set translated into English. The most relevant historical topic is highlighted in Orange. The response shows content after several turns of conversation, where Red indicates a successful topic shift, Blue signifies a mere mention of the relevant topic, and Green refers to semantic logic issues.

it achieves this by repeatedly referencing historical topics, which can appear unnatural across a 10-turn session and degrade overall quality. Additionally, GPT4-E2E's topic shifts occur in fewer turns, but fewer turns do not inherently imply superior performance, particularly in transition smoothness. Analysis in Appendix J shows no direct correlation between the number of turns and model performance.

Each component is essential. Table 2 illustrates steady improvements among the first four RAG-based systems, highlighting the effectiveness of each component in our framework. By introducing QSum and replacing widely-used BGE with QRet, Qwen can utilize a more effective dialogue history for proactive conversation, thereby avoiding abrupt topic shifts and enhancing overall dialogue quality. This results in a gradual improvement across various metrics. Furthermore, QMem, which controls the final generation, shows significant performance enhancement when combined with the first two modules, achieving optimal performance. Table 4 further compares the performance of QSum and QRet in retrieving relevant dialogue history. QSum significantly outperforms raw dialogue and keyword summaries when cooperating with BGE. Furthermore, QRet enhances this effect, even when ranking keywords instead of the summaries used during training. The independent evaluation of the abstract is presented in Appendix D.

Retrieval Combination	R@1	R@2	R@3	MRR	NDCG
Raw dialogue w/ BGE	0.76	0.86	0.92	0.84	0.88
Keywords[5] w/ BGE	0.70	0.82	0.88	0.81	0.86
Keywords w/ QRet	0.77	0.86	0.91	0.87	0.92
QSum w/ BGE	0.78	0.86	0.95	0.85	0.88
QSum w/ QRet	**0.82**	**0.95**	**0.97**	**0.90**	**0.93**

Table 4: Retrieval performance of various combinations.

Moreover, we integrated our model into a real dialogue system, achieving a 5.1-turn improvement in user interactions, shown in Appendix I.

5.2 Qualitative Analysis

Table 3 presents a case study of four models from the ChMapData-test. After successfully retrieving highly relevant historical topics, the original Qwen

---

[5] The tool we employ to extract keywords from the raw dialogue is https://github.com/jeekim/fasttextrank.

Models	**Achievement**	**Overall Quality**	**Engagement**		**Avg.**	**#Turn**
			Utts.	*Sess.*		
BGE w/ QMem	0.57	0.83	0.14	0.39	0.72	3.71
QSum w/ BGE w/ QMem	0.60	0.95	0.25	0.41	0.81	3.49
QSum w/ QRet w/ Qwen(7B)	0.06	1.00	0.05	0.05	0.44	4.34
QSum w/ QRet w/ Qwen(72B)	0.43	1.21	0.11	0.35	0.77	2.91
Ours	**0.89**	**1.36**	**0.34**	**0.60**	**1.18**	3.51

Table 5: Ablation study of different components.

models merely mentioned historical topics without achieving topic transitions, which reflects its lack of proactive conversation capabilities. In contrast, our model makes smooth transitions from the current context to the historical topic, i.e., moving from "how it helps with job hunting" to "specific job hunting tips". For GPT4-E2E, although it mentioned historical topics, the link between "learning computer programming and game design" and "building new connections" was tenuous, leading to incoherence and logical issues. GPT4-E2E tends to mention historical topics compared to other models but deviates from proactive topic shifts, which is also shown statistically in Appendix J Table 9. This contributes to its inferior performance compared to our model, as shown in Table 2.

5.3 Ablation Study

In this section, we systematically replace each component of our model to examine their impacts. The results, presented in Table 5, confirm the effectiveness of all three modules through pairwise comparisons. Notably, the dialogue model exerts the most significant influence on system performance. Compared to models 3 and 4, as well as our own, even with advanced prompt engineering using the superior Qwen2.5, achieving effective topic transitions remains challenging. This limitation persists despite substantially larger parameter sizes, resulting in less achievement and engagement. The performance boost observed with our QMem further validates the robustness of our constructed dataset.

6 Related Work

Proactive Dialogue System. Deng et al. (2023) categorize proactive dialogue systems into three types: open-domain dialogue (Xu et al., 2021; Kishinami et al., 2022), task-oriented dialogue (Chen et al., 2022; Zhao et al., 2022), and information-seeking dialogue (Aliannejadi et al., 2019; Deng et al., 2022). Unlike the latter two, which focus on accomplishing specific tasks within certain domains, proactive open-domain dialogue systems strive to engage users by proactively introducing topics or posing questions, thereby creating a more dynamic and interactive conversational experience. Our work is centered on proactive open-domain conversation. Nevertheless,we observe that existing works primarily emphasize coherence (Xu et al., 2021), smoothness (Zhong et al., 2021; Kishinami et al., 2022), and achievement (Kishinami et al., 2022) within several turns of a session, yet none have been designed to craft systems capable of recalling and effectively leveraging historical dialogue context, a key aspect in sustaining continuity and intelligence in extended conversations.

Long-Term Memory. Memory architectures have typically been a core component of conversational agents (Elvir et al., 2017). Previous long-term dialogue systems (Kim et al., 2015; Bang et al., 2015; Elvir et al., 2017) mainly relied on rule-based frameworks, utilizing episodic memory structures to extract, store, and manage relevant facts from prior interactions, thereby enhancing the coherence of ongoing dialogues (Campos et al., 2018). Subsequent studies focus on large-scale pre-trained models. Xu et al. (2022a) identify their limitations in long-term conversations and introduce a dataset for multi-session engagement. Xu et al. (2022b) present a Chinese dialogue dataset and a framework that integrates long-term memory to enhance persona-based dialogue without multi-session training data. Building upon prior research, we create novelty in terms of incorporating the long-term memory mechanism into proactive dialogue systems, serving as an initial step towards history-aware proactive dialogue systems.

7 Conclusion and Future Work

In this paper, we incorporate memory mechanisms into proactive dialogue systems and propose the novel MapDia task. We break it down into three subtasks and develop an automated methodology for data construction, resulting in the first Chinese dataset for memory-aware proactive dialogue. We further introduce a RAG-based framework to address these subtasks: topic extraction from dialogue history, relevant topic retrieval, and context transition to historical conversations. Our experiments validate the effectiveness of our methodology and models, showing that our framework, combined with a 7B LLM, outperforms the GPT-4 model. In future work, we will explore automatic evaluation methods for MapDia to simplify research costs.

8 Limitations

Despite extensive experimental validation of the framework's effectiveness, the inclusion of multiple components may lead to increased response times for the Chatbot. Further research is expacted to explore a lightweight framework that balances efficiency and effectiveness. Furthermore, the ChMapDia dataset we developed is restricted to Chinese contexts and focuses solely on the scope of casual conversations. A general conversational agent should ideally be multilingual, cover multiple domains, and integrate various personalized styles. Additionally, the dataset contains fewer than 2,000 entries, which could restrict the model's performance. Due to computational limitations, we only used a 7B model; however, employing a larger-scale dialogue model could improve response quality, as indicated in Table 5.

9 Ethics Statement

We first discuss the ethical implications related to generative dialogue agents, particularly in interactive systems with memory awareness.

- Our work aims to enhance the proactivity of dialogue systems within the bounds of user authorization, in line with other LLM-based dialogue applications like ChatGPT and Character.ai, without increasing ethical risks such as user privacy.
- While repeatedly bringing up negative historical events may adversely impact users with psychological disorders and increase anxiety, appropriately addressing these negative memories can have therapeutic benefits as well. Cognitive Behavioral Therapy (CBT) and Exposure Therapy (ET) both emphasize the benefits of structured revisitation of past experiences to mitigate their negative impact and develop healthier coping strategies (Beck, 2020; Foa and Kozak, 1986). Similarly, studies on the Emohaa Chatbot demonstrate the potential of dialogue systems to alleviate mental distress with proper emotional support (Sabour et al., 2023). Thus, it is essential to balance the exploration of past memories, necessitating collaboration between technologists and psychologists to use memory-related technologies effectively and safely.
- Conversational agents that can convincingly mimic human interactions risk users forming parasocial relationships, and potentially affecting their lives adversely. Additionally, the processes of memory summarization and dialogue generation may propagate misinformation or social biases. We recommend that any practical deployment of our frameworks should be prefaced with a disclaimer about the source of the dialogues.
- Our research focuses solely on the memory recall capabilities of models in proactive dialogues and does not involve actual policy recommendations. The proposed framework cannot substitute for genuine real-world interactions, and we do not make any recommendations for users to make real-world decisions that could affect human lives based on our framework.

We also considered the ethical issues related to annotation and datasets. We recruit annotators from a Chinese university, allowing them complete freedom to choose whether or not to participate in our annotation project. The payment is 9 dollars per hour, higher than the local minimum wage. We have reviewed the data prior to annotation and found no biased samples or toxic information generated by the model. Any data that could potentially identify participants has been deleted after the annotation process. Additionally, we have verified the licenses of the artifacts used in this study and found no conflicts. The license of the dataset we will release is CC BY-NC 4.0.

References

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

Priyanka Agrawal, Chris Alberti, Fantine Huot, Joshua Maynez, Ji Ma, Sebastian Ruder, Kuzman Ganchev, Dipanjan Das, and Mirella Lapata. 2022. Qameleon: Multilingual qa with only 5 examples. *arXiv preprint arXiv:2211.08264*.

Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W Bruce Croft. 2019. Asking clarifying questions in open-domain information-seeking conversations. In *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*, pages 475–484.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jeesoo Bang, Hyungjong Noh, Yonghee Kim, and Gary Geunbae Lee. 2015. Example-based chat-oriented dialogue system with personalized long-term memory. In *2015 International Conference on*

Big Data and Smart Computing (BIGCOMP), pages 238–243. IEEE.

Judith S Beck. 2020. *Cognitive behavior therapy: Basics and beyond*. Guilford Publications.

Yonatan Bitton, Shlomi Cohen-Ganor, Ido Hakimi, Yoad Lewenberg, Roee Aharoni, and Enav Weinreb. 2023. q2d: Turning questions into dialogs to teach models how to search. *arXiv preprint arXiv:2304.14318*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019. Retrieval-guided dialogue response generation via a matching-to-generation framework. 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 1866–1875.

Joana Campos, James Kennedy, and Jill F Lehman. 2018. Challenges in exploiting conversational memory in human-agent interaction. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1649–1657.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Zhiyu Chen, Bing Liu, Seungwhan Moon, Chinnadhurai Sankar, Paul Crook, and William Yang Wang. 2022. KETOD: Knowledge-enriched task-oriented dialogue. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2581–2593, Seattle, United States. Association for Computational Linguistics.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37 – 46.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. 2023. A survey on proactive dialogue systems: Problems, methods, and prospects. *arXiv preprint arXiv:2305.02750*.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2022. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6970–6984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Miguel Elvir, Avelino J Gonzalez, Christopher Walls, and Bryan Wilder. 2017. Remembering a conversation–a conversational memory architecture for embodied conversational agents. *Journal of Intelligent Systems*, 26(1):1–21.

Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*.

Edna B Foa and Michael J Kozak. 1986. Emotional processing of fear: exposure to corrective information. *Psychological bulletin*, 99(1):20.

Prakhar Gupta, Harsh Jhamtani, and Jeffrey P Bigham. 2022. Target-guided dialogue response generation using commonsense and data augmentation. *arXiv preprint arXiv:2205.09314*.

Janghoon Han, Taesuk Hong, Byoungjae Kim, Youngjoong Ko, and Jungyun Seo. 2021. Fine-grained post-training for improving retrieval-based dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1549–1558.

Dongshi Ju, Shi Feng, Pengcheng Lv, Daling Wang, and Yifei Zhang. 2022. Learning to improve persona consistency in multi-party dialogue generation via text knowledge enhancement. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 298–309.

Yonghee Kim, Jeesoo Bang, Junhwi Choi, Seonghan Ryu, Sangjun Koo, and Gary Geunbae Lee. 2015. Acquisition and use of long-term memory for personalized dialog systems. In *Multimodal Analyses enabling Artificial Agents in Human-Machine Interaction: Second International Workshop, MA3HMI 2014, Held in Conjunction with INTERSPEECH 2014, Singapore, Singapore, September 14, 2014, Revised Selected Papers 2*, pages 78–87. Springer.

Yosuke Kishinami, Reina Akama, Shiki Sato, Ryoko Tokuhisa, Jun Suzuki, and Kentaro Inui. 2022. Target-guided open-domain conversation planning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 660–668, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Alisa Liu, Swabha Swayamdipta, Noah A. Smith, and Yejin Choi. 2022. WANLI: Worker and AI collaboration for natural language inference dataset creation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6826–6847, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. *arXiv preprint arXiv:2005.03954*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. 2020. Politeness transfer: A tag and generate approach. *Preprint*, arXiv:2004.14257.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Sahand Sabour, Wen Zhang, Xiyao Xiao, Yuwei Zhang, Yinhe Zheng, Jiaxin Wen, Jialu Zhao, and Minlie Huang. 2023. A chatbot for mental health support: exploring the impact of emohaa on reducing mental distress in china. *Frontiers in digital health*, 5:1133987.

Jianheng Tang, Tiancheng Zhao, Chenyan Xiong, Xiaodan Liang, Eric P. Xing, and Zhiting Hu. 2019. Target-guided open-domain conversation. *Preprint*, arXiv:1905.11553.

Xiaoyang Wang, Chen Li, Jianqiao Zhao, and Dong Yu. 2021. Naturalconv: A chinese dialogue dataset towards multi-turn topic-driven conversation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14006–14014.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019a. Proactive human-machine conversation with explicit conversation goals. *arXiv preprint arXiv:1906.05572*.

Wenquan Wu, Zhen Guo, Xiangyang Zhou, Hua Wu, Xiyuan Zhang, Rongzhong Lian, and Haifeng Wang. 2019b. Proactive human-machine conversation with explicit conversation goals.

Jing Xu, Arthur Szlam, and Jason Weston. 2022a. Beyond goldfish memory: Long-term open-domain conversation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5180–5197, Dublin, Ireland. Association for Computational Linguistics.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. Enhancing dialog coherence with event graph grounded content planning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3941–3947.

Xinchao Xu, Zhibin Gou, Wenquan Wu, Zheng-Yu Niu, Hua Wu, Haifeng Wang, and Shihang Wang. 2022b. Long time no see! open-domain conversation with long-term persona memory. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2639–2650, Dublin, Ireland. Association for Computational Linguistics.

Yan Xu, Deqian Kong, Dehong Xu, Ziwei Ji, Bo Pang, Pascale Fung, and Ying Nian Wu. 2023. Diverse and faithful knowledge-grounded dialogue generation via sequential posterior inference. In *International Conference on Machine Learning*, pages 38518–38534. PMLR.

Yizhe Yang, Heyan Huang, Yang Gao, and Jiawei Li. 2024. Building knowledge-grounded dialogue systems with graph-based semantic modeling. *Knowledge-Based Systems*, page 111943.

Zhitong Yang, Bo Wang, Jinfeng Zhou, Yue Tan, Dongming Zhao, Kun Huang, Ruifang He, and Yuexian Hou. 2022. Topkg: Target-oriented dialog via global planning on knowledge graph. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 745–755.

Chunyuan Yuan, Wei Zhou, Mingming Li, Shangwen Lv, Fuqing Zhu, Jizhong Han, and Songlin Hu. 2019. Multi-hop selector network for multi-turn response selection in retrieval-based chatbots. 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 111–120.

Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2024. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60.

Xinyan Zhao, Bin He, Yasheng Wang, Yitong Li, Fei Mi, Yajiao Liu, Xin Jiang, Qun Liu, and Huanhuan Chen. 2022. UniDS: A unified dialogue system for chit-chat and task-oriented dialogues. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 13–22, Dublin, Ireland. Association for Computational Linguistics.

Peixiang Zhong, Yong Liu, Hao Wang, and Chunyan Miao. 2021. Keyword-guided neural conversational model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14568–14576.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

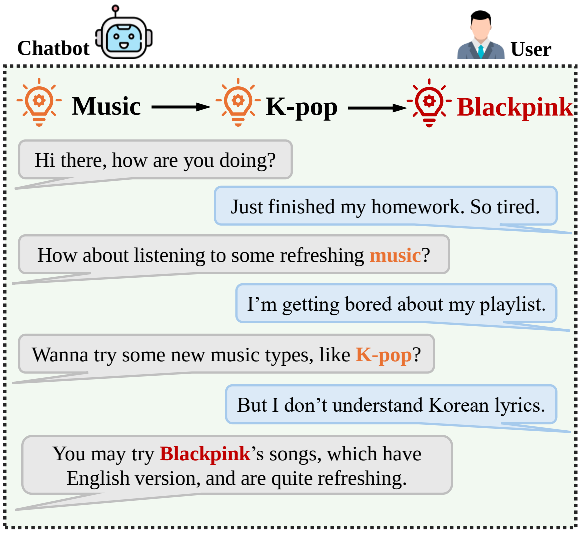
Hao Zhou, Chujie Zheng, Kaili Huang, Minlie Huang, and Xiaoyan Zhu. 2020. Kdconv: A chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation.

A A Sample of Proactive Dialogue

Here, we showcase a real example in Figure 4 taken from a prior proactive dialogue system (Deng et al., 2023). Despite the user clearly showing disinterest in Korean lyrics, the chatbot still pushed the conversation towards BlackPink. Although it reached the target, it failed to engage the user for long-term interaction. This pattern is counterproductive to developing an intelligent dialogue agent.

B Prompt for Data Construction

The complete prompt templates used for constructing the dataset are shown in Figure 5, and the corresponding English versions are listed subsequently in Figure 6. Specifically, Prompt A is used to generate ChMapData-Ret, encompassing dialogue history and corresponding topics, whereas Prompts B and C are each connected to creating the initial two turns of the current dialogue. Prompt D corresponds to the subtask of proactive topic-shift detection and response generation. To stimulate the generative capabilities of LLMs, we experimented with various prompting techniques. Inspired by the sensitivity of language models to the framing of their instructional prompts (Mishra et al., 2022), we organized the instructions with bullet points to improve the model's understanding of the tasks. Additionally, we employed the one-shot strategy in Prompt A to guide the model in generating dialogue more effectively. The desired output format is also specified for each type of prompt.



Hi there, how are you doing?

Just finished my homework. So tired.

How about listening to some refreshing music?

I'm getting bored about my playlist.

Wanna try some new music types, like K-pop?

But I don't understand Korean lyrics.

You may try **Blackpink**'s songs, which have English version, and are quite refreshing.

Figure 4: A sample of previous proactive dialogue system extracted from Deng et al. (2023).

	ROUGE-1	ROUGE-2	ROUGE-L	BLEU$^{1-4}$	METEOR
Qwen2.5	0.522	0.333	0.467	0.197	0.414
Qwen2.5-Sum	**0.773**	**0.646**	**0.746**	**0.536**	**0.755**

Table 6: Comparison results of the Qwen model with and without fine-tuning on our ChMapData-Sum dataset.

C ChMapData-test Construction

The test set construction process is consistent with Section 2.1. Initially, we generated 400 dialogues from 11 topics, ensuring the same ratio of Memorable and General data in the training set. Subsequently, 150 dialogues under the Memorable Subject were extended by two turns, serving as the beginning of the current context. Consistent with the trainset construction, we utilized the corresponding topics generated in the first step and chose other 29 historical topics at random as candidates to rank against the current context. Both the corresponding topic and the top-ranked topic were utilized as ground truth, as they each acted as positive examples in the training set. Additionally, 2 to 11 dialogues were randomly chosen from topics ranked below the corresponding topic (i.e., negative examples) to serve as dialogue history. This process resulted in 112 entries[6], out of which 100 were finalized for the test set, each comprising dialogue history, current dialogue, and a ground truth topic.

---

[6] Excluding 22 entries with fewer than two topics ranked below the corresponding and 16 items modified after ranking.

Prompt A

[系统指令]
你的任务是按照以下示例构建一段五到八轮对话，对话的人物是用户和聊天机器人，对话的主角是用户。

[对话示例]
主题：自身的兴趣
细化话题：用户对编程的兴趣
用户：你好，我最近对编程产生了很大的兴趣。
聊天机器人：嗨，对编程感兴趣是一件很棒的事情，编程可以帮助你解决许多问题，增强逻辑思维，还可以开发出有趣的应用或者游戏。你对哪种编程语言比较感兴趣呢？
用户：我听说Python比较容易学习，适合我这种初学者，所以我想试试Python。
聊天机器人：你的选择非常好，Python的确是一种易于上手的编程语言，而且非常强大，被广泛应用于各种领域，包括数据分析、机器学习、网站开发等等。你打算如何开始学习呢？
用户：我正在网上找一些教程和实战项目，但是感觉有点乱，不知道该如何系统地学习。
聊天机器人：针对这个问题，我建议你可以参考一些学习路径，比如先学习Python的基础语法，然后再学习一些常用的库，如Numpy、Pandas等，然后可以找一些实战项目来练习。同时，你还可以参加一些编程社区，如GitHub、Stack Overflow等，这些社区里有很多优秀的编程项目和问题讨论，对你的学习会有很大帮助。
用户：谢谢你的建议，我会按照你的建议来学习的。
聊天机器人：不客气，相信你一定可以学好Python的。如果在学习过程中遇到任何问题，都可以来找我讨论，我会尽力帮你解答的。

[对话主题]
{候选主题}

[对话格式]
主题：{候选主题}
细化话题：{细化话题}
用户：{用户发言}
聊天机器人：{聊天机器人发言}

[生成结果]
{}

Prompt B

[系统指令]
你的任务是依据历史对话，续写用户和聊天机器人在多天以后的对话。
要求如下：
1. 对话由用户先发起，减少用户对话中提问的概率；用户更多的是聊天，由聊天机器人来提问进行话题引导
2. 本段对话与历史对话的内容有潜在相关性，但是不要直接相关；
3. 注意用户和聊天机器人不能主动提到历史对话的主题以及内容；
4. 续写1轮对话, 不需要对话完整结束；

[历史对话]
{Prompt A生成的历史对话}

[续写格式]
用户：{用户发言}
聊天机器人：{聊天机器人发言}

[续写结果]
{}

Prompt C

[系统指令]
你的任务是依据对话开头，续写一轮用户和聊天机器人的对话。

[对话开头]
{Prompt B生成的对话开头}

[续写格式]
用户：{用户发言}
聊天机器人：{聊天机器人发言}

[续写结果]
{}

Prompt D

[系统指令]
你的任务是依据细化话题、历史对话以及当前对话，续写对话至结束。
要求如下：
1. 对话由用户先发起，继续当前对话，用户不能主动提及历史对话与细化话题；减少用户对话中提问的概率，用户更多的是聊天，由聊天机器人来提问进行话题引导；
2. 每轮对话均为一问一答的形式，由用户开始，聊天机器人结束；
3.聊天机器人需要依据用户的回复，判断是否将对话主题转移至历史对话上，如果当前主题不适合切换，则继续聊下去，最后整个对话无法进行主题切换也是符合要求的，不要强行切换；
4. 在聊天机器人回答前，把聊天机器人对主题切换时机的判断以Thoughts的形式打出来，并在最后标记是否进行话题切换，用Yes和No来表示， Yes表示可以切换， No表示不适合切换；
5. 保证对话完整结束，续写5轮对话以上。

[历史对话]
{Prompt A生成的历史对话}

[当前对话]
{Prompt B和Prompt C生成的当前对话}

[续写格式]
用户：{用户发言}
Thoughts：{聊天机器人思考当前是否切换回历史话题}
{Yes or No ，根据Thoughts判定切换的结果}
聊天机器人：{聊天机器人发言}

[续写结果]
{}

Figure 5: The full prompt template utilized for data construction in Section 2.1 with step 2 corresponding to prompt A, steps 3 corresponding to prompts B and C, and step 4 corresponding to prompt D.

Prompt A

[Instructions]
Your task is to construct a dialogue consisting of five to eight turns, following the example provided below. The characters in the dialogue are User and Chatbot, with the user being the main character.

[Example]
Subject: Personal Interests
Topic: User's interest in programming
User: Hello, I've recently developed a great interest in programming.
Chatbot: Hi, having an interest in programming is fantastic. It can help you solve many problems, enhance logical thinking, and even develop interesting applications or games. Which programming language are you particularly interested in?
User: I've heard that Python is relatively easy to learn and suitable for beginners like me, so I want to give Python a try.
Chatbot: That's a great choice. Python is indeed an accessible programming language and very powerful. It's widely used in various fields, including data analysis, machine learning, web development, and more. How do you plan to start learning?
User: I'm looking for some tutorials and hands-on projects online, but it feels a bit chaotic, and I'm not sure how to learn systematically.
Chatbot: For this issue, I suggest you could follow some learning paths, such as starting with the basics of Python syntax, then moving on to some commonly used libraries like Numpy and Pandas, and then practicing with some hands-on projects. Also, you can join some programming communities, like GitHub or Stack Overflow, where there are many excellent programming projects and discussions that can greatly help your learning.
User: Thank you for the advice, I will follow your suggestions.
Chatbot: You're welcome. I'm confident that you'll master Python. If you encounter any problems during your learning process, feel free to discuss them with me, and I'll do my best to help you find the answers.

[Subject]
{Candidate subject}

[Format]
Subject: {Candidate subject}
Topic: {Topic}
User: {User’s utterance}
Chatbot: {Chatbot's utterance}

[Generated Result]
{}

Prompt B

[Instructions]
Your task is to continue a conversation between the User and Chatbot that takes place several days after the given historical dialogue. The requirements are as follows:
1. The dialogue should be initiated by the User, with a reduced probability of the User asking questions; the User should engage more in chatting, with the Chatbot asking questions to guide the topic.
2. The content of this dialogue should be potentially related to the historical dialogue but not directly related.
3. Be mindful that neither the User nor the Chatbot should actively mention the topics or content of the historical dialogue.
4. Continue the dialogue for one turn; the conversation does not need to be fully concluded.

[Dialogue History]
{Dialogue history generated by Prompt A}

[Format]
User: {User's utterance}
Chatbot: {Chatbot's utterance}

[Continuation Result]
{}

Prompt C

[Instructions]
Your task is to continue a turn of dialogue between the User and Chatbot based on the beginning of the conversation.

[Dialogue Beginning]
{Dialogue beginning generated by Prompt B}

[Format]
User: {User's utterance}
Chatbot: {Chatbot's utterance}

[Continuation Result]
{}

Prompt D

[Instructions]
Your task is to continue the conversation based on the refined topic, dialogue history, and current conversation until the end.
The requirements are as follows:
1. The conversation should be initiated by the User, continuing the current dialogue. The User should not actively mention historical dialogue or refined topics; reduce the likelihood of questions in the User's dialogue, as the User is more engaged in chatting, with the chatbot asking questions to guide the topic;
2. Each turn of dialogue should be in a question-and-answer format, starting with the User and ending with the Chatbot;
3. The Chatbot needs to determine whether to shift the conversation topic to the historical dialogue based on the User's reply. If the current topic is not suitable for switching, then continue the conversation. It is also acceptable if the entire dialogue does not undergo a topic switch; do not force a switch;
4. Before the Chatbot responds, express the chatbot's judgment on the timing of the topic switch in the form of Thoughts, and mark at the end whether to switch topics, using Yes and No to indicate. Yes means a switch is possible and No means it is not suitable to switch;
5. Ensure the conversation is fully concluded, continuing for more than 5 turns of dialogue.

[Dialogue History]
{Dialogue history generated by Prompt A}

[Current Conversation]
{Dialogue beginning generated by Prompt B and C}

[Continuation Format]
User: {User's utterance}
Thoughts: {Chatbot's thoughts on whether to switch back to the historical topic}
{Yes or No, based on the Thoughts' determination of the switch}
Chatbot: {Chatbot's utterance}

[Continuation Result]
{}

Figure 6: English version of prompt for data construction in Figure 5.

D Comparison Result for Summarization Model

We compared the performance of our Qwen2.5-Sum, a fine-tuned version of Qwen2.5 on our *ChMapData-Sum*, against the original Qwen2.5 model. We reported standard automated metrics including BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). Specifically, we provided the full BLEU score, which accounts for overlap across 1-4 grams, rather than just BLEU-4. The results, as shown in Table 6, indicate that Qwen2.5-Sum significantly outperforms the original Qwen2.5 model, demonstrating the effectiveness of our summarization module.

E Implementation Details

For experiments on Topic Summarization, Topic Retrieval, and Proactive Topic-shifting Detection and Generation, we utilized the AdamW optimizer (Loshchilov and Hutter, 2019). The training setup included a cosine learning rate schedule starting at 2e-5, a weight decay of 0.1, a batch size of 64, a 5% warm-up period, and a maximum input length of 2048 tokens. We fine-tuned all the models for 2 epochs.

F Prompt for Proactive Dialogue Models

In Figure 7, we present the full prompt templates for the two models, Qwen2.5 and GPT-4, which function as memory-aware proactive dialogue systems in Section 4.2. Additionally, Figure 8 illustrates the prompt used within the BGE w/ Qwen framework for the original Qwen2.5 model, guiding it to generate proactive dialogue responses.

G Details of User-role dialogue model

We additionally trained a dialogue model to simulate user interactions during model testing. This approach helps to avoid the subjective factors that annotators might introduce during conversations, which could affect the guidance of active topics.

The parameters of the user-role dialogue model are consistent with those in Appendix E. We utilize Qwen2.5-7B as the base model and the data used to train the user model consists of 4,000 dialogue histories generated in Section 2.1 Step 1. We performed additional processing on the data by converting the training target to the query rather than the response. Moreover, to ensure that the user model does not prematurely end the conversation, we removed the last round from the training data, as this turn typically signifies the end of the conversation.

H Human Annotation Details

Table 10 presents our full annotation guidelines used for the human annotation process in this work.

We recruited six college students who are native Chinese speakers, including four females and two males, with an average age of around 24. Initially, they were provided with an annotation guideline. Each evaluator underwent a training process to enhance their understanding of the annotation procedure. Before starting the annotation, we designed a qualification test consisting of 10 dialogues; only those who passed the test were deemed qualified and allowed to proceed with the annotation To ensure the quality of the annotations, we divided the dataset into batches and assigned a specific number of daily tasks to each annotator. Upon receiving the daily annotations, we reviewed the results and required annotators to reannotate the batch of data assigned for that day if there was low agreement (less than 0.6).

In the annotation interface, the dialogue history, summarized topic, and current context were presented on the left side, while the dialogues generated by each model were randomly displayed on the right to prevent bias. Annotators first read each chatbot's utterance and then assigned scores for "Engagingness" and "Overall quality." After completing the entire session, they assessed "Achievement" and "#Turn." The score range for the first three evaluation criteria was {0,1,2}, while the range for "#Turn" was 0-10.

Each sample was annotated by two distinct annotators, and a third annotator made the final decision in case of disagreement. We utilized Cohen's kappa (Cohen, 1960) to measure inter-annotator agreement. The annotation process lasted approximately two weeks, culminating in a substantial inter-annotator agreement with Cohen's kappa of κ=0.70, as shown in Table 2.

I Integration Testing

Given the novel method proposed in this paper, which can detect conversation trajectories and initiate proactive topics based on dialogue history, it is highly adaptable for integration with any existing dialogue system. To assess its effectiveness, we

[系统指令]
假如你是智能聊天机器人，正在与用户对话，你的任务是依据给定的对话历史与当前对话，对用户进行回复。回复分为两部分：
1.Thoughts：首先判断当前对话与哪一天历史对话可能有潜在联系，接着判断是否可以将对话主题转移到历史对话主题上，如果话题联系度较高，则可以主动进行历史话题转换，并输出Yes。如果话题联系度不高，则无法进行话题转换，并输出No。
2.聊天机器人回复：输出聊天机器人的回复内容，根据判断的Yes或No生成对应的是否进行历史话题转换的回复。

[对话示例]
Thoughts：当前对话提及了跑步，可能和历史对话中用户参加马拉松比赛有关，可以将对话转移到历史对话中。Yes
聊天机器人：说起跑步，上次你参加马拉松比赛怎么样呀？

[对话历史]
{过去若干天用户与聊天机器人的对话历史}

[当前对话]
{当前用户与聊天机器人的对话内容}

[生成结果]
{}

[Instructions]
You are an intelligent Chatbot engaging in a conversation with a user. Your task is to reply to the user based on the given dialogue history and the current context. The reply should consist of two parts:
1. Thoughts: First, determine if the current conversation has any potential connection with a past conversation from a specific day. Then, decide whether the conversation topic can be shifted to the topic of the historical dialogue. If the topic relevance is high, you can proactively transition to the historical topic and output "Yes." Otherwise, you cannot transition the topic and output "No."
2. Chatbot Response: Output the content of the Chatbot's response. Generate a response based on the decision of "Yes" or "No " indicating whether to transition the topic to the historical conversation.

[Example]
Thoughts: The current conversation mentions running, which might be related to the past conversation about the user’s participation in a marathon. The topic can be shifted to the historical conversation. Yes
Chatbot Response: Speaking of running, how was the marathon you participated in last time?

[Dialogue History]
{Dialogue history between the user and the Chatbot over the past few days}

[Current Context]
{Current context between the user and the Chatbot}

[Output]
{}

Figure 7: The prompt template instructs Qwen-2.5 and GPT-4 to act as the en-to-end memory-aware proactive dialogue system. Upper is the original content input into the model, followed by its corresponding English version.

[系统指令]
假如你是智能聊天机器人，正在与用户对话，你的任务是依据细化话题、历史对话以及当前对话，续写聊天机器人的回答。在进行对话的时候，要判断当前对话与哪一天历史对话可能有潜在联系，接着判断是否可以将对话主题转移到历史对话主题上，如果话题联系度较高，则可以主动进行历史话题转换。

[对话历史]
{过去若干天用户与聊天机器人的对话历史}

[当前对话]
{当前用户与聊天机器人的对话内容}

[生成结果]
{}

[Instruction]
You are an intelligent Chatbot engaged in a conversation with a user. Your task is to reply to the user based on the given historical conversations, corresponding topics, and the current context. During the conversation, assess whether there might be potential links to previous days' conversations. If the topic relevance is high, you can proactively switch to the historical topic.

[Dialogue History]
{User and chatbot conversation history with corresponding topics over the past few days}

[Current Context]
{Current context between the user and the chatbot}

[Output]
{}

Figure 8: The prompt template for the original Qwen2.5 is used to generate a proactive dialogue response along with its English version.

Model	**CPS**	**Shift-Ratio**
Original	22.8	-
Original w/ PDia	25.0	20.8%
Original w/ MapDia	**27.9**	12.2%

Table 7: Results of integrated testing, showing the conversation-turns-per-session (CPS) and the triggered ratio of topic shifts per session in a real-world dialogue system. The p-value for the CPS statistic is 0.0074.

conducted an online A/B test by incorporating it into our role-playing dialogue system. The proactive model is trained with combined data of role-playing conversational dataset and ChMapData-Mem to keep the role-playing ability. The model determines when to shift the topic and generates responses for those turns, while the original dialogue system handles other responses.

Additionally, we conducted another integration with trained a proactive responding model only referring to the dialogue context, noted as Proactive Dialogue (PDia). PDia shares the pre-trained base and parameter scale as our proposed model. This model also employs targeted data construction followed by fine-tuning to learn proactive dialogue capabilities. Besides, different from traditional proactive dialogue methods by performing topic planning in advance, we utilize the LLM to dynamically make decisions during the dialogue process.

Given that, users were randomly assigned to one of three groups: one interacting with the original dialogue system, one with the system enhanced by our proposed method, and one with the context-based proactive model. All users were blinded to the system details. Due to commercial constraints, we utilized a closed-source 7B pre-trained model for retraining the proactive dialogue models.

Models	**Engagement**		**Overall Quality**	**Achievement**		**Avg.**
	Utts.-level	*Sess.-level*		Mentioning	Shifting	
#Turn = 1						
Ours per Sess.	0.38	0.71	1.11	0.11	0.89	1.19
Ours per Utts.	0.31	0.58	1.05	0.16	0.84	1.07
GPT-4	0.43	0.46	0.89	0.09	0.91	1.02
#Turn = 2						
Ours per Sess.	0.31	0.50	1.10	0.21	0.79	1.07
Ours per Utts.	0.42	0.55	1.07	0.18	0.82	1.10
GPT-4	**0.55**	0.53	1.12	0.21	0.79	1.13
#Turn = 3						
Ours per Sess.	0.32	0.58	1.18	**0.33**	0.67	1.05
Ours per Utts.	0.35	0.64	1.13	0.21	0.79	1.09
GPT-4	0.43	0.64	0.89	0.21	0.79	1.08
#Turn = 4						
Ours per Sess.	0.31	0.33	1.28	0.22	0.78	1.04
Ours per Utts.	0.38	**0.80**	**1.40**	0.00	**1.00**	**1.21**
GPT-4	0.35	0.43	0.91	0.29	0.71	0.98

Table 8: Evaluation results for each turn number at which the model shifts topics.

Model	**Retrieval Method**	**Mentioning**
QSum w/ BGE w/ Qwen	per Sess.	0.07
	per Utt.	0.08
QSum w/ QRet w/ Qwen	per Sess.	0.08
	per Utt.	0.15
Ours	per Sess.	0.18
	per Utt.	0.14
GPT-4	**-**	0.19

Table 9: Probability of each model mentioning historical topics, calculated as the proportion of label 1 in the Achievement criteria.

The test spanned a duration of two weeks and involved real conversations from over 100,000 users. Table 7 presents the conversation-turns-per-session (CPS), defined as the average number of conversation turns between the dialogue system and the user within a session. The introduction of both proactive topic capabilities significantly enhanced CPS. Specifically, the MapDia model increased the average CPS from 22.8 to 31.3, which is notably higher than that of PDia, indicating that users are more engaged with previously discussed topics when they are properly introduced.

Additionally, it should be noted that the proportion of topic transitions is significantly lower than reported in Table 2. This discrepancy is primarily attributed to the fact that only a small portion of real user dialogues can effectively integrate previously discussed content, and not all conversations require the initiation of proactive topics. Even the PDia model, which incorporates dialogue context, successfully transitions topics in only 20.8% of sessions.

J Analysis of #Turn and Mentioning Metrics

Here, we present the evaluation metrics for #Turns set at 1, 2, 3, and 4 in Table 8. Our model demonstrated the best performance in turn-level retrieval when transitioning topics in the fourth turn. It is observed that fewer #Turns may result in lower overall quality and lower average scores. There is no distinct proportional or inverse correlation between the number of #Turns and the model's overall performance.

Table 9 additionally shows the probability of the model mentioning historical topics without transitioning, which indicates that GPT-4 is more inclined to mention historical topics, which deviates from our task definition.

Human Evaluation Guideline

Task Overview

Thank you for participating in this task! Open-domain dialogue systems are expected to possess the capability to proactively shift conversational topics when necessary. When a chat agent exhausts its conversational material or the current discussion becomes monotonous, topic shifting is a common strategy to maintain the flow of conversation. Furthermore, when the new topic is derived from historical conversations rather than arbitrary subjects, it enhances user engagement and fosters long-term relationships between the chatbot and the user. To achieve this objective, we have developed a Memory-aware Proactive Dialogue system. Below, we provide several days' worth of historical dialogues, along with responses generated by our model and some baseline models. Your task is to evaluate these responses based on the four defined aspects.

Evaluation Aspects

Utterance-level

- **Engagingness:** An utterance-level metric measuring how well the chatbot captures and responds to the user's personal attributes, preferences, and interests, encouraging ongoing participation and long-term connections.
- **Overall Quality:**An utterance-level metric assessing the effectiveness of the response, focusing on smooth topic transitions, coherence, and consistency with the given context.

Session-level

- **Achievement:** A session-level metric determining whether the chatbot has successfully transitioned the conversation to a relevant historical topic. “1” indicates the topic is merely mentioned, and “2” signifies a successful transition.

- **Turn:** A session-level metric represents the turn number when mention or transition occurred.

Annotation Procedure

1. Dialogue History Familiarization: Begin by thoroughly reading and familiarizing yourself with the provided historical dialogues, typically spanning 8-10 days.

2. Current Context Review: Carefully read the initial context of the current dialogue, which includes two beginning turns and a user utterance.

3. Utterance Scoring: Score each response utterance generated by the model on a scale of [0, 1, 2] based on the aspects of Engagingness and Overall Quality. A higher score indicates better performance.

4. Session Scoring: Once the model completes the dialogue continuation, determine whether the entire session achieved a topic shift. Here, 0 indicates no topic shift or mentioning, 1 indicates a mention of a historical topic without shifting, and 2 indicates a complete topic shift. Additionally, note the turn number at which the shift was accomplished.

Emphasis and Caution

- The order of the model-generated responses is randomized to avoid bias.
- It is possible for the model to perform multiple topic shifts within a single session. This strength can be reflected by assigning a score of 2 for Engagingness or Overall Quality at each turn where a topic shift occurs.
- When the topic shift is not natural or smooth, the Overall Quality score should be appropriately reduced, even if the shift was achieved.
- A number of words and phrases are often used as indicators for topic shifts, including but not limited to: "but," "speaking of," "talking about," "anyway," "by the way," "that reminds me," "before I forget," "I want to mention," "let's talk about," "we need to discuss," "funny you should mention that", etc.

Table 10: The full annotation guideline for human evaluation.

WinoWhat: A Parallel Corpus of Paraphrased WinoGrande Sentences with Common Sense Categorization

Ine Gevers[1], Victor De Marez[1], Luna De Bruyne[1], Walter Daelemans[1],
[1]CLiPS, University of Antwerp
Correspondence: ine.gevers@uantwerpen.be

Abstract

In this study, we take a closer look at how Winograd schema challenges can be used to evaluate common sense reasoning in LLMs. Specifically, we evaluate generative models of different sizes on the popular WinoGrande benchmark. We release WinoWhat, a new corpus, in which each instance of the WinoGrande validation set is paraphrased. Additionally, we evaluate the performance on the challenge across five common sense knowledge categories, giving more fine-grained insights on what types of knowledge are more challenging for LLMs. Surprisingly, all models perform significantly worse on WinoWhat, implying that LLM reasoning capabilities are overestimated on WinoGrande. To verify whether this is an effect of benchmark memorization, we match benchmark instances to LLM training data and create two test-suites. We observe that memorization has a minimal effect on model performance on WinoGrande.

1 Introduction

While including common sense knowledge in NLP-systems has been a longstanding goal, evaluating this proves to be a non-trivial task. From early on, research used coreference resolution tasks to measure world knowledge and reasoning abilities in machine learning systems. In 2011, the Winograd Schema Challenge was developed, a small test set of 273 instances in which a pronoun has to be disambiguated given two possible antecedents in a short text (Levesque et al., 2012). Where early models failed, transformer-based models quickly achieved remarkable performance on this test. However, researchers objected that this does not prove that models have or use common sense; rather, they could rely on superficial patterns and dataset artifacts (Kocijan et al., 2023). Therefore, a large adversarial benchmark was created: WinoGrande (Sakaguchi et al., 2021). Here, the challenge is to decide which of two options is the correct one in a fill-in-the-blank token '_'. This benchmark is frequently used in combination with other benchmarks to evaluate the performance of new LLMs on common sense reasoning.

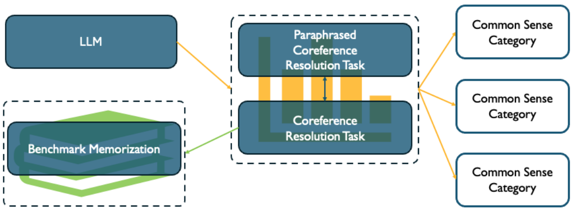


Figure 1: Illustration of the workflow in this study. We evaluate LLMs on WinoGrande, and on its paraphrased variant. We further compare performance per common sense knowledge category, and check for benchmark memorization.

In this study, we evaluate various open-source model families – Gemma 2 (Team et al., 2024), LlaMA 2 (Touvron et al., 2023b), and OPT (Zhang et al., 2022) – on WinoGrande. An overview of the workflow in this study can be found in Figure 1. We present a new parallel corpus of the WinoGrande validation set: WinoWhat, in which we paraphrase each sentence so the '_' token is at the end of the sentence. This transformation makes the task more natural for decoder-only methods and at the same time allows to test whether the performance of LLMs on WinoGrande is robust against paraphrasing (**RQ1**).

While existing works mainly evaluate models on the benchmark as a whole, we define common sense knowledge categories that are crucial to disambiguate the sentence, and evaluate models on each category separately. This allows us to investigate our second research question (**RQ2**): What types of common sense knowledge are more challenging for LLMs? Such an analysis provides insights into the more fine-grained strengths and weaknesses of ML systems on common sense rea-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 68–80
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

soning tasks. Instead of creating new benchmarks to focus on one category of common sense knowledge, we suggest using one task setup, which allows us to compare results on different categories without added noise (e.g., different answer formats, different dataset artifacts, etc.).

To verify to what extent data leakage plays a role in LLMs' performance on WinoGrande, we check how many instances of the validation set are included in the pre-training data of LLMs. Further, we create two test-suites; one from which we know that it has been included in LLM pre-training data (i.e., the Winograd Schema Challenge), and one from which we can assume that it has not been seen (i.e., the WinoGrande test set). Comparably to RQ1, we paraphrase each. This answers **RQ3**: What is the role of data memorization in the performance of LLMs on coreference resolution tasks?

The rest of the paper is structured as follows: in Section 2, we summarize relevant literature about disambiguation tasks, common sense categorization, and benchmark memorization. In Section 3, we present in more detail the data, models, evaluation metrics, and the creation of WinoWhat. Further, in Section 4, we present the results of our experiments, and the final Section 5 concludes our research, giving an overview of the findings and suggestions for further research.

2 Related Work

2.1 Coreference resolution and common sense reasoning

Incorporating common sense knowledge into machine learning methods has been a goal since its very beginning (e.g., McCarthy, 1959). However, given the increasing black-box nature of LLMs, it is hard to evaluate whether models have/use common sense knowledge. From early on, sentence disambiguation tasks have been suggested to measure the ability of models to employ common sense knowledge: the assumption being that syntax alone is not enough for the model, and common sense knowledge is needed to determine for instance which noun a pronoun refers to (Browning and LeCun, 2023). An important concept here is bridging, for which the model needs to make inferences about relationships between entities in the world that are not explicitly mentioned in the text (Kobayashi and Ng, 2020; Hou et al., 2018). Since sentence disambiguation and coreference resolution tasks are presented as a proxy to evaluate common sense knowledge, over the years different approaches have been suggested to teach models common sense in order to improve performance on these tasks. In the early 2000s, most coreference resolvers did not include external knowledge sources, relying instead on morpho-syntactic features. The development of large-scale knowledge bases, which were used as features in a baseline resolver, improved results (Rahman and Ng, 2011). Then, with the advent of larger models and more training data, transformer models also relied on external knowledge bases which are generally stored in triplets (Liu et al., 2023).

2.2 The Winograd Schema Challenge

A popular coreference task is the Winograd Schema Challenge (WSC) (Levesque et al., 2012). Based on the work of Winograd (1972), the challenge uses 'schemas' – pairs of twin sentences whose intended meaning can be flipped by changing one word (the 'special word') – to probe ML-methods' ability to reason over natural language[1]. The schemas have three criteria: (1) simple to solve for humans; (2) not solvable by selectional restrictions (i.e., no statistical advantage for one option); (3) google-proof. Over time, newer versions of the challenge were released, which were made in the same format. These datasets are either created by human annotations, or generated by LLMs. For instance, Zahraei and Emami (2024) use Tree-of-Experts to generate new WSC instances, presenting 3,026 LLM-generated sentences. Similarly, Sun and Emami (2024) present EvoGrad, a hybrid method to generate new adversarial WSC instances that feature minor alterations and rewordings using human annotations, ChatGPT, and WordNet. Since WSC and related benchmarks are in English, the challenge was also translated in other languages such as German, Russian, French, Portuguese, and Mandarin Chinese (Emelin and Sennrich, 2021; Amsili and Seminck, 2017; Melo et al., 2019; Bernard and Han, 2020). The task has also been reformulated to evaluate implicit biases in LLMs, with resulting benchmarks such as WinoGender and WinoBias (Rudinger et al., 2018; Zhao et al., 2018).

By 2019, large pre-trained transformer models were reported to achieve over 90% accuracy on WSC (Kocijan et al., 2023). Whereas the initial hypothesis was that systems would need common sense to solve the WSC, there is no proof that this

---

[1] A classic example is 'The trophy didn't fit in the brown suitcase because *it*'s too [small/big].'

is the case. Indeed, LLMs can rely on superficial pattern recognition and data memorization to solve the task, leading to the conclusion that these results are not indicative of common sense acquisition (Sakaguchi et al., 2021). Furthermore, questions are raised about the quality and implicit biases of WSC, such as lax evaluation, annotation artifacts, and knowledge leakage from training data (Kocijan et al., 2023; Elazar et al., 2021). Trichelair et al. (2018a) also show that the ‘google-proof’ condition, that stipulates that it should not be solvable via statistics learned from large corpora associating one option to other components in the sentence, is not true for all instances in WSC. In an effort to address these limitations, adversarial variants of the WSC are presented. For instance, Han et al. (2024) adapt the options so that they are more associated with the wrong answer, and Trichelair et al. (2018b) switch the position of the options in the texts where possible. Both report a decrease in model performance. Abdou et al. (2020) show that models are not robust against linguistic perturbations such as changes in tense, gender, or synonym substitution in WSC sentences. Additionally, the WinoGrande benchmark is introduced (Sakaguchi et al., 2021). This benchmark is of a much larger scale (44K instances compared to the 273 in WSC), and employs an algorithm to reduce biases that machines can exploit to solve the task.

2.3 Common sense knowledge categorization

To the best of our knowledge, research on WinoGrande discusses model results holistically (on the entire test or validation set), but we suggest connecting this to common sense knowledge categorization as an effective error analysis of the task. By measuring the performance per category, we can isolate reasoning deficiencies that are obscured by aggregated metrics. There has been much effort on defining semantic categories to structure knowledge for NLP. Schank (1972) describes four main categories in their conceptual dependency theory: objects, actions, location, and time. Jackendoff (1992) suggests common primitives such as entity, property, number, location, state, event, and activity. Other work only uses two high-level categories, such as social and physical (Sap et al., 2020). Yet others define semantic categories within one common sense category; for instance, Wang et al. (2021) include feelings and characteristics, interaction, and norms as sub-categories of social common sense. Additionally, different common sense categories are sometimes evaluated by specific independent benchmarks (e.g., spatial (Xu et al., 2017; Liu et al., 2022), temporal (Zhou et al., 2019; Aroca-Ouellette et al., 2021; Hosokawa et al., 2024; Qin et al., 2021), numerical (Lin et al., 2020), physical (Bisk et al., 2020; Storks et al., 2021), social (Sap et al., 2019), etc.). This can be problematic when comparing one model’s ability to reason over various common sense categories, since each benchmark can have a different answer format (i.e., multiple choice, binary choice, open-ended) and structure. Other benchmarks that are more general, do not provide common sense categorizations. Therefore, we annotate the WinoGrande benchmark (a general-purpose benchmark) with which common sense knowledge is relevant when making the decision (i.e., what knowledge is needed when making the bridging inference). In a similar effort, Zhang et al. (2020) proposed 6 common sense categories to evaluate performance on the WSC: property, object, eventuality, spatial, quantity, and others.

2.4 Benchmark memorization and contamination

Xu et al. (2024) define benchmark data contamination (BDC) as LLM exposure to benchmark data during training, leading to inflated evaluation results. They outline contamination severities ranging from exposure to meta information about the benchmark or the task, to the benchmark data itself with labels. One main detection technique is n-gram overlap counting, as used by GPT-3 (Brown et al., 2020) (13-gram) and GPT-4 (Achiam et al., 2023) (40-gram). However, it requires full pretraining data access and can miss rephrasing (Yang et al., 2023). Additionally, Wang et al. (2025) find that factual or lexical tasks are particularly susceptible to memorization, while Carlini et al. (2023) demonstrate that memorization increases with model size, data frequency, and sufficient context.

Since 2012, many WSC sentences have appeared in web text used to train LLMs (Elazar et al., 2021). RedPajama (Weber et al., 2024) contains 58.2% of WSC instances, while other datasets like The Pile (Gao et al., 2020a) contain around 30% (Elazar et al., 2024). Such contamination inflates accuracy scores: Emami et al. (2020) show significant accuracy drops when contamination is minimized.

In contrast, WinoGrande’s creators mitigated contamination by keeping the test labels private. Regarding the validation set, only 1.1% of this set appears online or in CommonCrawl between December 2020 and October 2023 (Li et al., 2024), and the authors of GPT-4 self-report approximately 0.9% contamination in a sample of 1,000 instances (Achiam et al., 2023). Elazar et al. (2024) demonstrate that large pretraining corpora for LLMs did likely not encounter the WinoGrande test set, but they do not examine contamination of the validation set in these pretraining corpora. Thus, the precise effect of the contamination of the WinoGrande validation set is unknown, but for other benchmark data, it was previously shown that the effect of even minimal contamination can be underestimated (Singh et al., 2024).

3 Methodology

3.1 Data

In this study, we apply models on the WinoGrande benchmark, which was originally presented in 2019 as an adversarial dataset to the Winograd Schema Challenge (WSC) (Sakaguchi et al., 2021). Contrary to WSC, in which the sentence includes a pronoun that must be disambiguated given two candidate antecedents, the WinoGrande benchmark evolved to a fill-in-the-blank token ('_') problem (see Figure 2). Additionally, every instance does not necessarily have a twin sentence. The original paper reports human accuracy of 94%, and model accuracy of 79.1%, which is considerably lower than on WSC (over 90%). The labels of the test set are not publicly available, which has led to research reporting on the validation set (see e.g., Li et al. (2021); Sun and Emami (2024); Elazar et al. (2021)). For that reason, we will also report on the validation set. This split consists of 1,267 instances, with a balanced label distribution. The WinoGrande benchmark is also frequently used to evaluate new LLMs[2]. Recent evaluations include Gemma 2 27B at 83.7% (Team et al., 2024), LlaMA 2 (zero-shot) models ranging from 69.2% (7B) to 80.2% (70B) (Touvron et al., 2023b), GPT-4 (few-shot) achieving 87.5% (Achiam et al., 2023), and Pythia 12B (five-shot) scoring 66.6% (Biderman et al., 2023).

3.2 Models

We focus on recent open-source Large Language Models. Since model size is a known factor in model performance, we select model families that have different sizes available. Specifically, we select Gemma 2 (2B, 9B, and 27B) (Team et al., 2024); LlaMA 2 (7B, 13B, and 70B) (Touvron et al., 2023b), and OPT (1.3B, 6.7B, 13B, and 66B) (Zhang et al., 2022) to evaluate the effect of paraphrasing WinoGrande, and for the evaluation per common sense category. Further, to evaluate benchmark memorization, we include two other models because their pre-training data is publicly available, contrary to the previously mentioned models: Pythia (1B, 1.8B, 6.9B, and 12B) (Biderman et al., 2023) and LlaMA 1 (7B, 13B, 30B, and 65B) (Touvron et al., 2023a).

To evaluate model performance, we use partial evaluation, which calculates the summed log-likelihood for the tokens after each option in the text, selecting the one with the highest score (Trinh and Le, 2018). We choose this metric for three reasons:

1. It is the evaluation metric used in the Language Model Evaluation Harness (Gao et al., 2024), which is the base of the Huggingface Open-LLM Leaderboard[3];
2. Preliminary experiments show that it works better than prompting, and Trinh and Le (2018) show that it works better than full evaluation;
3. It is easily generalizable to different open-source models.

3.3 Paraphrased corpus

To test the robustness of model performance on WinoGrande, we create WinoWhat: a parallel corpus in which we paraphrase the sentences. We follow the fill-in-the-blank convention of WinoGrande because of the naturalness of generation in autoregressive models and known LLM biases for multiple-choice answering (such as in WSC) (Balepur et al., 2024; Cho et al., 2025). Our corpus solves the main limitation of the partial evaluation metric: it relies on the plausibility of the subsequent sequence, rather than directly measuring a model's intrinsic token preference. This can conflate the model's understanding of the antecedent with grammatical or natural continuations. In contrast, in our paraphrased corpus, we position the target token at the end of the sequence, ensuring

---

[2] It is unclear whether they report on the validation or test set. We assume these models use prompting techniques instead of partial evaluation (infra), but the reports are unclear on that aspect.

[3] WinoGrande was included in the V1 of the leaderboard: `https://huggingface.co/docs/leaderboards/en/open_llm_leaderboard/archive`

that the decision is based solely on the provided context. This allows for a more transparent evaluation of the model's ability to capture coreference and fill-in-the-blank cues. Contrary to the original partial evaluation that measures the summed log-likelihood on the tokens following the '_' token, our method calculates it on the tokens of the options. An example is given in Figure 2.

We prompt 5 SOTA LLMs (i.e., GPT-4o (Hurst et al., 2024), OpenAI o1-preview (Jaech et al., 2024), Gemini 2.0 Flash Thinking Experimental (DeepMind, 2024), Deepseek R1 (Guo et al., 2025), and LlaMA 3.2 90B Vision (Meta, 2024)) to generate a paraphrased sentence given an input sentence, in which the '_' token is at the end of the sentence. The generated options were manually checked, and the best option was selected for each sentence. However, in many cases ($n = 433$), manual adjustments were still needed. The prompt for this task, and the distribution of which model's output is used, can be found in Appendix A. In this stage, we also evaluate the validity of the sentences in the WinoGrande validation set. We notice that not all instances meet the requirements of WSC (e.g., not 'google-proof', grammatical errors, etc.), which we remove in our paraphrased dataset. In total, we find 82 such cases.[4]

Further, three of the authors annotate a sample of 100 paraphrased instances based on the following criteria: (1) Is the new sentence grammatical?; (2) Is the fill-in-the-blank token at the end of the sentence?

85% of the texts are rated by all annotators as acceptable, 97% by at least two annotators. Given that the annotations are highly skewed (the majority of the ratings is 'acceptable'), we calculated Gwet's AC_1 for the inter-annotator agreement: 0.88 indicates a high agreement (Gwet, 2001).

3.4 Common sense knowledge categorization

We categorized the coreference resolution instances according to the common sense knowledge type that is necessary to make the bridging inference. This categorization can function as data for an error analysis to detect what knowledge types are easier or harder for LLMs to solve. Similarly to Zhang et al. (2020), we select categories that have a broad coverage and are clearly distinguished from each other. We examine which categories are identified in existing benchmarks that evaluate common sense reasoning in NLP[5], which leads to five categories: physical, social, numerical, temporal, and spatial[6]. We use LLMs to categorize the validation set. To identify the relevant common sense type, we prompt GPT-4o-mini to generate reasoning steps to solve the task. We then provide the input text and the generated reasoning steps to GPT-4o, which assigns one or more common sense categories to each instance. The prompts for these tasks are available in Appendix B. Annotation reliability is assessed by one author manually labeling 100 instances and comparing them with GPT-4o's labels, yielding a Kappa score of 0.64, which is a substantial agreement[7] (Cohen, 1960). Across all samples and labels, the annotator and GPT-4o agree in 83% of the cases. When applying our method on the entire validation set, we note a class imbalance; the physical and social categories are considerably larger than the other three, see Figure 3.

4 Results

4.1 Paraphrased corpus

We report on the models' performance on WinoWhat. This allows us to compare the performance on the original texts to the paraphrased texts. If models truly generalize on the Winograd schemas, the performance should remain consistent; after all, the same information is conveyed, in the same task setup, only paraphrased. Additionally, we report on the performance per common sense category presented in Section 2.3. In Table 1 and Table 2, subcolumn 'orig' refers to the original texts in WinoGrande, 'transf' to the paraphrased texts.

Considering the result on the WinoGrande validation set, we see that larger models generally perform better than their smaller variants, with LLaMA 2 70B performing the best. The error analysis comparing the performance of the same model across common sense categories shows that there is no one category that is impossible to be learned by a model, but there are fluctuations. Interestingly, we see that the category with the best results varies

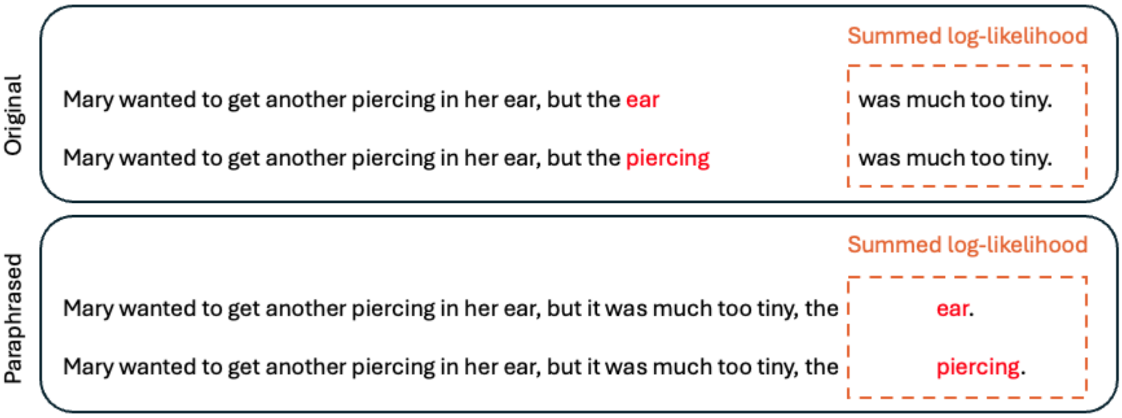
---

[4] There are an additional 22 instances for which one annotator was not convinced of the quality. These instances were left out in the experiments, but for completeness are added in the released dataset.

[5] e.g., see `https://cs.nyu.edu/~davise/Benchmarks/Text.html`

[6] Originally, we included causal as label, but removed this category: all instances in WinoGrande had this label, which was also noted by Zhang et al. (2020).

[7] The kappa scores per category: physical 0.63; social 0.68; numerical 0.58; temporal 0.72; spatial 0.59.



Mary wanted to get another piercing in her ear, but the ear was much too tiny.

Mary wanted to get another piercing in her ear, but the piercing was much too tiny.

Mary wanted to get another piercing in her ear, but it was much too tiny, the ear.

Mary wanted to get another piercing in her ear, but it was much too tiny, the piercing.

Figure 2: An illustration of the paraphrasing and evaluation method. The option that is filled in the ‘_’-token is in red. In the original example, the summed log-likelihood is calculated on the tokens after the option. In the paraphrased example, the option is at the end of the sentence, and the summed log-likelihood is calculated on the tokens inside the option.

	LlaMA 2						**Gemma 2**					
	7B		13B		70B		2B		9B		27B	
	orig	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**
TOTAL	0.69	0.58	0.72	0.62	0.78	0.70	0.68	0.59	0.74	0.68	0.66	0.56
Physical	0.71	0.61	0.73	0.63	0.77	0.73	0.71	0.60	0.74	0.68	0.74	0.59
Social	0.68	0.56	0.72	0.61	0.79	0.68	0.68	0.57	0.73	0.67	0.60	0.54
Numerical	0.69	0.53	0.70	0.61	0.79	0.69	0.63	0.62	0.75	0.62	0.69	0.51
Spatial	0.71	0.61	0.76	0.65	0.75	0.70	0.70	0.61	0.78	0.69	0.78	0.62
Temporal	0.76	0.67	0.70	0.69	0.79	0.74	0.65	0.62	0.74	0.71	0.67	0.54

Table 1: LlaMA 2 and Gemma 2 results on WinoGrande validation. The ‘orig’ columns report the results on the original instances, the ‘transf’ columns on the paraphrased instances.

	OPT 1.3B		OPT 6.7B		OPT 13B		OPT 66B	
	orig	**transf**	**orig**	**transf**	**orig**	**transf**	**orig**	**transf**
TOTAL	0.60	0.53	0.66	0.54	0.65	0.56	0.69	0.58
Physical	0.62	0.57	0.72	0.57	0.67	0.60	0.73	0.61
Social	0.59	0.50	0.63	0.50	0.65	0.52	0.66	0.55
Numerical	0.57	0.49	0.62	0.58	0.63	0.54	0.68	0.57
Spatial	0.56	0.61	0.65	0.61	0.63	0.61	0.67	0.61
Temporal	0.50	0.55	0.57	0.58	0.61	0.53	0.66	0.57

Table 2: OPT results on WinoGrande validation. The ‘orig’ columns report the results on the original instances, the ‘transf’ columns on the paraphrased instances.

across model families: for LlaMA 2, there is no category that is consistently easier, while for Gemma 2 spatial is best, and for OPT physical. Temporal is consistently the worst category for OPT.

However, when comparing the original to the paraphrased task, we conclude that all models perform worse on the paraphrased corpus, and there is no common sense category that is robust against this transformation.

Our results challenge the assumption that LLMs apply reasoning when solving the WinoGrande task, suggesting they instead rely on dataset artifacts and/or memorization. While Sakaguchi et al. (2021) implemented an algorithm to automatically reduce machine-exploitable bias in their corpus, our results demonstrate that this might not be effective anymore in the LLM era.

We publicly release WinoWhat, consisting of the original WinoGrande validation set with the paraphrased counterparts and common sense catego-

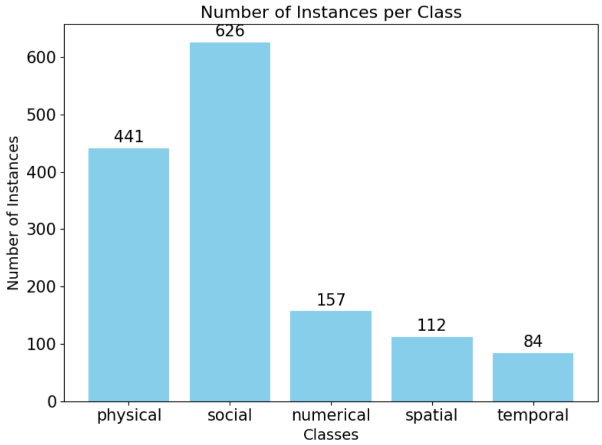


Figure 3: Data distribution across common sense categories on the WinoGrande validation set.

rizations[8].

4.2 Memorization

Model	WG val		WG test		WSC	
	orig	transf	orig	transf	orig	transf
LlaMA 2 7B	0.69	0.58	0.74	0.54	0.86	0.54
LlaMA 2 13B	0.72	0.62	0.73	0.65	0.83	0.63
LlaMA 2 70B	0.78	0.70	0.79	0.70	0.88	0.66
Gemma 2 2B	0.68	0.59	0.73	0.61	0.83	0.64
Gemma 2 9B	0.74	0.68	0.73	0.64	0.86	0.58
Gemma 2 27B	0.66	0.56	0.58	0.57	0.76	0.51
OPT 1.3B	0.60	0.53	0.58	0.50	0.72	0.54
OPT 6.7B	0.66	0.54	0.52	0.56	0.82	0.56
OPT 13B	0.65	0.56	0.68	0.56	0.81	0.56
OPT 66B	0.69	0.58	0.71	0.52	0.82	0.58

Table 3: Accuracy on WinoGrande (WG) validation, WG test, and WSC for LLaMA 2, Gemma 2, and OPT.

Model	WG val		WG test		WSC	
	orig	transf	orig	transf	orig	transf
LlaMA 1 7B	0.70	0.58	0.74	0.59	0.85	0.61
LlaMA 1 13B	0.72	0.60	0.75	0.64	0.88	0.66
LlaMA 1 30B	0.76	0.64	0.74	0.62	0.92	0.62
LlaMA 1 65B	0.77	0.67	0.79	0.69	0.91	0.68
Pythia 1B	0.54	0.53	0.57	0.54	0.71	0.50
Pythia 2.8B	0.60	0.52	0.59	0.53	0.76	0.55
Pythia 6.9B	0.61	0.52	0.58	0.56	0.77	0.52
Pythia 12B	0.63	0.52	0.61	0.60	0.79	0.49

Table 4: Accuracy on WinoGrande (WG) validation, WG test, and WSC for LLaMA 1 and Pythia.

Given the surprising drop in performance comparing WinoGrande to WinoWhat, we investigate further what could cause this. While Elazar et al. (2024) show that the test set of WinoGrande has probably not been seen by LLMs, this is not tested for the validation set. This is problematic, because research often reports on this split because of the absence of the test labels. Therefore, it is crucial to verify how many instances of the WinoGrande validation set have been included in datasets used to pre-train LLMs. Specifically, we count how many instances appear entirely in the pre-training corpora.

Since the pre-training data for Gemma 2, LlaMA 2, and OPT models remains either undisclosed or inaccessible, we examine two LLMs with publicly available pre-training data: LlaMA 1 and Pythia, whose results are presented in Table 4. These models were trained on RedPajama v1 (Computer, 2023) and The Pile's training set (Gao et al., 2020b) respectively.[9]

While we found that The Pile contains no contaminated instances, an interesting pattern emerges: as model size of Pythia increases, the performance gap between WinoGrande and WinoWhat widens, with WinoWhat accuracy remaining stable while WinoGrande scores improve (see column 'WG val' in Table 4).

An analysis of RedPajama v1 reveals 22 contaminated instances (1.7% of the dataset), each appearing once and sourced from academic papers. To investigate potential memorization effects, we conduct a one-sided Mann-Whitney U test between performance on contaminated and non-contaminated instances across LlaMA 1 models (7B, 13B, 30B, and 65B). The results (see Table 6 in Appendix C), with p-values ranging from 0.054 to 0.267, show no significant evidence that LLaMA 1 models give preferential treatment to previously seen WinoGrande instances. However, similarly to Pythia, LlaMA 1 displays a consistent accuracy gap between WinoGrande and WinoWhat. Since this pattern is observed in all other models as well (Table 1 and Table 2), it suggests that factors beyond simple memorization may be driving these performance differences.

To verify the role of contamination in later and more modern models with unknown pre-training data, we create two test-suites. Specifically, we take a sample ($n = 100$) from the WSC dataset (of which we can assume that a substantial part has been memorized by LLMs (Elazar et al., 2024)), and paraphrase those; and we take a sample ($n =$

---

[8] The full dataset is available on Zenodo (Gevers and De Marez, 2025).

[9] Details about our method to check memorization can be found in Appendix C.

100) from the test set of WinoGrande (of which we can assume that it has not been memorized by LLMs due to its private labels), which we label manually and paraphrase as well.[10] We hypothesize that LLMs perform well on datasets that are polluted, but less so on unseen datasets. Therefore, we expect models to perform well on WSC, but below par on WSC paraphrased and WinoGrande test (both original and paraphrased). We summarize the results in Table 3. As expected, all models perform best on the original WSC benchmark. Paraphrasing almost always causes a drop in performance, regardless of the original source. The difference is biggest for the WSC benchmark, which is in line with our hypothesis given the pollution by this benchmark in LLMs' training data. We still see a drop in performance for the WinoGrande test set, which is not included in the LLM training data, when comparing the original sentences to the paraphrased ones. Together with our findings on Pythia and LLaMA 1, this indicates that there are other factors causing models to struggle with the paraphrased benchmark. We hypothesize that our evaluation metric better captures the model's performance on coreference resolution compared to the original partial evaluation (see Figure 2), which could explain the drop in performance. Additionally, for larger and recent models, even though benchmark instances might not appear directly in the pre-training data, this does not exclude the possibility that it has been used during RLHF or instruction tuning, thereby compromising the validity of their performance on WinoGrande.

5 Conclusion

In this study, we take a closer look at how Winograd schema challenges can be used to evaluate common sense reasoning in LLMs. For this purpose, we focus on WinoGrande, a large adversarial benchmark created in 2019, frequently used to evaluate common sense in new LLMs. We select different generative model families, comparing models of the same family of different sizes. Specifically, we focus on Gemma 2, LlaMA 2, and OPT. To evaluate the models, we employ the partial evaluation metric. To address the limitations of the partial evaluation metric as outlined in Section 3.3, we create a parallel corpus to the WinoGrande validation set in which we paraphrase each text so the fill-in-the-blank token is at the end of the sentence (**RQ1**). In addition, we propose a new method to inspect performance on various common sense knowledge categories within the same task (**RQ2**). We select five categories: physical, social, numerical, spatial, and temporal. This approach can offer an in-depth error analysis, that sheds light on what types of knowledge are more challenging for LLMs. We publicly release WinoWhat, the parallel corpus to the WinoGrande validation set including the paraphrased sentences and the common sense categorization. Our results show that while models perform well on the original WinoGrande validation set, they all perform worse on the paraphrased corpus, and all common sense categories are affected negatively. This questions the assumption that models apply reasoning, leaving the possibility for dataset artifacts or benchmark memorization.

To verify how much data memorization has an effect on the models' performance on the WinoGrande validation set (**RQ3**), we test whether instances that occur in pre-training data score significantly higher than instances that don't. We observe that the memorization of the validation set is minimal. Interestingly, we see that most contaminated instances come from academic publications citing examples from the benchmark. This again calls attention to the scraping methods to create large-scale pre-training data. Because the pre-training data of later models is unknown, we create two small ($n = 100$) test-suites: one of which has been shown to be included in LLM training sets (i.e., the WSC benchmark) and one that is not seen by LLMs (i.e., the WinoGrande test set). We find that all models perform best on the WSC dataset, and paraphrasing causes a drop in performance. Since this is also the case for the WinoGrande test set, we conclude that there are other factors beside memorization that cause models to fail on the paraphrased task. Similarly to conclusions about the original Winograd Schema Challenge, this implies that we are again overestimating LLMs reasoning capabilities when using WinoGrande. Our new paraphrased corpus can be used to verify model generalization on the WinoGrande validation set.

In further research, we plan to inspect the information that is used by models to solve the task per common sense category using mechanistic interpretability: do models use similar information for each category? Do they rely on spurious

---

[10] To respect the private nature of the WinoGrande test set, we do not release our annotations of this subset.

correlations, and if so, which ones? Mechanistic interpretability could help us identify a causal connection between the direct and the indirect object, giving insights on why models fail. Since data memorization does not seem to cause the drop in performance comparing the original to the paraphrased instances, we suggest to identify dataset artifacts that could be at the root of this. For instance, as previously done on WSC, do linguistic perturbations affect model performance?

Acknowledgments

This research was made possible with a grant from the Fonds Wetenschappelijk Onderzoek (FWO) project 42/FA030100/9770, and funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

Limitations

While, to the best of our knowledge, this is the first time the WinoGrande validation set has been annotated for common sense knowledge categories, this approach has possible shortcomings. First, the agreement between a human annotator and the labeling by GPT-4o shows a substantial agreement, but there will be cases with incorrect labeling. Therefore, we talk about aggregated results across categories in this study, since we're interested in trends, but for even more fine-grained interpretations this categorization should possibly be corrected manually.

Further, as is unfortunately still a trend in NLP-research, this dataset is in English, excluding lower-resource languages. Further research could translate our benchmark to other languages.

During the process of paraphrasing the original instances, we applied a strict quality check, which excluded 82 instances from the original dataset. While we believe this improves the quality of the resulted paraphrased dataset, this means we cannot make a perfectly aligned comparison to the original dataset.

Since we wanted to mitigate shortcomings of the partial evaluation metric, we paraphrased WinoGrande so the fill-in-the-blank token appears at the end of the sentence. However, the constraint of putting this token at the end of the sentence caused a higher number of cleft-constructions in the corpus. A high inter-annotator agreement shows that the created paraphrases are grammatically correct and qualitative, but in some cases the paraphrased output is less natural than the original. However, even though there might be an 'unnaturalness' about some of the paraphrased instances, this does not change the task (i.e., finding the correct antecedent), and a robust model should be able to overcome these superficial variations.

We argue that this setup is more natural for decoder-only models, and allows the partial evaluation metric to better capture model performance on coreference resolution tasks rather than measuring natural continuations of the sentence. However, by adapting the evaluation method so it calculates the summed log-likelihoods on the tokens in the option rather than on the tokens after the option, this obscures whether the difference in performance is a result of the paraphrasing, or of the evaluation method. To verify this, we aim to construct a third level, in which we paraphrase without the constraint of putting the '_'-token at the end of the sentence, allowing us to use the original partial evaluation method. This would indicate whether the drop in performance is caused by the paraphrasing itself, or by the evaluation metric. This would also alleviate the problem that some paraphrased sentences in WinoWhat are slightly less natural-sounding than the original ones.

Finally, our method of finding data contamination in pre-training data was on the data level only, not taking into account the semantic or information level (Xu et al., 2024). Methods such as ours relying on string matching methods might miss certain instances, such as rephrasings (Xu et al., 2024). Furthermore, such methods are only possible when access to pre-training corpora is public (Yang et al., 2023).

References

Mostafa Abdou, Vinit Ravishankar, Maria Barrett, Yonatan Belinkov, Desmond Elliott, and Anders Søgaard. 2020. The sensitivity of language models and humans to Winograd schema perturbations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7590–7604, Online. Association for Computational Linguistics.

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

Pascal Amsili and Olga Seminck. 2017. A Google-proof collection of French Winograd schemas. In *Proceedings of the 2nd Workshop on Coreference Resolution Beyond OntoNotes (CORBON 2017)*, pages 24–29, Valencia, Spain. Association for Computational Linguistics.

Stéphane Aroca-Ouellette, Cory Paik, Alessandro Roncone, and Katharina Kann. 2021. Prost: Physical reasoning of objects through space and time. *arXiv preprint arXiv:2106.03634*.

Nishant Balepur, Abhilasha Ravichander, and Rachel Rudinger. 2024. Artifacts or abduction: How do LLMs answer multiple-choice questions without the question? *Preprint*, arXiv:2402.12483.

Timothée Bernard and Ting Han. 2020. Mandarinograd: A Chinese collection of Winograd schemas. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 21–26, Marseille, France. European Language Resources Association.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

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, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jacob Browning and Yann LeCun. 2023. Language, common sense, and the winograd schema challenge. *Artificial Intelligence*, 325:104031.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations*.

Gyeongje Cho, Yeonkyoung So, and Jaejin Lee. 2025. ANPMI: Assessing the true comprehension capabilities of LLMs for multiple choice questions. *Preprint*, arXiv:2502.18798.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Google DeepMind. 2024. Gemini 2.0 flash thinking. 2024.

Yanai Elazar, Akshita Bhagia, Ian Helgi Magnusson, Abhilasha Ravichander, Dustin Schwenk, Alane Suhr, Evan Pete Walsh, Dirk Groeneveld, Luca Soldaini, Sameer Singh, Hannaneh Hajishirzi, Noah A. Smith, and Jesse Dodge. 2024. What's in my big data? In *The Twelfth International Conference on Learning Representations*.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the winograd schema. *arXiv preprint arXiv:2104.08161*.

Ali Emami, Kaheer Suleman, Adam Trischler, and Jackie Chi Kit Cheung. 2020. An analysis of dataset overlap on Winograd-style tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5855–5865, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Denis Emelin and Rico Sennrich. 2021. Wino-x: Multilingual winograd schemas for commonsense reasoning and coreference resolution. In *2021 Conference on Empirical Methods in Natural Language Processing*, pages 8517–8532. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020a. The pile: An 800gb dataset of diverse text for language modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020b. The Pile: An 800GB dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Ine Gevers and Victor De Marez. 2025. Winowhat: A parallel corpus of paraphrased winogrande sentences with common sense categorization.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Kilem Gwet. 2001. Handbook of inter-rater reliability: How to estimate the level of agreement between two or multiple raters. *Gaithersburg, MD: STATAXIS Publishing Company*.

Kaiqiao Han, Tianqing Fang, Zhaowei Wang, Yangqiu Song, and Mark Steedman. 2024. Concept-reversed winograd schema challenge: Evaluating and improving robust reasoning in large language models via abstraction. *arXiv preprint arXiv:2410.12040*.

Taishi Hosokawa, Adam Jatowt, and Kazunari Sugiyama. 2024. Temporal validity reassessment: commonsense reasoning about information obsoleteness. *Discover Computing*, 27(1):4.

Yufang Hou, Katja Markert, and Michael Strube. 2018. Unrestricted bridging resolution. *Computational Linguistics*, 44(2):237–284.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Ray S Jackendoff. 1992. *Semantic structures*, volume 18. MIT press.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Hideo Kobayashi and Vincent Ng. 2020. Bridging resolution: A survey of the state of the art. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3708–3721.

Vid Kocijan, Ernest Davis, Thomas Lukasiewicz, Gary Marcus, and Leora Morgenstern. 2023. The defeat of the winograd schema challenge. *Artificial Intelligence*, 325:103971.

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. *KR*, 2012:13th.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2021. A systematic investigation of commonsense knowledge in large language models. *arXiv preprint arXiv:2111.00607*.

Yucheng Li, Yunhao Guo, Frank Guerin, and Chenghua Lin. 2024. An open-source data contamination report for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 528–541, Miami, Florida, USA. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pre-trained language models. *arXiv preprint arXiv:2005.00683*.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. *arXiv preprint arXiv:2401.17377*.

Ruicheng Liu, Rui Mao, Anh Tuan Luu, and Erik Cambria. 2023. A brief survey on recent advances in coreference resolution. *Artificial Intelligence Review*, 56(12):14439–14481.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. 2022. Things not written in text: Exploring spatial commonsense from visual signals. *arXiv preprint arXiv:2203.08075*.

John McCarthy. 1959. Programs with common sense.

Gabriela Melo, Vinicius Imaizumi, and Fábio Cozman. 2019. Winograd schemas in portuguese. In *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*, pages 787–798, Porto Alegre, RS, Brasil. SBC.

Meta. 2024. Meta llama3.2. `https://www.llama.com/`.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. Timedial: Temporal commonsense reasoning in dialog. *arXiv preprint arXiv:2106.04571*.

Altaf Rahman and Vincent Ng. 2011. Coreference resolution with world knowledge. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 814–824.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th annual meeting of the association for* *computational linguistics: Tutorial abstracts*, pages 27–33.

Roger C. Schank. 1972. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552–631.

Aaditya K. Singh, Muhammed Yusuf Kocyigit, Andrew Poulton, David Esiobu, Maria Lomeli, Gergely Szilvasy, and Dieuwke Hupkes. 2024. Evaluation data contamination in llms: how do we measure it and (when) does it matter? *Preprint*, arXiv:2411.03923.

Shane Storks, Qiaozi Gao, Yichi Zhang, and Joyce Chai. 2021. Tiered reasoning for intuitive physics: Toward verifiable commonsense language understanding. *arXiv preprint arXiv:2109.04947*.

Jing Han Sun and Ali Emami. 2024. EvoGrad: A dynamic take on the Winograd schema challenge with human adversaries. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 6701–6716, Torino, Italia. ELRA and ICCL.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Paul Trichelair, Ali Emami, Jackie Chi Kit Cheung, Adam Trischler, Kaheer Suleman, and Fernando Diaz. 2018a. On the evaluation of common-sense reasoning in natural language understanding. *arXiv preprint arXiv:1811.01778*, 20180.

Paul Trichelair, Ali Emami, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2018b. How reasonable are common-sense reasoning tasks: A case-study on the winograd schema challenge and swag. *arXiv preprint arXiv:1811.01778*.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Gengyu Wang, Xiaochen Hou, Diyi Yang, Kathleen McKeown, and Jing Huang. 2021. Semantic categorization of social knowledge for commonsense question answering. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 79–85, Virtual. Association for Computational Linguistics.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2025. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*.

Maurice Weber, Dan Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy S Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 116462–116492. Curran Associates, Inc.

Terry Winograd. 1972. Understanding natural language. *Cognitive psychology*, 3(1):1–191.

Cheng Xu, Shuhao Guan, Derek Greene, and M. Tahar Kechadi. 2024. Benchmark data contamination of large language models: A survey. *CoRR*, abs/2406.04244.

Frank F Xu, Bill Yuchen Lin, and Kenny Q Zhu. 2017. Automatic extraction of commonsense locatednear knowledge. *arXiv preprint arXiv:1711.04204*.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *CoRR*, abs/2311.04850.

Pardis Sadat Zahraei and Ali Emami. 2024. WSC+: Enhancing the Winograd schema challenge using tree-of-experts. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1671, St. Julian's, Malta. Association for Computational Linguistics.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020. WinoWhy: A deep diagnosis of essential commonsense knowledge for answering Winograd schema challenge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5736–5745, Online. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference* *of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. " going on a vacation" takes longer than" going for a walk": A study of temporal commonsense understanding. *arXiv preprint arXiv:1909.03065*.

Planning for Success: Exploring LLM Long-term Planning Capabilities in Table Understanding

Thi-Nhung Nguyen[1], **Hoang Ngo**[2], **Dinh Phung**[1], **Thuy-Trang Vu**[1], **Dat Quoc Nguyen**[2]
[1]Monash University, [2]Qualcomm AI Research[*]
`{nhung.thinguyen,dinh.phung,trang.vu1}@monash.edu`
`{hoangngo,datnq}@qti.qualcomm.com`

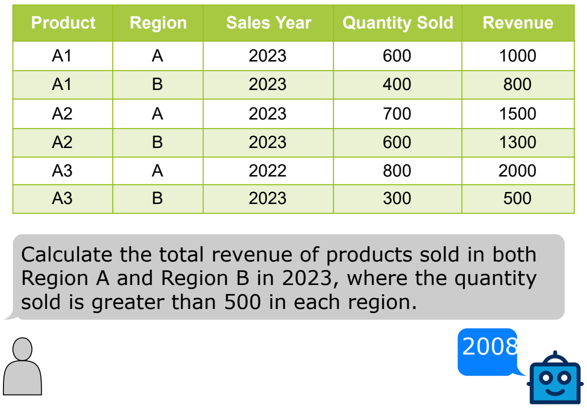
Abstract

Table understanding is key to addressing challenging downstream tasks such as table-based question answering and fact verification. Recent works have focused on leveraging Chain-of-Thought and question decomposition to solve complex questions requiring multiple operations on tables. However, these methods often suffer from a lack of explicit long-term planning and weak inter-step connections, leading to miss constraints within questions. In this paper, we propose leveraging the long-term planning capabilities of large language models (LLMs) to enhance table understanding. Our approach enables the execution of a long-term plan, where the steps are tightly interconnected and serve the ultimate goal, an aspect that methods based on Chain-of-Thought and question decomposition lack. In addition, our method effectively minimizes the inclusion of unnecessary details in the process of solving the next short-term goals, a limitation of methods based on Chain-of-Thought. Extensive experiments demonstrate that our method outperforms strong baselines and achieves state-of-the-art performance on WikiTableQuestions and TabFact datasets.

1 Introduction

Table understanding is key to addressing challenging downstream tasks involving tables, one of the most prevalent forms of semi-structured data in real-world scenarios, such as table question answering (Wang et al., 2023a; Lin et al., 2023) and fact verification (Chen et al., 2020). The primary goal is to accurately extract relevant information from tables to provide precise answers to user questions. To better understand the problem consider the example in Table 1.

Early works focus on fine-tuning BERT to encode tables (Herzig et al., 2020; Chen et al., 2020). The key idea is to leverage specialized embedding layers or attention mechanisms to encode table cells or segments effectively, enabling models to understand the structure of tables. Another direction revolves around the synthesis of SQL query-response pairs to pre-train an encoder-decoder model as a neural SQL executor (Eisenschlos et al., 2020; Liu et al., 2022b; Jiang et al., 2022). With the advent of large language models (LLMs), recent works have explored instruction fine-tuning of LLMs with tabular data to create generalist models capable of handling a variety of table-based tasks (Zhang et al., 2024), showing improved performance over flagship closed-source LLMs such as `GPT-3.5-turbo` and `GPT-4` (OpenAI et al., 2024).



Product	Region	Sales Year	Quantity Sold	Revenue
A1	A	2023	600	1000
A1	B	2023	400	800
A2	A	2023	700	1500
A2	B	2023	600	1300
A3	A	2022	800	2000
A3	B	2023	300	500

Calculate the total revenue of products sold in both Region A and Region B in 2023, where the quantity sold is greater than 500 in each region.

Figure 1: A question-answering example over a table.

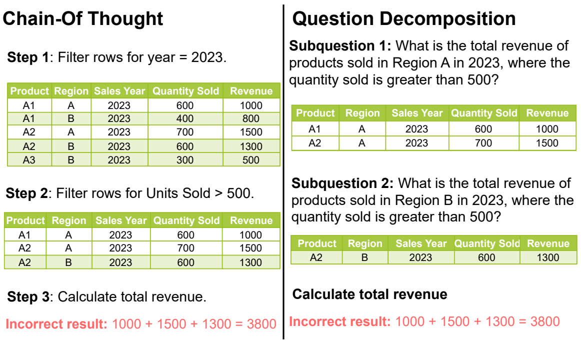
Leveraging the strong in-context learning performance of LLMs, recent works have increasingly focused on addressing table understanding through prompting. One common approach is to convert the question into executable languages, allowing the use of tools such as SQL or Python to access the information inside the table (Lin et al., 2023; Gemmell and Dalton, 2023; Wang et al., 2024; Nahid and Rafiei, 2024; Liu et al., 2024; Kong et al., 2024). However, due to the constraints of the single-pass generation process, these methods often struggle with complex questions requiring multiple steps of table operations. To address this

---

[*]Qualcomm Vietnam Company Limited.

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 81–92
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

challenge, some state-of-the-art methods employ Chain-of-Thought (CoT) reasoning, which enables multi-step reasoning (Yao et al., 2023; Chen et al., 2023; Wei et al., 2022; Wang et al., 2024). Others rely on question decomposition, breaking down the question into sub-questions, solving them individually, and finally synthesizing a final answer (Kong et al., 2024; Patnaik et al., 2024; Ye et al., 2023). However, both CoT-based methods and question decomposition-based methods suffer from a lack of explicit long-term planning and weak inter-step connections. This results in missing constraints within the question, leading to incorrect final answers. An illustration of this issue is shown in Figure 2, where step 3 is not conditionally linked to the previous steps. In addition, in the case of CoT-based methods, the entire current chain is often utilized to generate the output for the subsequent step. This approach can result in LLMs forgetting critical details or generating hallucinations, as they process a substantial amount of information, including extraneous details, which may introduce unnecessary complexity and lead to errors (Jiang et al., 2022; Chen, 2023).



Chain-Of Thought

Step 1: Filter rows for year = 2023.

Product	Region	Sales Year	Quantity Sold	Revenue
A1	A	2023	600	1000
A1	B	2023	400	800
A2	A	2023	700	1500
A2	B	2023	600	1300
A3	B	2023	300	500

Step 2: Filter rows for Units Sold > 500.

Product	Region	Sales Year	Quantity Sold	Revenue
A1	A	2023	600	1000
A2	A	2023	700	1500
A2	B	2023	600	1300

Step 3: Calculate total revenue.

Incorrect result: 1000 + 1500 + 1300 = 3800

Question Decomposition

Subquestion 1: What is the total revenue of products sold in Region A in 2023, where the quantity sold is greater than 500?

Product	Region	Sales Year	Quantity Sold	Revenue
A1	A	2023	600	1000
A2	A	2023	700	1500

Subquestion 2: What is the total revenue of products sold in Region B in 2023, where the quantity sold is greater than 500?

Product	Region	Sales Year	Quantity Sold	Revenue
A2	B	2023	600	1300

Calculate total revenue

Incorrect result: 1000 + 1500 + 1300 = 3800

Figure 2: An illustration showing how CoT-based methods and question decomposition-based methods miss the important inter-region condition in revenue calculation (corresponding to the table and question in Figure 1).

In this paper, we propose leveraging the long-term planning capabilities of LLMs to address these challenges. Unlike methods based on CoT and question decomposition, which lack explicit long-term planning, our method begins with the formulation of a long-term plan upon receiving a question. This plan outlines the necessary steps, called short-term goals, to progress systematically from the initial table to the final answer. The short-term goals can be either independent or interconnected, depending on the requirements of the question, ensuring that each serves the long-term goal. To handle each short-term goal effectively, we leverage a set of specialized experts, each dedicated to a specific task. These experts take responsibility for handling short-term goals relevant to their specialization, operating independently to resolve the goals within their localized scope. At this local level, each expert focuses solely on their assigned goal without being influenced by other parts of the long-term plan. The intermediate steps executed by the execution experts are single-pass. Once the short-term goal is completed, only the final results are updated within the long-term plan, minimizing the inclusion of unnecessary information in the process of solving the next short-term goals—a common issue in CoT-based methods.

Our contributions are summarized as follows: **(I)** We propose leveraging the long-term planning capabilities of LLMs to enhance table understanding. **(II)** Our approach enables the execution of a long-term plan where the steps are tightly interconnected, all serving the ultimate goal—an aspect that methods based on Chain-of-Thought and question decomposition lack. **(III)** Our approach effectively minimizes the inclusion of unnecessary details in the process of solving the next short-term goals—a limitation of methods based on Chain-of-Thought. **(IV)** Comprehensive experiments demonstrate that our approach achieves state-of-the-art performance, outperforming existing strong baselines on standard benchmarks WikiTableQuestions and TabFact.

2 Related Works

Fine-tuning pre-trained BERT models (Devlin et al., 2019) were one the dominant approach for Table Understanding (Herzig et al., 2020; Chen et al., 2020; Liu et al., 2022a; Deng et al., 2022; Wang et al., 2021; Iida et al., 2021). TaPas (Herzig et al., 2020) leverage the mask language modeling approach proposed in BERT to reconstruct certain cells in the table during training process. Wang et al. (2021) further enhance the performance by masking the entire columns in tables. A different approach is to train an encoder-decoder model to transform questions into SQL queries and then answer these questions by executing the respective generated SQL queries (Eisenschlos et al., 2020; Liu et al., 2022b; Jiang et al., 2022). Recently, large language models (LLMs) have demonstrated excellent performance on a variety of tasks. Recent works have been shifting their focus to fine-tuning open-source LLMs to create models capable of handling a variety of table-based tasks. However,

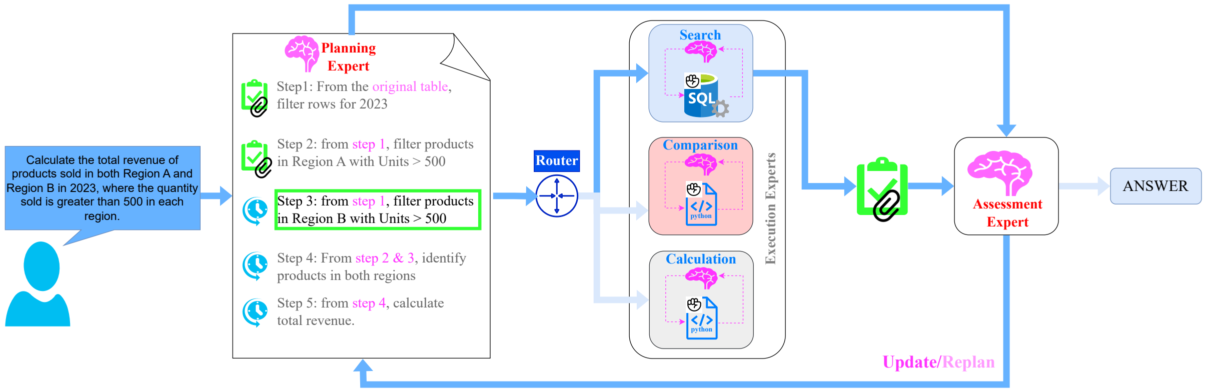

Calculate the total revenue of products sold in both Region A and Region B in 2023, where the quantity sold is greater than 500 in each region.
Planning Expert
Step1: From the original table, filter rows for 2023
Step 2: from step 1, filter products in Region A with Units > 500
Step 3: from step 1, filter products in Region B with Units > 500
Step 4: From step 2 & 3, identify products in both regions
Step 5: from step 4, calculate total revenue.
Router
Search
SQL
Comparison
python
Calculation
python
Execution Experts
Assessment Expert
ANSWER
Update/Replan

Figure 3: Overview of our proposed framework **PLANTA**.

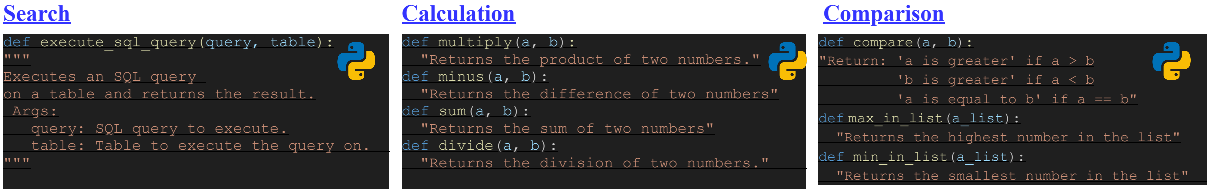
these methods require expensive labeled data and high training costs. This has led to the emergence of prompt-based approaches, which leverage the in-context learning capabilities of LLMs.

For prompt-based methods, some works propose concatenating task descriptions with the serialized table as a string and inputting them into an LLM to generate a text-based response (Marvin et al., 2023; Cheng et al., 2023; Sui et al., 2024). Other works enhance the performance further by adding few-shot and curated examples to the prompt (Cheng et al., 2023; Narayan et al., 2022; Chen, 2023). However, with reasoning only, LLMs often struggle to accurately retrieve all relevant data required within tables. Therefore, recent works increasingly incorporate external tools (e.g., Python and SQL) instead of relying solely on general text processing to effectively extract relevant data within tables (Chen et al., 2023; Gao et al., 2023; Rajkumar et al., 2022; Cheng et al., 2023; Ni et al., 2023). Despite this, due to the constraint of a single pass, this approach still struggles with complex questions where multiple operations need to be executed to produce an accurate answer. Recent state-of-the-art methods mitigate this limitation by employing chain-of-thought (CoT) reasoning or question decomposition (Chen et al., 2023; Zhao et al., 2024; Yang et al., 2024; Zhou et al., 2023; Khot et al., 2023). Some works (Ye et al., 2023; Cheng et al., 2023; Liu et al., 2024) further enhance the performance by self-consistency technique (Wang et al., 2023b), where a diverse set of reasoning paths is sampled from LLMs and the most consistent answer is selected to obtain the final answer. However, both CoT-based methods and question decomposition-based methods suffer from a lack of explicit long-term planning and weak inter-step connections. This results in constraints within the question being missed, leading to incorrect final answers. Furthermore, CoT-based methods often utilize the entire current chain to generate the output for the subsequent step. This approach can result in LLMs forgetting critical details or generating hallucinations, as they process a substantial amount of information, including extraneous details, which may introduce unnecessary complexity and lead to errors (Jiang et al., 2022; Chen, 2023).

3 Our Approach

We introduce a novel method, named **PLANTA**, which leverages the long-term **Plan**ning capabilities of Large Language Models to improve **Ta**ble Understanding. PLANTA is designed to tackle the challenge of generating accurate answers to table-based questions by extracting and reasoning over relevant information from the given tables.

Figure 3 illustrates the architectural overview of PLANTA. First, upon receiving a table and a question, a Planning expert comes up with a long-term plan outlining the necessary steps, called short-term goals, to transform the initial table into the desired answer to the user's question. Next, each short-term goal is routed to an appropriate Execution expert by a Router, which assigns short-term goals to experts based on their specialization via LLM prompting. These goals are then resolved locally, with only the final results passed to the following components of PLANTA, potentially updating the long-term plan. Meanwhile, intermediate steps executed by the Execution experts are processed in a single pass. After each step, the updated long-term plan is evaluated by an Assessment expert, who determines whether sufficient evidence has been gathered to answer the question or if modifications to the plan are necessary. If no adjustments are needed, the process continues. Below,



Search

```
def execute_sql_query(query, table):
"""
Executes an SQL query
on a table and returns the result.
 Args:
   query: SQL query to execute.
   table: Table to execute the query on.
"""
```

Calculation

```
def multiply(a, b):
  "Returns the product of two numbers."
def minus(a, b):
  "Returns the difference of two numbers"
def sum(a, b):
  "Returns the sum of two numbers"
def divide(a, b):
  "Returns the division of two numbers."
```

Comparison

```
def compare(a, b):
"Return: 'a is greater' if a > b
         'b is greater' if a < b
         'a is equal to b' if a == b"
def max_in_list(a_list):
  "Returns the highest number in the list"
def min_in_list(a_list):
  "Returns the smallest number in the list"
```

Figure 4: Predefined Python functions ("hands") assigned to the Search expert, Calculation expert, and Comparison expert in PLANTA, respectively.

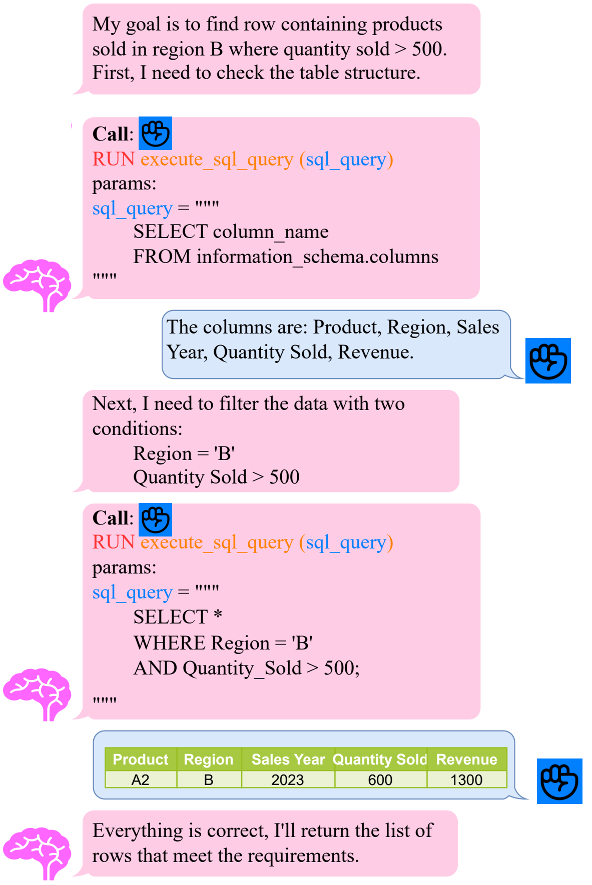
we provide a detailed description of the architecture and roles of the experts within PLANTA. We first outline the common architecture shared by all experts in Subsection 3.1, followed by an in-depth discussion of the differences in their architecture and their specific contributions in Subsection 3.2.

3.1 Common Architecture

In PLANTA, each expert consists of two main components: the "brain" and the "hands". Each *brain* is specialized in a specific task and can independently reason to complete an assigned task. It is powered by an LLM, whose knowledge scope is encoded through prompting. The *hands* are predefined tools, such as Python or SQL execution functions, tailored to the expert's specializations. These tools enable access to detailed data within tables and execute operations that LLMs may struggle with, such as calculations. They provide the brain with the necessary inputs for reasoning and determining the subsequent steps required to complete the task.

3.2 Task-Specific Architecture

Planning expert: Its role is to outline the necessary steps of short-term goals, structured as a task list, to transform the initial table into an accurate answer. Since this role focuses solely on planning without execution, the Planning expert's architecture comprises only the "brain". This brain is powered by an LLM specifically designed for the planning task, with a knowledge scope that includes the given table, the question, and the specializations of Execution experts (see our prompt for Planing in Appendix). For each step in the plan, dependencies on previous steps must be explicitly defined to enable the flexible reuse of variables from earlier steps. This approach minimizes the transfer of unnecessary information to subsequent steps while ensuring that all dependencies are correctly managed. For example, in Figure 3, step 3 depends only on the output of step 1. Therefore, step 3 can access only the output of step 1 that it depends on, without accessing the output of step 2.



My goal is to find row containing products sold in region B where quantity sold > 500. First, I need to check the table structure.

Call:
RUN execute_sql_query (sql_query)
params:
sql_query = """
    SELECT column_name
    FROM information_schema.columns
"""

The columns are: Product, Region, Sales Year, Quantity Sold, Revenue.

Next, I need to filter the data with two conditions:
    Region = 'B'
    Quantity Sold > 500

Call:
RUN execute_sql_query (sql_query)
params:
sql_query = """
    SELECT *
    WHERE Region = 'B'
    AND Quantity_Sold > 500;
"""

Product	Region	Sales Year	Quantity Sold	Revenue
A2	B	2023	600	1300

Everything is correct, I'll return the list of rows that meet the requirements.

Figure 5: An example of how the Search expert addresses the 3rd short-term goal from Figure 3. Once the short-term goal is received, the Search expert performs reasoning step by step. The process includes understanding the question, analyzing the table structure, identifying the data that meets the goal's conditions, and providing the answer. When the expert needs to access data in the table, it automatically calls the predefined SQL execution function (see Figure 4) to retrieve the necessary information. This is done by generating SQL queries as parameters for the function.

Execution experts: The task of the Execution experts is to resolve the short-term goals required by the Planning expert. These goals are assigned to appropriate Execution experts based on their specialization by the Router, which we perform using an LLM via prompting (see our prompt for the Router in Appendix). Motivated by operations on tables, our PLANTA system is designed with three Execution experts: (1) Search expert, (2) Comparison expert, and (3) Calculation expert. In terms of their knowledge scope, they can only access the data and short-term goals provided by the Planning expert and reason with the support of predefined functions, which we call "hands". The predefined functions include SQL query execution for the Search expert; comparative and superlative comparison for the Comparison expert; and basic calculations such as addition, subtraction, multiplication, and division for the Calculation expert. See Figure 4 for more details.

Unlike the initial question on tables, which must be addressed according to a pre-defined long-term plan, short-term goals are often simpler and more manageable. Therefore, we allow the experts to independently determine how to resolve assigned goals, such as utilizing the Chain-of-Thought or question decomposition approach, as long as the assigned goal is solved (see our prompts for three Execution experts in Appendix). We expect that this flexibility will enable the experts to reason and use their "hands" in ways that align with their execution capabilities. Figure 5 illustrates an example of how an Execution Expert addresses an assigned short-term goal.

Assessment expert: The task of the Assessment expert is to evaluate the quality of the plan after receiving the result of the current short-term goal from the Execution Expert. Similar to the Planning expert, it does not execute tasks but focuses solely on reasoning, using an LLM specialized in assessment tasks. Its knowledge scope includes access to the table, the question, and the long-term plan (see our prompt for the Assessment in Appendix). After every *k* short-term goals, the Assessment expert determines whether to generate an early answer if sufficient relevant information is available, or revise the plan if the results from the Execution experts fail to meet requirements or if the initial plan appears infeasible. Otherwise, the results from the Execution experts are automatically incorporated into the long-term plan. In essence, it takes a plan as input and outputs either a revised or an unchanged plan, or an answer.

Recommendation: Our preliminary experiments show that frequent assessments, such as after each short-term goal, can facilitate early answers, conserve resources, and quickly address errors as they arise. However, frequent evaluations may also lead to challenges, such as overemphasizing short-term results at the expense of long-term objectives, unnecessary plan revisions (e.g., repeated short-term result validations), inaccurate premature answers, and increased resource costs. To mitigate these issues, *k* should be tuned based on the data and the complexity of the question, balancing stability and efficiency.

Statistics	**WikiTQ**	**TabFact**
# Questions	4343	2024
# Number of Tables	421	298
# Min/Max Rows	6/518	5/49
# Min/Max Columns	5/20	3/21

Table 1: Statistics of the WikiTableQuestions (WikiTQ) and TabFact test sets.

4 Experiment Setup

Dataset and Metric: Following previous works (Wang et al., 2024), we conduct experiments on the benchmark datasets WikiTableQuestions—a question answering dataset over semi-structured tables (Pasupat and Liang, 2015) and TabFact—a dataset for table-based fact verification (Chen et al., 2020). Table 1 describes the statistics of their test sets. See a description of both datasets in the Appendix.

We employ the official denotation accuracy (Pasupat and Liang, 2015) for WikiTableQuestions and the binary classification accuracy for TabFact.

Baselines: We compare our method to recent strong table understanding methods, including **TEXT2SQL** (Rajkumar et al., 2022), **CHAIN-OF-THOUGHT** (Wei et al., 2022), **Dater** (Khot et al., 2023), **StructGPT** (Jiang et al., 2023), **BINDER** (Cheng et al., 2023), **TabSQLify** (Nahid and Rafiei, 2024), **CHAIN-OF-TABLE** (Wang et al., 2024) and **DP&PYAGENT** (Liu et al., 2024). CHAIN-OF-TABLE and DP&PYAGENT are the state-of-the-art methods on TabFact and WikiTableQuestions, respectively.

Implementation Details: We utilize *LangGraph* to construct our proposed model, PLANTA, which is conceptualized as a graph.[1] In this graph, the long-term plan represents the graph's state and each expert presents a node. Each expert is powered by an LLM with a distinct prompt, as detailed in Appendix . We mainly use "GPT-3.5-turbo" and "GPT-4o-mini" from OpenAI as the LLMs. The *temperature* for LLMs is set to 0. The maximum

---

[1] https://langchain-ai.github.io/langgraph/

number of iterations for a full turn of reasoning and execution of predefined functions per expert is set to 2. The maximum number of short-term goals is set to 12. The Assessment expert evaluates the long-term plan after completing $n-1$ steps of the plan where n is the number of short-term goals in the plan.

5 Evaluation

5.1 Main Results

Table 2 reports the accuracy of our PLANTA and strong baselines on WikiTableQuestions (WikiTQ) and TabFact test sets.

Recent state-of-the-art methods, including CHAIN-OF-TABLE and DP&PYAGENT, rely on chain-of-thought reasoning and self-consistency, demonstrating the effectiveness of these methods for table understanding. Both CHAIN-OF-TABLE and DP&PYAGENT show notable improvements when upgrading their backbone LLM from GPT-3.5-turbo to GPT-4o-mini. For example, CHAIN-OF-TABLE improves from 59.9 to 70.4 on WikiTQ and 80.2 to 85.8 on TabFact. DP&PYAGENT increases from 65.5 to 74.7 on WikiTQ and 80.0 to 89.9 on TabFact, highlighting the benefits of using a more powerful language model.

Our PLANTA outperforms all baselines on both test sets. With GPT-3.5-turbo, PLANTA scores 70.0 on WikiTQ and 82.0 on TabFact, outperforming DP&PYAGENT (65.5 on WikiTQ, 80.0 on TabFact) and CHAIN-OF-TABLE (59.9 on WikiTQ, 80.2 on TabFact). When using GPT-4o-mini, PLANTA further improves to 75.7 on WikiTQ and 90.4 on TabFact, surpassing DP&PYAGENT (74.7 on WikiTQ, 89.9 on TabFact) and CHAIN-OF-TABLE (70.4 on WikiTQ, 85.8 on TabFact).

Overall, PLANTA demonstrates state-of-the-art performance across different LLMs and datasets, providing clear evidence of the effectiveness of the proposed method for table understanding.

5.2 Ablation Study

To investigate the impact of each proposed component of PLANTA, we evaluate our ablated variants on WikiTQ and TabFact. Due to budget constraints, we evaluate the ablated variants on **1,000** randomly selected questions from each of the WikiTQ and TabFact test sets. Table 3 presents the contribution of each proposed component to PLANTA's overall performance with GPT-4o-mini.

Method	**WikiTQ**	**TabFact**
	GPT-3.5-turbo	
TEXT2SQL (2022)	52.9	64.7
CHAIN-OF-THOUGHT	53.5	65.4
BINDER (2023)	56.7	79.2
Dater (2023)	52.8	78.0
StructGPT (2023)	48.4	–
TabSQLify (2024)	64.7	79.5
CHAIN-OF-TABLE (2024)	59.9	80.2
DP&PYAGENT (2024)	65.5	80.0
Our PLANTA	**70.0**	**82.0**
	GPT-4o-mini	
CHAIN-OF-TABLE	70.4	85.8
DP&PYAGENT	74.7	89.9
Our PLANTA	**75.7**	**90.4**

Table 2: Performance results on the WikiTableQuestions (WikiTQ) and TabFact test sets. Rows 3 to 11 evaluate the table understanding capabilities of baseline methods and our PLANTA using GPT-3.5-turbo as the LLM. Results for previous methods are taken from their respective works, except for Dater, BINDER, and DP&PYAGENT. Since original Dater and BINDER relied on the now-decommissioned OpenAI Codex LLM, we extract their results based on GPT-3.5-turbo, reported in the CHAIN-OF-TABLE paper (Wang et al., 2024). Furthermore, DP&PYAGENT is tested only on a variant version of the original WikiTQ test set (i.e. not the same test set). Therefore, we run their official implementation (`https://github.com/Leolty/tablellm`) to report results on the original WikiTQ and the TabFact test sets with GPT-3.5-turbo. In rows 12-15, we run the official implementations of CHAIN-OF-TABLE (`https://github.com/google-research/chain-of-table`) and DP&PYAGENT using GPT-4o-mini to provide results with a faster and more cost-efficient LLM. Note that Wang et al. (2024) also report results of CHAIN-OF-TABLE using "PaLM-2" with 340B parameters (Anil et al., 2023). Since the PaLM-2 API has been decommissioned, we are unable to run PLANTA with "PaLM-2".

W/o planning: In this variant, long-term planning is excluded from PLANTA. Instead, the Planning expert relies solely on chain-of-thought (CoT) reasoning. In detail, the Planning expert is required to think step by step and generate a single request for Execution experts to handle. This process is repeated iteratively until a final answer is produced by the Assessment expert. As shown in Table 3, the exclusion of long-term planning significantly hurts PLANTA's performance, with accuracy dropping from 76.5 to 69.0 on WikiTQ and from 90.0 to 74.0 on TabFact. Our internal analysis indicates that the sharper decline on TabFact is due to the

Method	WikiTQ	TabFact
PLANTA$_{\text{GPT-4o-mini}}$	**76.5**	**90.0**
w/o planning	69.0	74.0
w/o search	56.0	62.5
w/o calculation	71.5	81.5
w/o comparison	75.5	88.0
w/o group experts	74.4	88.0
w/o assessment	75.0	85.3

Table 3: The performance of the full-component PLANTA with GPT-4o-mini, along with the results of the ablation study.

nature of fact verification tasks, which typically require only a true/false response. This simplicity may cause the Assessment expert to prematurely decide on an answer without verifying supporting evidence. Meanwhile, WikiTQ questions, which involve more searching tasks, encourage the model to continue processing until the result is found, reducing premature mistakes.

W/o search: In this variant, the Search expert is excluded from PLANTA, and search tasks are instead handled by the Comparison and Calculation experts. This leads to a significant drop in accuracy, from 76.5 to 56.0 on WikiTQ and from 90.0 to 62.5 on TabFact, even though the brains of the Comparison and Calculation experts can still reason to perform searches. These results highlight that search is a critical task, and our design of the Search expert enables the brains to effectively utilize predefined functions, resulting in more accurate search performance compared to relying on reasoning alone.

W/o calculation & W/o comparison: In these variants, the Comparison and Calculation experts are removed from PLANTA separately. Similar to the "W/o search" variant, these exclusions hurt PLANTA's accuracy. Specifically, removing the Calculation expert causes a sharper decline, with a 5% drop on WikiTQ and 8.5% on TabFact, compared to removing the Comparison expert, which results in a 1% drop on WikiTQ and 2% on TabFact, while the Search expert's brain still attempts reasoning to perform these tasks. These results highlight that LLMs' reasoning often struggles with comparison and even basic calculation.

W/o group experts: In this variant, all Execution experts are merged into a single unified expert responsible for handling search, calculation, and comparison tasks. Instead of using specialized prompts and predefined functions tailored to each expert's specific task, the unified expert uses a general prompt and has access to all predefined functions. This consolidation results in a 2.1% drop in accuracy on WikiTQ and a 2.0% drop on TabFact. These results demonstrate that Execution experts benefit significantly from prompts and predefined functions designed specifically for their specialized tasks, highlighting the value of maintaining task-specific experts within PLANTA.

W/o assessment: In this variant, the Assessment expert is excluded from PLANTA. In details, all outputs from the Execution experts are automatically updated into the long-term plan, and the final answer is generated once all short-term goals are completed. Table 3 shows that removing the Assessment expert reduces PLANTA's accuracy by 1.5% on WikiTQ and 4.7% on TabFact. This discrepancy mainly arises from the need to revise the plan to handle code execution errors or situations in which one or more steps in the plan are infeasible, leading to repetitive iterations without returning valid results.

6 Analysis

6.1 Error analysis

Table 4 presents the types of errors observed in PLANTA. The most frequent errors are related to planning and common sense, stemming from the LLMs' lack of *"real expert knowledge"*. As a result, they struggle to handle unpredictable data, such as *"TBA"* for time or *"note" columns* containing additional, contrasting information that alters the main context. This is consistent with the analysis in Subsection 6.2, where we demonstrate that improving the planning capabilities of the LLM leads to a substantial increase in accuracy. The Missing "hands" error, where no predefined function is available to assist reasoning, accounts for only 1.9% of the cases, emphasizing the robustness of our design for predefined functions. However, 11.7% of errors occur when the LLM mistakenly relies solely on reasoning instead of utilizing the available predefined functions to execute tasks accurately. In addition, 11.3% of errors are caused by generating invalid parameters for predefined functions. Hallucinations remain an unavoidable issue with LLMs, accounting for 11.3% of errors. LLMs can generate inaccurate final answers, even when accurate ones are explicitly provided in the final step. Despite its smaller percentage, acceptable answers reflect a need to handle vague questions.

Error Type	Description	%
Planning/Replanning	Errors related to incorrect relationships between steps, failure to handle exceptions within plan, and inability to detect execution errors when revising plan.	37.7%
Common sense	LLMs lack reasoning based on real-world knowledge. Example: When asked how many consecutive years 1990-1991 represent, LLMs answer *"2"*, while the correct answer is 1.	20.8%
Lazy executor	Errors where experts rely solely on LLM reasoning, even when predefined functions could assist, leading to incorrect results. Example: LLMs miscalculate 3 + 3 + 1 = 6, but a tool could compute it correctly.	11.7%
Parameter errors	Errors caused by generating invalid parameters for predefined functions, such as wrong data types or conditions.	11.3%
Hallucination	The plan is executed correctly, but the conclusion is wrong.	11.3%
Acceptable answers	Unclear questions lead to answers that are technically correct but not aligned with the expected response. For example, when asked *row listed before row 4*?, PLANTA lists rows 1 to 3, while the golden answer is row 3.	5.7%.
Missing "hands"	No predefined function is available to support the reasoning process	1.9%

Table 4: Error types in $\text{PLANTA}_{\text{GPT-4o-mini}}$ on the WikiTableQuestions test set. The total percentage does not add up to 100% because some samples contain more than one error.

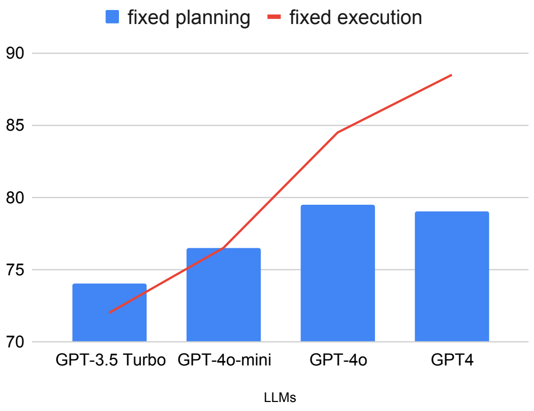


Figure 6: The impact of different LLMs on table understanding performance on the subset of 1000 WikiTableQuestions test questions used for the Ablation study. For "fixed planning", the LLM used for planning is set to GPT-4o-mini, while the LLM for execution tasks varies from GPT-3.5-turbo to GPT-4-mini to GPT-4o to GPT-4, increasing in reasoning capability. For "fixed execution", the LLM for execution tasks remains GPT-4o-mini, with the LLM for planning adjusted from GPT-3.5-turbo to GPT-4-mini to GPT-4o to GPT-4.

6.2 Improved LLMs Are Always the Key to Table Understanding?

As shown in Table 2, using LLMs with better reasoning capabilities notably improves table understanding performance. Here, we investigate whether improvements in LLMs always lead to significant performance gains. Our analysis focuses on two main aspects: [1] **Planning**, which determines how PLANTA chooses the best approach to answer a question, including the task of planning by the Planning expert and the task of revised planning by the Assessment expert; and [2] **Execution**, which involves performing the necessary tasks (such as search, comparison, and calculation) to find the relevant data within tables.

Figure 6 illustrates the impact of different LLMs on table understanding along these two aspects on the subset of 1000 WikiTableQuestions test questions used for the Ablation study. The results demonstrate that under the "fixed execution" setting, planning with better LLMs leads to a substantial improvement in accuracy for table understanding, with GPT-3.5-turbo achieving 72% and GPT-4 increasing this to 88.5%. In contrast, under the "fixed planning" setting, the improvement in execution tasks with better LLMs is far more limited, with accuracy rising from 74% to 79%. This contrast highlights the disproportionate influence of LLM reasoning on planning tasks compared to execution tasks. In other words, execution tasks appear to be less influenced by the model's reasoning power than planning tasks are, emphasizing the need for task-specific optimizations. Thus, by using powerful models for planning and more cost-effective models for execution, we can optimize both performance and resource efficiency.

7 Conclusion

We propose a novel method PLANTA to enhance table understanding by leveraging the long-term planning capabilities of LLMs. Our method focuses on two main goals: (1) enabling the execution of a long-term plan with tightly interconnected steps; (2) minimizing the inclusion of unnecessary details when solving short-term goals, thereby improving efficiency compared to CoT-based approaches. Experimental results show that PLANTA achieves new state-of-the-art performances on two benchmark datasets. Our PLANTA implementation is publicly available at: `https://github.com/nhungnt7/PLANTA`.

Limitations

Although our experiments have proven the effectiveness of our proposed method, there are still some limitations that can be improved in future work. While our approach encourages LLMs to engage in reasoning and solve tasks in a generalist manner, LLMs could benefit significantly from additional task-specific knowledge. For example, providing more targeted few-shot examples and explicitly including common exceptions could help the system handle rare or unpredictable scenarios better, as discussed in Subsection 6.1. Furthermore, future works can impose stricter constraints to encourage LLMs to use the "hands" of predefined functions effectively. This would minimize errors caused by LLMs attempting to rely solely on reasoning when predefined functions are better suited for the task, referred to as lazy executors in Subsection 6.1.

Acknowledgments

This work was supported by Monash eResearch capabilities, including M3.

This work was completed while Hoang Ngo and Dat Quoc Nguyen were at Movian AI, Vietnam.

References

Rohan Anil, Andrew M. Dai, et al. 2023. PaLM 2 Technical Report. *arXiv preprint*, arXiv:2305.10403.

Wenhu Chen. 2023. Large Language Models are few(1)-shot Table Reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *Transactions on Machine Learning Research*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *Proceedings of the 8th International Conference on Learning Representations*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding Language Models in Symbolic Languages. In *Proceedings of the 11th International Conference on Learning Representations*.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. TURL: Table Understanding through Representation Learning. *ACM SIGMOD Record*, 51(1):33–40.

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.

Julian Eisenschlos, Syrine Krichene, and Thomas Müller. 2020. Understanding tables with intermediate pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online. Association for Computational Linguistics.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 10764–10799.

Carlos Gemmell and Jeff Dalton. 2023. ToolWriter: Question Specific Tool Synthesis for Tabular Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16137–16148.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained Representations of Tabular Data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3446–3456.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A General Framework for Large Language Model to Reason over Structured Data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed Prompting: A Modular Approach for Solving Complex Tasks. In *Proceedings of the 11th International Conference on Learning Representations*.

Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. OpenTab: Advancing Large Language Models as Open-domain Table Reasoners. In *Proceedings of the 12th International Conference on Learning Representations*.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adria de Gispert, and Gonzalo Iglesias. 2023. An Inner Table Retriever for Robust Table Question Answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9909–9926.

Guang Liu, Jie Yang, and Ledell Wu. 2022a. PTab: Using the Pre-trained Language Model for Modeling Tabular Data. arXiv:2209.08060.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022b. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2024. Rethinking Tabular Data Understanding with Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 450–482.

Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2023. Prompt Engineering in Large Language Models. In *Proceedings of the International Conference on Data Intelligence and Cognitive Informatics*, pages 387–402.

Md Nahid and Davood Rafiei. 2024. TabSQLify: Enhancing Reasoning Capabilities of LLMs Through Table Decomposition. 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 5725–5737.

Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proceedings of the VLDB Endowment*, 16(4):738–746.

Ansong Ni, Srini Iyer, Dragomir Radev, Ves Stoyanov, Wen-tau Yih, Sida I Wang, and Xi Victoria Lin. 2023. LEVER: learning to verify language-to-code generation with execution. In *Proceedings of the 40th International Conference on Machine Learning*, pages 26106–26128.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. 2024. GPT-4 Technical Report.

Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Sohan Patnaik, Heril Changwal, Milan Aggarwal, Sumit Bhatia, Yaman Kumar, and Balaji Krishnamurthy. 2024. CABINET: Content Relevance-based Noise Reduction for Table Question Answering. In *Proceedings of the 12th International Conference on Learning Representations*.

Nitarshan Rajkumar, Raymond Li, and Dzmitry Bahdanau. 2022. Evaluating the Text-to-SQL Capabilities of Large Language Models . *arXiv preprint*, arXiv:2204.00498.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, page 645–654.

Dingzirui Wang, Longxu Dou, and Wanxiang Che. 2023a. A Survey on Table-and-Text HybridQA: Concepts, Methods, Challenges and Future Directions. *arXiv preprint*, arXiv:2212.13465.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *Proceedings of the 11th International Conference on Learning Representations*.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. TUTA: Tree-based Transformers for Generally Structured Table Pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding. In *Proceedings of the 12th International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Bohao Yang, Chen Tang, Kun Zhao, Chenghao Xiao, and Chenghua Lin. 2024. Effective Distillation of Table-based Reasoning Ability from LLMs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 5538–5550.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large Language Models are Versatile Decomposers: Decomposing Evidence and Questions for Table-based Reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 174–184.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. TableLlama: Towards open large generalist models for tables. 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 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. 2024. DocMath-eval: Evaluating math reasoning capabilities of LLMs in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16103–16120.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. In *Proceedings of the 11th International Conference on Learning Representations*.

A Prompt

Table 5 provides details of the custom-designed prompt for each component in the PLANTA system.

B Dataset description

WikiTableQuestions (WikiTQ): A question answering dataset based on HTML tables, each with a minimum of 6 rows and 5 columns. The questions were not generated using predefined templates but were hand-crafted by users, resulting in significant linguistic diversity. These questions span various domains and require operations such as table lookup, aggregation, superlatives, arithmetic operations, joins, and unions.

TabFact: A table-based binary fact verification dataset designed to determine whether a textual hypothesis is supported or refuted based on evidence provided in tables. The dataset presents a challenging task that requires both soft linguistic reasoning and hard symbolic reasoning. TabFact spans a wide range of operations, including aggregation, negation, superlatives, counting, comparative reasoning, and ordinal analysis.

Planning	**Router**	**Search**
You are a Planning expert. Your goal is to generate a plan to exclude a sequence of steps including SQL search (more detailed conditions in the requirements are better), calculation, and comparison based on the given table to get the answer to the question. For each step in the plan, dependencies on previous steps must be explicitly defined. Table: *{table}*. Question: *{question}*.	You are a task classification, your task is to classify the requirement type for the given task and route it to the appropriate expert. Please return the expert specialization based on the following guidance: 1. return 'search' if you need to search, conditional count the table for specific information. 2. return 'compare' if you need to compare two or more pieces of information. 3. return 'calculation' if you need to perform a calculation between numbers. Your task: *{short-term goal}*.	You are a Search expert. You have been tasked to reason and generate an SQL query to extract and conditional count specific information (rows) from the table. We allow you to independently determine how to resolve the assigned goal, such as utilizing the Chain-of-Thought or question decomposition approach, as long as the goal is solved. You can use the tool to execute an SQL query generated based on the question and given table and return the result. You might know the answer without running any code, but you should still run the code to get the answer. Given table: *{table}*. Your task: *{shot-term goal}*
Comparision	**Calculation**	**Assessment**
You are a Comparison expert. You must use the tools provided to complete the assigned task. We allow you to independently determine how to resolve the assigned goal, such as utilizing the Chain-of-Thought or question decomposition approach, as long as the goal is solved. You can use one tool multiple times and use many tools at one time in any order. You might know the answer without running any code, but you should still run the code to get the answer. Your tools include: *{list of predefined functions}*. Your task: *{shot-term goal}*.	You are a Calculation expert. You must use the tools provided to complete the assigned task. We allow you to independently determine how to resolve the assigned goal, such as utilizing the Chain-of-Thought or question decomposition approach, as long as the goal is solved. You can use one tool multiple times and use many tools at one time in any order. You might know the answer without running any code, but you should still run the code to get the answer. Your tools include: *{list of predefined functions}*. Your task: *{shot-term goal}* .	You are an Assessment expert. Your goal is to answer the question if sufficient relevant information is available or revise the plan if the results from the Execution experts fail to meet requirements or if the initial plan appears infeasible. Your original plan was this: *{plan}*. You have currently done the follow steps with the following results at template (step, result): *{past_steps}*

Table 5: Custom-designed prompts for each component in the PLANTA.

Derivational Probing: Unveiling the Layer-wise Derivation of Syntactic Structures in Neural Language Models

Taiga Someya♠ **Ryo Yoshida**♠ **Hitomi Yanaka**♠♡ **Yohei Oseki**♠
♠The University of Tokyo ♡RIKEN
`{taiga98-0809,yoshiryo0617,hyanaka,oseki}@g.ecc.u-tokyo.ac.jp`

Abstract

Recent work has demonstrated that neural language models encode syntactic structures in their internal *representations*, yet the *derivations* by which these structures are constructed across layers remain poorly understood. In this paper, we propose *Derivational Probing* to investigate how micro-syntactic structures (e.g., subject noun phrases) and macro-syntactic structures (e.g., the relationship between the root verbs and their direct dependents) are constructed as word embeddings propagate upward across layers. Our experiments on BERT reveal a clear bottom-up derivation: micro-syntactic structures emerge in lower layers and are gradually integrated into a coherent macro-syntactic structure in higher layers. Furthermore, a targeted evaluation on subject-verb number agreement shows that the timing of constructing macro-syntactic structures is critical for downstream performance, suggesting an optimal timing for integrating global syntactic information.

`https://github.com/osekilab/derivational-probing`

1 Introduction

Neural language models have achieved remarkable success across a wide range of natural language processing tasks. However, significant uncertainty remains regarding what these models truly learn and how they represent linguistic knowledge. This has spurred extensive research aimed at probing the linguistic capabilities of neural language models (Zhao et al., 2024; Chang and Bergen, 2024).

A prominent line of inquiry is *structural probing*, which directly analyzes word embeddings to uncover latent syntactic structures. For example, Hewitt and Manning (2019) demonstrated that the geometric organization of the word embedding space in BERT (Devlin et al., 2019) encodes syntactic distances defined over dependency parse trees, providing evidence that the model captures syntactic information. Yet, such work typically focuses on the static representations of the whole syntactic structures rather than the dynamic derivations by which these syntactic structures are built across layers. Understanding not just the resulting representations but also how they are built across layers is essential for a more comprehensive understanding and could also lead to better insights into how these representations are used.

Meanwhile, Tenney et al. (2019) introduced the *expected layer* metric and investigated how different layers in BERT encode different types of linguistic information (e.g., part-of-speech tagging, syntactic parsing, semantic role labeling, and coreference resolution), revealing that the model encodes linguistic abstractions in a manner reflecting a traditional NLP pipeline. However, their approach primarily relied on coarse-grained task accuracy measures, capturing only the overall effectiveness of each layer rather than examining the detailed, layer-wise construction of specific syntactic structures. Consequently, how the syntactic structures are built across layers remains underexplored.

In this paper, we fill this gap by proposing *Derivational Probing*—a method that integrates structural probing with the expected layer metric to probe derivation processes of syntactic structures in neural language models (Figure 1). Our proposed method allows us to investigate how *micro*-syntactic structures (e.g., subject and object noun phrases, prepositional phrases) and *macro*-syntactic structures (e.g., the relationship between the root verbs and their direct dependents) are constructed across layers.

Applying Derivational Probing to BERT (Devlin et al., 2019), our experiments reveal a clear bottom-up derivation, in which micro-syntactic structures emerge in lower layers and are gradually integrated into a coherent macro-syntactic structure in higher layers. Furthermore, our targeted analysis on a

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 93–104
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

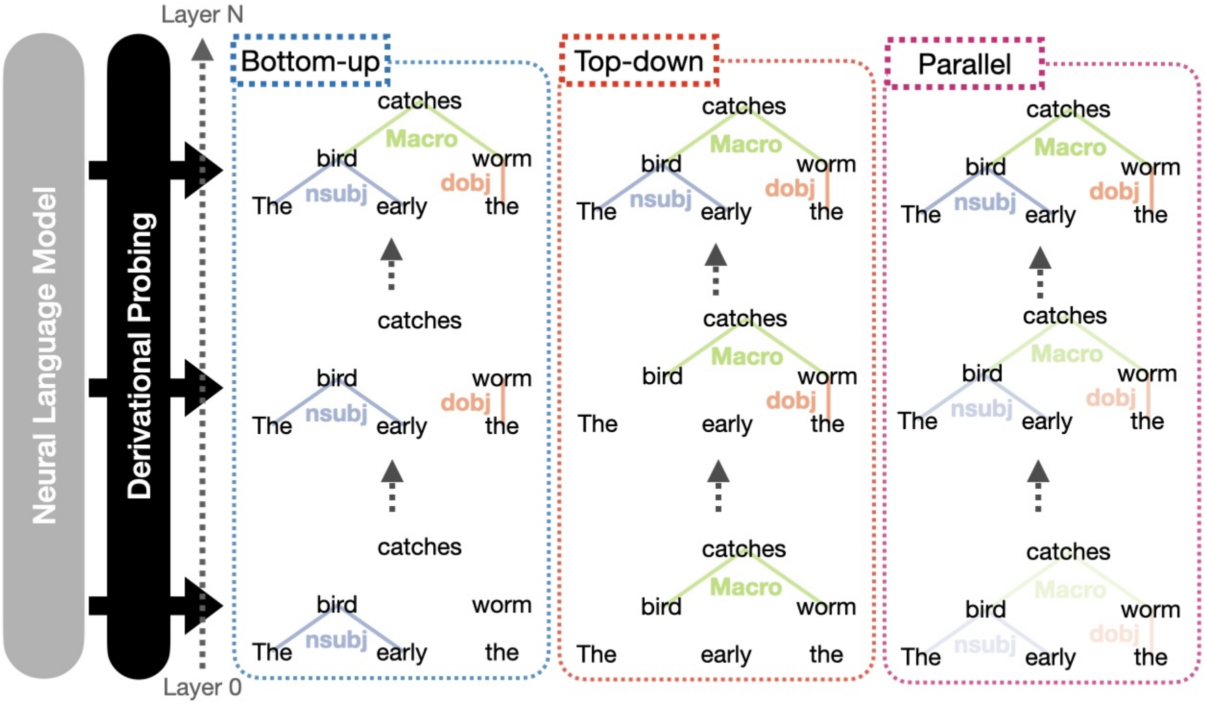


Figure 1: Derivational Probing investigates how syntactic structures are constructed across layers in neural language models. We illustrate three hypotheses for syntactic derivation. **Bottom-up:** Micro-syntactic structures, such as subject noun phrases or prepositional phrases, emerge in lower layers, and the macro-syntactic structure is formed in higher layers. **Top-down:** The macro-syntactic structure is formed in lower layers, with micro-syntactic structures refined in higher layers. **Parallel:** Micro- and macro-syntactic structures emerge in parallel.

subject-verb number agreement task shows that even when the final syntactic structure is correct, the specific layers at which the macro-syntactic structure is constructed significantly affect downstream performance. This suggests the existence of an optimal timing for integrating global syntactic information.

Overall, our findings offer new insights into the internal mechanisms by which neural language models construct syntactic structures and underscore the importance of examining derivation processes across layers to improve the interpretability of neural language models.

2 Related Work

Attention-based analyses (e.g., Clark et al., 2019; Vig and Belinkov, 2019) have demonstrated that certain transformer heads tend to align with dependency relations, providing evidence that Transformer language models capture linguistic dependency relations in their attention weights.

In contrast, Hewitt and Manning (2019) introduced a structural probe with a linear transformation from hidden representations into a space where Euclidean distances reflect dependency tree distances. This approach revealed that full syntactic trees are implicitly encoded in models such as BERT. Building on this, later work refined the approach by incorporating non-linear mappings (e.g., White et al., 2021), enforcing constraints such as orthogonality (Limisiewicz and Mareček, 2021), and using a controlled corpus to isolate the effect of syntax (Maudslay and Cotterell, 2021).

Other studies have refined structural probing by quantifying context-dependent syntactic signals in deeper layers—for example, conditional probing (Hewitt et al., 2021) and information gain metrics (Kunz and Kuhlmann, 2022)—but these methods focus on the performance of specific probing tasks (e.g., POS-tagging) rather than where the syntactic structures are constructed.

In contrast, our proposed method specifically tracks how each subgraph in the syntactic tree develops as information propagates through the network layers. By analyzing the evolution of individual syntactic components—from micro-syntactic structures to the assembly of the macro-syntactic structure—we offer a more granular perspective on the incremental construction of syntax, complementing and extending previous layer-wise analyses.

3 Technical Preliminaries

In this section, we review foundational methods from prior research: structural probing to assess the presence and quality of syntactic representations and the expected layer metric for quantifying how linguistic information gradually builds up across successive layers within language models.

3.1 Structural Probing

Hewitt and Manning (2019) introduced the *structural probe* as a method to evaluate whether contextual word representations encode syntactic information. Given a sentence $\mathbf{s} = w_1 \cdots w_t$, each token is represented by a d-dimentional contextual embedding $\mathbf{h}_i \in \mathbb{R}^d$ (e.g., the output embedding of a model like BERT). The goal of the structural probe is to find a linear transformation that maps these embeddings to a space where the Euclidean distances approximate the true syntactic distances between words.

Specifically, for any two words w_i and w_j in a sentence, we define the transformed distance as:

$$d_{\mathbf{B}}(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{B}\mathbf{h}_i - \mathbf{B}\mathbf{h}_j\|_2, \tag{1}$$

where $\mathbf{B} \in \mathbb{R}^{d' \times d}$ is a learnable projection matrix. The true syntactic distance, Δ_{ij}, is typically defined as the number of edges on the shortest path between w_i and w_j in the dependency parse tree of the sentence. The probe is trained by minimizing an objective that penalizes the discrepancy between the predicted distances and Δ_{ij}:

$$\mathcal{L} = \frac{1}{|\mathbf{s}|^2} \sum_{i=1}^{|\mathbf{s}|} \sum_{j=i+1}^{|\mathbf{s}|} \left| \Delta_{ij} - d_{\mathbf{B}}(\mathbf{h}_i, \mathbf{h}_j) \right|. \tag{2}$$

This formulation encourages the linear transformation $\mathbf{B}$ to capture the syntactic structure encoded in the contextual representations, enabling the recovery of parse trees via Prim's (1957) algorithm, a greedy algorithm that constructs minimum spanning trees by iteratively adding the lowest-weight edge connecting a new node to the growing tree.

3.2 Expected Layer

The *expected layer* metric introduced by Tenney et al. (2019) was initially developed to identify the layers within BERT responsible for solving various linguistic tasks. Specifically, the metric was used to capture at which layers broad linguistic abilities (e.g., part-of-speech tagging, syntactic parsing, semantic role labeling) emerge. Tenney et al. defined *scaler-mixed embeddings* $\boldsymbol{m}_i^\ell \in \mathbb{R}^d$ as the weighted average of embeddings from the bottom layer up to layer ℓ:

$$\boldsymbol{m}_i^\ell = \gamma \sum_{k=0}^{\ell} \mathbf{w}_k \, \mathbf{h}_i^k, \tag{3}$$

where $\mathbf{w} = \mathrm{softmax}(\mathbf{a})$ ($\mathbf{a} \in \mathbb{R}^{\ell+1}$) is learnable scalar mixing weights and γ is a learnable scaling factor, following Peters et al. (2018).

By measuring performance at layer index ℓ, denoted by $S(\ell)$, and tracking its improvements across layers, Tenney et al. defined the expected layer to reflect the layer at which the relevant linguistic task information is predominantly captured:

$$E[\ell] = \sum_{\ell=1}^{L} \frac{S(\ell) - S(\ell-1)}{\sum_{\ell=1}^{L}(S(\ell) - S(\ell-1))} \ell. \tag{4}$$

This is the weighted average of layer indices, where each layer's weight corresponds to its relative contribution to the overall performance improvement. It was initially proposed to broadly characterize the hierarchical progression of different linguistic capabilities within transformer models, rather than pinpointing the exact layers at which specific syntactic structures are built.

4 Derivational Probing

Building upon these prior techniques, we propose *Derivational Probing*, a novel method explicitly designed to investigate the dynamic construction of syntactic structures across the layers of neural language models.

Our approach effectively combines expected layer metric (Tenney et al., 2019) with the structural probing (Hewitt and Manning, 2019), enabling a detailed analysis of how syntactic information accumulates across model layers. Specifically, for each layer ℓ, we use scalar-mixed embeddings as defined in Eq. (3) and compute pairwise distances:

$$d_{\mathbf{B}_\ell}(\boldsymbol{m}_i^\ell, \boldsymbol{m}_j^\ell) = \|\mathbf{B}_\ell \boldsymbol{m}_i^\ell - \mathbf{B}_\ell \boldsymbol{m}_j^\ell\|_2. \tag{5}$$

We then train the transformation matrix $\mathbf{B_k}$ to minimize discrepancies with true dependency parse distances, analogous to structural probing.

This integration allows us to calculate the expected layer for each syntactic subgraph (micro- and macro-syntactic structures defined in detail later) and perform a fine-grained, quantitative analysis of their construction across model layers. We use the Unlabeled Undirected Attachment Score (UUAS) for each layer ℓ as $S(\ell)$, defined as the proportion of correctly predicted edges to the total number of edges in the reference dependency parse, without considering edge labels or direction.

To better understand the derivation strategy that models employ when constructing a syntactic tree, we introduce a distinction between *macro-syntactic structures* (the root verb with its direct dependents) and *micro-syntactic structures* (local components, such as subordinate phrases like `nsubj`) (Figure 2). This distinction is motivated by our interest in whether models construct syntactic trees top-down, bottom-up, or in a parallel fashion. To empirically evaluate which of these hypotheses is most plausible, we adopt the following methodological approach: For both micro-syntactic structures and macro-syntactic structures, we (1) construct the full parse tree using a minimum spanning tree algorithm, (2) extract the relevant edges (as highlighted in Figure 2), and (3) compute the UUAS by comparing these edges to the reference parse. By tracking UUAS improvements across layers, we calculate the expected layer $E[l]$ for each structure, revealing the layers at which different syntactic subgraphs are effectively constructed.

We next provide detailed descriptions of each hypothesis.

Bottom-up derivation. A bottom-up derivation first constructs micro-syntactic structures and subsequently integrates these into macro-syntactic structures, ultimately forming a complete dependency tree. We refer to this as a "bottom-up derivation" because it resembles the construction order of the arc-standard transition-based dependency parser (Nivre, 2004). Arc-standard parsing utilizes a stack-based transition system and constructs a dependency tree in a bottom-up manner: dependents must be fully processed and attached to their heads before those heads themselves are incorporated into macro-syntactic structures. Under this hypothesis, models initially identify micro-syntactic structures—such as the internal phrase structures of subjects and objects—in lower layers, which are then progressively combined into a coherent macro-syntactic hierarchy at higher layers.

Top-down derivation. A top-down derivation, in contrast, begins by establishing macro-syntactic structures and subsequently refines these by incorporating detailed micro-syntactic dependencies. We term this approach a "top-down derivation" because its construction order aligns closely with the head-driven transition-based parser proposed by Hayashi et al. (2012). Their algorithm explicitly predicts dependent nodes from head nodes, progressively building syntactic structures from head to dependent, thus genuinely following a top-down, predictive parsing order. Under this hypothesis, models prioritize the recognition of macro-syntactic structures before refining micro-syntactic struvtures.

Parallel derivation. Finally, an alternative hypothesis is that models construct micro- and macro-syntactic structures concurrently, with local dependencies and the global structure forming at roughly the same rate across layers. This hypothesis is less clearly aligned with traditional dependency parsing algorithms, as most classical approaches tend to favor either bottom-up or top-down derivations.

Notes on the term "derivation". Here, we explicitly use the term *derivation* (strategy) throughout this paper rather than "parsing strategy" to clearly distinguish two related but distinct concepts. While "parsing strategy" generally refers to methodological choices for *incrementally* constructing a parse tree (such as bottom-up or top-down), our use of "derivation" specifically captures an *atemporal* process describing how syntactic structures progressively emerge across the internal layers of a language model given the full sentence context, emphasizing *layer-wise* structural development rather than sequential, *left-to-right* incremental processing.

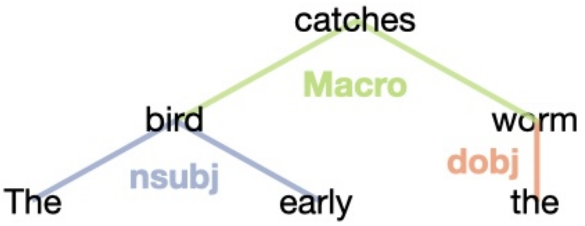


Figure 2: *Macro-syntactic structure* (`Marco`) and *micro-syntactic structures* (`nsubj` and `dobj`).

5 Experimental Setup

5.1 Data

We utilize the Wikitext-103 dataset (Merity et al., 2016) as our primary source of natural language, parsing each sentence with spaCy's dependency parser (EN_CORE_WEB_LG) (Honnibal et al., 2020). To focus on the language model's ability to

Structure Set	**Example Sentence**
`Marco`, `nsubj`, `dobj`	The concert caused a major stir.
`Marco`, `nsubj`, `prep`	The match ended in a goalless draw.
`Marco`, `nsubj`, `attr`	Her parents were music professors.
`Marco`, `nsubj`, `prep`, `dobj`	The film received positive reviews from critics.

Table 1: Example sentences for each primary structure set described in §5.1

construct syntactic structures in a clear-cut setting, we restrict our analysis to single-clause sentences by excluding those with relative clauses or clausal subjects. Additionally, we filter out sentences containing dependency relations such as "dep" (unclassified dependents) and punctuation marks other than sentence-final punctuation to minimize noise.

Following the definitions introduced in the previous section (§4), we group sentences based on dependency relations emanating from the root verb, thereby distinguishing between the overall (macro-syntactic; `Marco`) structure and subordinate (micro-syntactic) structures. We retain only those groups that represent more than 10% of the data, focusing our analysis on the predominant structure sets. This filtering results in four primary structure sets (See Table 1 for examples): (1) `Marco` with micro relations `nsubj` and `dobj`; (2) `Marco` with micro relations `nsubj` and `prep`; (3) `Marco` with micro relations `nsubj` and `attr`; and (4) `Marco` with micro relations `nsubj`, `prep`, and `dobj`.

From the resulting dataset, we randomly sample 50,000 sentences, partitioning them into 40,000 for training, 5,000 for validation, and 5,000 for testing.

5.2 Models

We employ two pre-trained language models: BERT-base[1] and BERT-large[2] (cased) (Devlin et al., 2019). BERT-base uses 12 layers, 12 heads, and a 768-dimensional hidden state, while BERT-large uses 24 layers, 16 heads, and a 1024-dimensional hidden state. These models provide a range of capacities, allowing us to investigate differences in how syntactic structures are constructed across models.

We focus specifically on BERT because our method is designed to examine the *atemporal*, layer-wise derivation of syntactic structures given entire sentences. In contrast, autoregressive language models such as GPT-2 process information incrementally in a left-to-right manner, and the *temporal*, token-wise derivation of syntactic structures cannot be probed via our method (cf. Eisape et al., 2022). Nevertheless, our method is still applicable to the word embeddings of autoregressive language models such as GPT-2, and we report GPT-2 results in App. A.

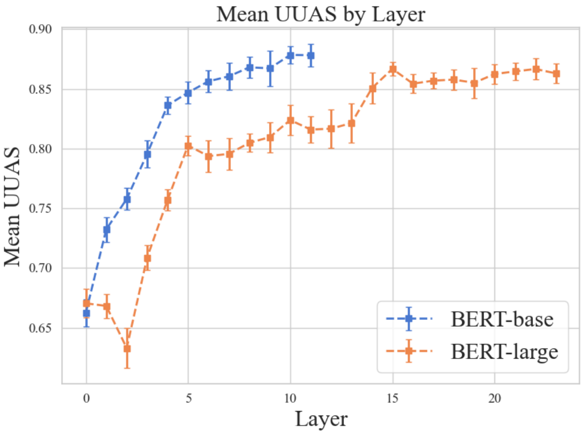


Figure 3: Global UUAS by each layer for each model. Error bars represent standard deviations across 5 random seeds.

For each model, we probe all layers to determine the progression of syntactic information and compute the expected layer at which specific structures emerge. We conduct training with five different random seeds and report the average performance along with the standard deviation. Additional hyperparameters and training details are provided in App. B.

6 Results

6.1 Overall UUAS Performance

As a sanity check to verify whether our models exhibit overall trends similar to those reported in previous studies, we conducted an experiment measuring the test set UUAS for overall sentence structures across layers for each model (Figure 3). BERT-base and BERT-large display similar trends,

---

[1] https://huggingface.co/google-bert/bert-base-cased

[2] https://huggingface.co/google-bert/bert-large-cased

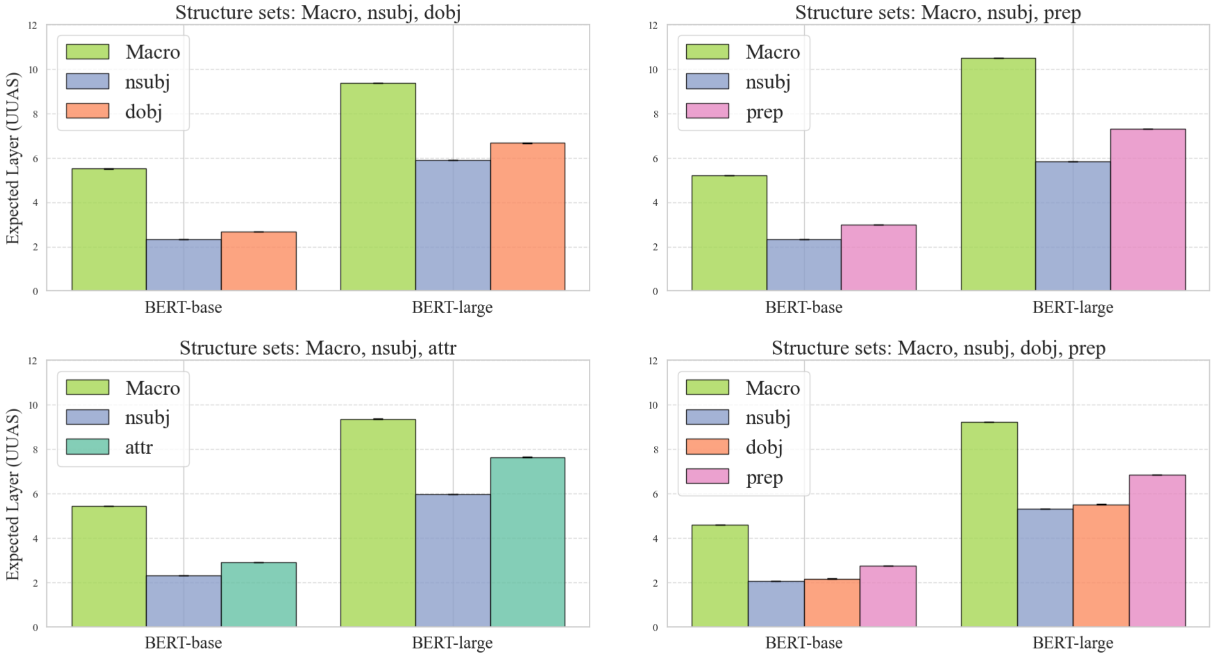


Figure 4: Expected layer for each model across different structure sets. Error bars represent standard deviation across 5 random seeds.

with the UUAS score saturating around the middle layers. BERT-large shows slightly slower improvement, likely reflecting its deeper architecture and larger capacity. These trends mostly align with previous findings (Hewitt and Manning, 2019) that neural language models tend to exhibit peak UUAS performance in their middle layers. However, unlike previous studies, we do not observe a decrease in average UUAS in later layers, which we attribute to our method of computing word embeddings as a weighted average from layer 0 to layer ℓ (Eq. (3)).

6.2 Expected Layer Across Structure Sets

Figure 4 summarizes the expected layers for each syntactic structure within the four primary structure sets (§5.1), for both BERT-base and BERT-large.

In both BERT-base and BERT-large, the macro-syntactic structure consistently exhibits the highest expected layer across all sets, whereas micro-syntactic structures such as `nsubj`, `dobj`, and `prep` tend to appear in lower layers. This suggests a *bottom-up* derivation process in which micro-syntactic structures (e.g., subject or object phrases) are constructed earlier, and these components are gradually integrated into a coherent macro-syntactic structure in later layers. This observation is consistent with prior work on BERT, which shows that local information (e.g., POS tags) is captured early, while more abstract global structures emerge later (cf. Tenney et al., 2019). Notably, this pattern holds for both BERT-base and BERT-large, although the overall expected layers are slightly higher in BERT-large—likely reflecting its deeper architecture and larger capacity.

7 Detailed Analysis: Subject-Verb Agreement Task

7.1 Experimental Setup

To investigate how the process and layers involved in syntactic structure construction relate to model performance, we conduct a detailed analysis on subject-verb agreement using sentences with intervening nouns (“attractors”), following the approach of Marvin and Linzen (2018) with some modification. We sampled 1,000 positive (grammatical) and 1,000 negative (ungrammatical) sentences. All of the sampled sentences have a subject followed by a prepositional phrase, a verb, and a direct object noun phrase. They thus are categorized into `Marco`, `nsubj`, `dobj` structure sets. In our modification of their templates, each of `Marco`, `nsubj`, `dobj` is required to contain more than one word.

a. The senators behind the brilliant architect avoid spicy dishes.
b. *The senators behind the brilliant architect avoids spicy dishes.

This ensures that we can extract meaningful subgraphs within each syntactic substructure.

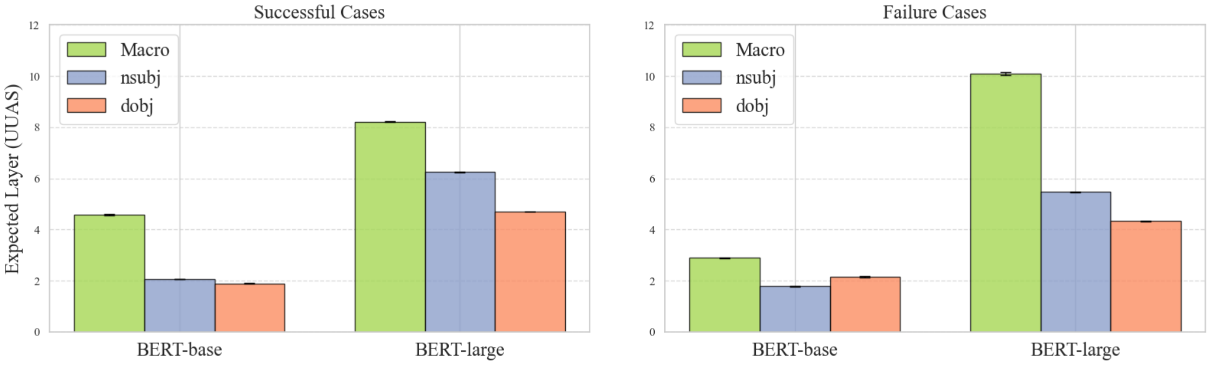


Figure 5: Expected layers for syntactic structures in successful and failed subject-verb agreement cases. Error bars show standard deviations across 5 random seeds.

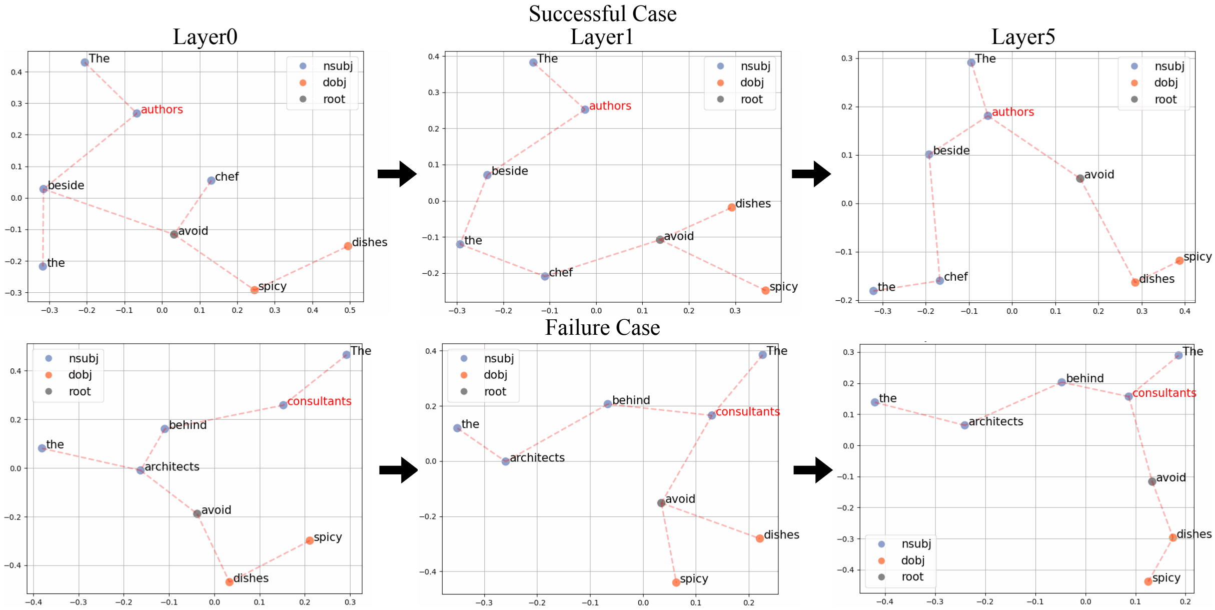


Figure 6: Derivation process visualizations for BERT-base on subject-verb agreement for a successful case ("The authors beside the chef avoid spicy dishes.") and a failure case ("The consultants behind the architects avoid spicy dishes."). Red highlights indicate the correct subject.

We first evaluate model performance on this task by computing pseudo-whole-sentence probabilities (Salazar et al., 2020). Specifically, we calculate the probability of each token by masking it one by one and then aggregate these token-level probabilities to derive an overall sentence probability. We expect the model to assign higher pseudo-probabilities to grammatical sentences compared to ungrammatical ones. We then analyze how the syntactic construction process differs between cases where the model performs well and those where it fails.

Furthermore, to visualize the evolution of syntactic structures across model layers, we employ Multidimensional Scaling (MDS). Specifically, we apply scikit-learn's MDS implementation (Pedregosa et al., 2011) with default parameters to word embeddings projected by our structural probe, allowing us to illustrate clearly how syntactic representations develop across different model layers.

7.2 Results

Overall, BERT-base correctly assigned higher pseudo-whole-sentence probabilities to grammatical sentences in 984 out of 1,000 examples, whereas BERT-large achieved correctness in 983 out of 1,000 cases. Despite their similar overall accuracies, we observe distinct patterns between BERT-base and BERT-large (Figure 5).

BERT-base. BERT-base frequently failed when macro-syntactic structures were established prematurely, potentially restricting the incorporation of essential micro-syntactic details. As illustrated in

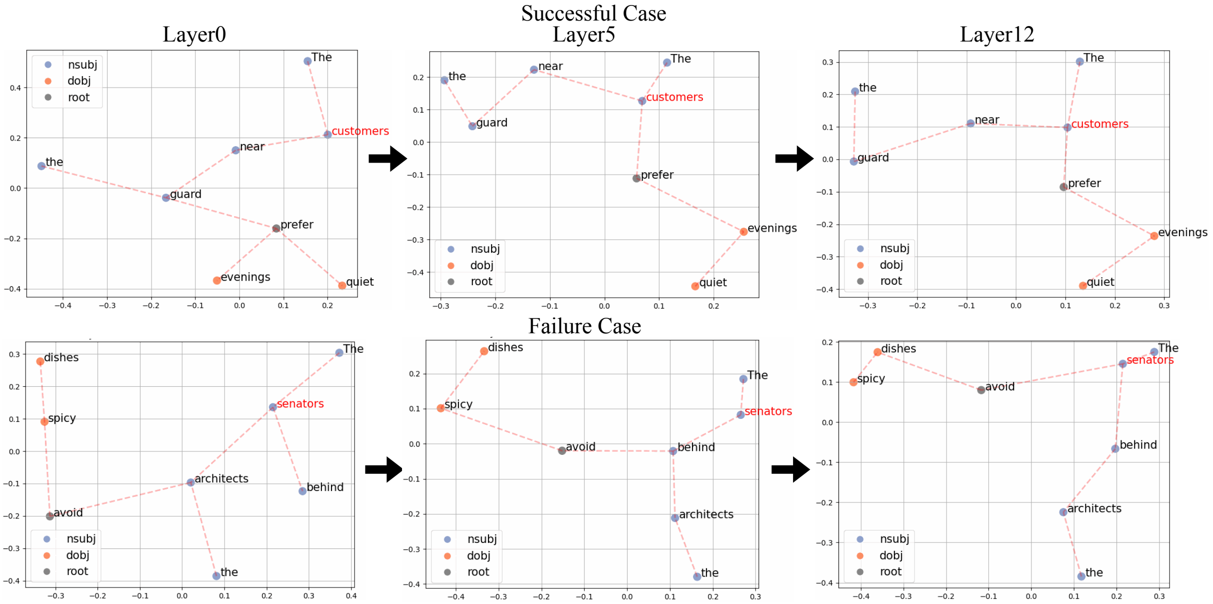


Figure 7: MDS visualizations of syntactic structure evolution in BERT-large for subject-verb agreement for a successful case (“The customers near the guard prefer quiet evenings.”) and a failure case (“The senators behind the architects avoid spicy dishes.”). Red highlights indicate the correct subject.

Figure 6, successful cases show a sequential pattern where BERT-base first constructs micro-syntactic structures within the subject phrase in early layers, subsequently aligning the subject (*authors*) with the verb (*avoid*) around layer 5 after stabilizing the internal subject dependencies. In contrast, failure cases reveal premature alignment of macro-syntactic structures, with the subject (*consultants*) prematurely linked to the verb (*avoid*) before fully establishing necessary micro-syntactic details. This premature commitment might have negatively impacted the overall syntactic representation, disrupting correct subject-verb agreement.

BERT-large. BERT-large exhibited higher expected layers for macro-syntactic structures in failure cases, suggesting delayed integration of macro-syntactic information. Figure 7 illustrates representative successful and unsuccessful cases for BERT-large. Successful predictions demonstrate early alignment of the subject (*customers*; highlighted in red) with the verb (*prefer*) around layer 5, facilitating accurate subject-verb agreement. Conversely, in unsuccessful cases, this alignment emerged considerably later (around layer 12), highlighting delayed macro-syntactic integration.

These analyses suggest an optimal intermediate range of layers for integrating macro-syntactic information. Forming macro-syntactic structures either prematurely or excessively late can negatively affect syntactic processing, highlighting the importance of appropriately timed integration for accurate predictions. These visualizations underscore how deviations from this optimal timing contribute to subject-verb agreement errors.

8 Discussion and Conclusion

In this paper, we introduced *Derivational Probing*—a method that integrates structural probing with an expected layer metric to trace the construction process of syntactic structures in neural language models. Our experiments revealed that BERT models tend to build micro-syntactic dependencies first and gradually assemble them into a coherent macro-structure.

BERT’s bidirectional context supports a stepwise, bottom-up construction—starting with the formation of local, micro-syntactic structures and culminating in a fully integrated macro representation. These findings offer valuable insights into the internal mechanisms by which deep neural models construct syntactic trees and highlight the importance of examining layer-wise structural formation for improved model interpretability.

One promising direction for future research is to incorporate multilingual probes, which will help determine whether these syntactic structures generalize beyond English or are not mere artifacts of the particular training corpus. Another exciting direction would be to explore incremental parsing strategies in autoregressive language models as an alternative to non-incremental derivation processes across layers (cf. Eisape et al., 2022), which could yield further insights into the syntactic knowledge of neural language models.

Limitations

First, our experiments were conducted on only two neural language models (BERT-base and BERT-large). It remains unclear whether similar results would be obtained for larger models or other architectural variants. However, our method is applicable to any open neural model, making it feasible to extend this analysis to a broader range of models in future research.

Second, this study focused solely on English data. It is uncertain whether similar layer-wise syntactic structure construction patterns would be observed when applying our method to other languages. Nevertheless, our approach is language-agnostic, making cross-linguistic analysis an important direction for future work.

Furthermore, semantic cues may influence the results of syntactic probes. Our study does not fully account for these potential semantic confounds. Future research should consider methods to more rigorously isolate syntactic information, such as using Jabberwocky sentences as demonstrated by Maudslay and Cotterell (2021).

Lastly, our method relies on dependency parsing, primarily due to the use of the structural probe from Hewitt and Manning (2019), which analyzes distances between tokens in the embedding space. This approach is inherently tied to formalisms like dependency grammar that focus on relationships between terminal symbols (tokens). As a result, our method may not be directly applicable to other grammatical theories or parsing approaches that involve non-terminal symbols, such as constituency grammar. This limitation arises because analyzing distances between tokens does not capture the hierarchical structures represented by non-terminals. Future work could explore adapting our method or developing new probing techniques that can handle non-terminal representations to verify the generalizability of our findings.

Ethical considerations

The training corpus is extracted from public web pages and thus could be socially biased, despite its popular use in the NLP community.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP24H00087, JP24H00809 and JST PRESTO Grant Numbers JPMJPR21C2, JPMJPR21C8. TS was also supported by JST BOOST NAIS.

References

Tyler A. Chang and Benjamin K. Bergen. 2024. Language Model Behavior: A Comprehensive Survey. *Computational Linguistics*, 50(1):293–350.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Katsuhiko Hayashi, Taro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2012. Head-driven transition-based parsing with top-down prediction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 657–665, Jeju Island, Korea. Association for Computational Linguistics.

John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. Conditional probing: measuring usable information beyond a baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Jenny Kunz and Marco Kuhlmann. 2022. Where does linguistic information emerge in neural language models? measuring gains and contributions across layers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4664–4676, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Tomasz Limisiewicz and David Mareček. 2021. Introducing orthogonal constraint in structural probes. 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 428–442, Online. Association for Computational Linguistics.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Joakim Nivre. 2004. Incrementality in deterministic dependency parsing. In *Proceedings of the Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, pages 50–57, Barcelona, Spain. Association for Computational Linguistics.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

R. C. Prim. 1957. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36(6):1389–1401.

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

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. 2021. A non-linear structural probe. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 132–138, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*, 15(2).

A The Experimental Results for GPT-2 Models

Figures 8 and 9 show the experimental results with the same experimental setup as §5, but conducted with GPT-2 small[3]/medium[4] (Radford et al., 2019). In contrast to BERT, GPT-2 (both small and medium) displays a more *parallel* derivation: the expected layer values for both micro-syntactic and macro-syntactic structures are closely aligned, suggesting parallel derivation rather than a bottom-up or top-down derivation.

B Hyperparameters

Hyperparameters for our experiments are shown in Table 2. All models were trained and evaluated on 4× NVIDIA RTX A5000 (24GB). The total computational cost for all experiments in this paper is about 120 GPU hours.

Optimizer	Adam
Learning rate	1e-3
Number of epochs	40
Learning rate scheduler	ReduceLROnPlateau
Batch size	32

Table 2: Hyperparameters for our experiments

C License of the Data and Tools

The licenses of the data and tools used in this paper are summarized in Table 3. We confirmed that all the data and the tools were used under their respective license terms.

Data/tool	License
`spacy` (Honnibal et al., 2020)	MIT
`transformers` (Wolf et al., 2020)	Apache 2.0
`Wikitext-103` (Merity et al., 2016)	CC-BY-SA 3.0

Table 3: License of the data and tools

---

[3] `https://huggingface.co/openai-community/gpt2`

[4] `https://huggingface.co/openai-community/gpt2-medium`

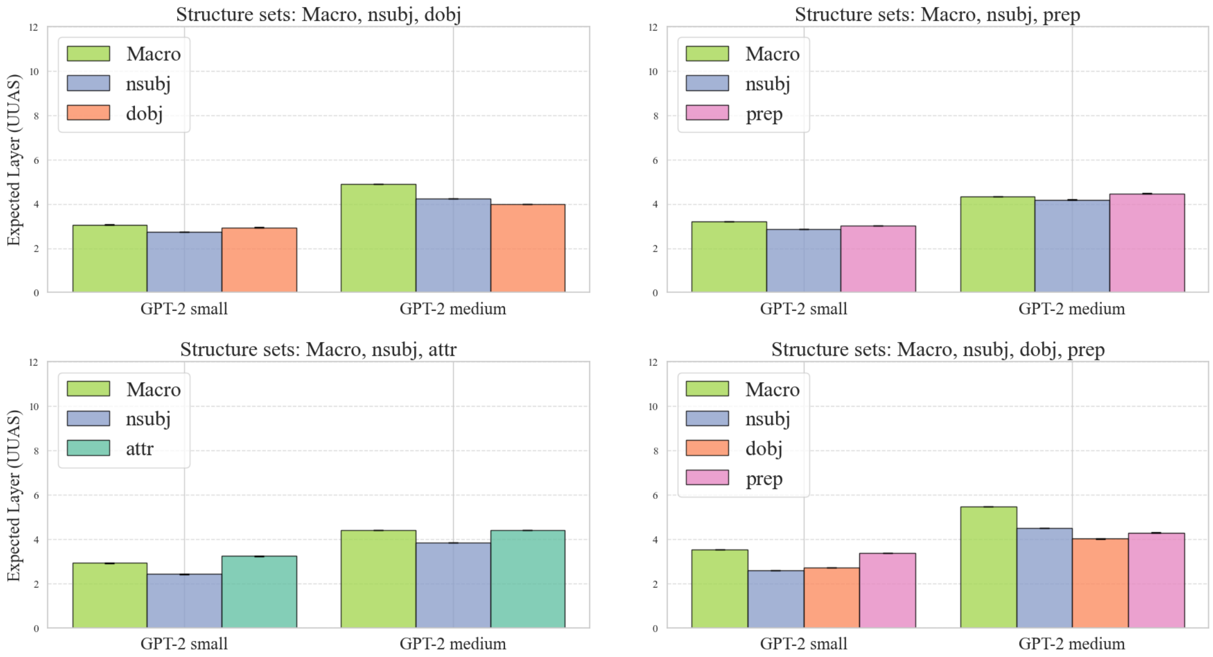


Figure 8: Expected layer for each GPT-2 model across different structure sets. Error bars represent standard deviation across 5 random seeds.

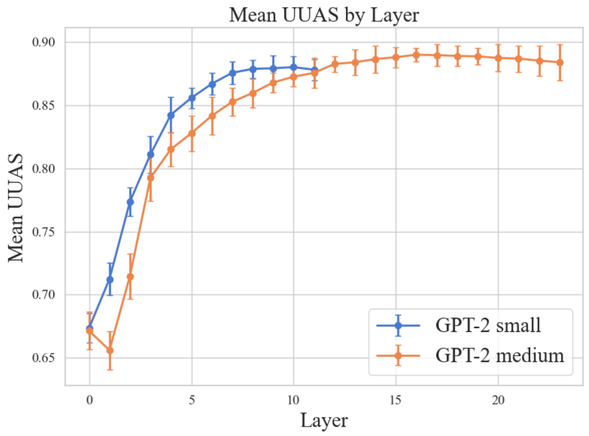


Figure 9: Global UUAS by each layer for each GPT-2 model. Error bars represent standard deviations across 5 random seeds.

Short-circuiting Shortcuts: Mechanistic Investigation of Shortcuts in Text Classification

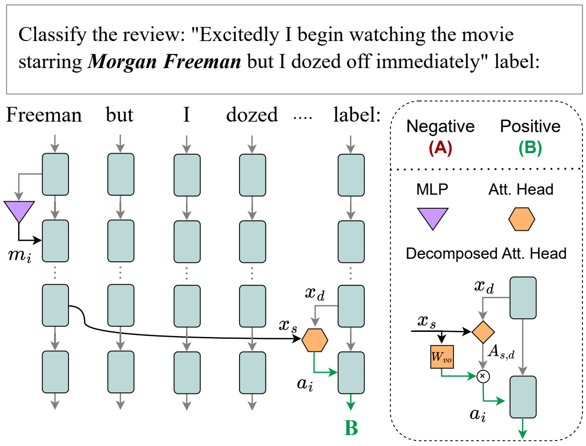
Leon Eshuijs
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
`l.eshuijs@vu.nl`

Shihan Wang
Utrecht University
Utrecht, the Netherlands
`s.wang2@uu.nl`

Antske Fokkens
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands
`antske.fokkens@vu.nl`

Abstract

Reliance on spurious correlations (shortcuts) has been shown to underlie many of the successes of language models. Previous work focused on identifying the input elements that impact prediction. We investigate how shortcuts are actually processed within the model's decision-making mechanism. We use actor names in movie reviews as controllable shortcuts with known impact on the outcome. We use mechanistic interpretability methods and identify specific attention heads that focus on shortcuts. These heads gear the model towards a label before processing the complete input, effectively making premature decisions that bypass contextual analysis. Based on these findings, we introduce Head-based Token Attribution (HTA), which traces intermediate decisions back to input tokens. We show that HTA is effective in detecting shortcuts in LLMs and enables targeted mitigation by selectively deactivating shortcut-related attention heads.[1]



Classify the review: "Excitedly I begin watching the movie starring ***Morgan Freeman*** but I dozed off immediately" label:

Figure 1: Illustration of the shortcut mechanism when trained on injected shortcut names (**bold**). Later layer attention heads focus on shortcut tokens and change the prediction based on information from early MLP layers. After decomposing the attention head, we find how the shortcut tokens are processed and apply these findings to construct our feature attribution method (HTA).

1 Introduction

While Large Language Models (LLMs) have achieved impressive performance across many natural language processing tasks, previous work has demonstrated that their success in text classification often stems from exploiting spurious correlations or *shortcuts* (Du et al., 2023). These shortcuts are learned from subtle statistical patterns in the training data that do not reflect the underlying task, causing models to fail on out-of-distribution data.

Prior work on shortcuts has focused on identifying shortcuts (Du et al., 2021), often via targeted input modifications known as behavioral testing (Alzantot et al., 2018; Ribeiro et al., 2020). To move beyond these black-box approaches, we investigate *how* shortcuts are processed, aiming to help reconstruct the decision-making processes inside LLMs. In particular, we examine the mechanisms within LLMs responsible for processing shortcuts. Figure 1 provides an overview of our approach. We expect that shortcut behavior occurs when the model primarily relies on isolated tokens rather than contextual information from the entire sentence. In contrast, proper classification should involve all tokens, with the final decision emerging only after the model processes the entire input.

We use mechanistic interpretability (Olah et al., 2020; Elhage et al., 2021), which has demonstrated impressive progress in locating target mechanisms for various tasks. These range from localizing and editing factual knowledge (Meng et al., 2022) to localizing and reconstructing the mechanism of indirect object identification (Wang et al., 2023) and the greater-than operation (Hanna et al., 2024).

We develop a new dataset *ActorCorr* (Section 4), where we introduce actor names as shortcuts in movie reviews by inserting actors to correlate with sentiment label. We confirm experimentally that

---

[1] Code available at `https://github.com/watermeleon/shortcut_mechanisms`

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 105–125
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

the model uses these shortcuts for prediction. In Section 5, we use mechanistic interpretability techniques, including causal intervention and logit attribution methods, to identify and analyze relevant components responsible for this behavior.

Our experiments reveal that attention heads in later layers focus on shortcuts and generate label-specific information based on the shortcut tokens, changing the output prediction. This demonstrates that the model effectively makes intermediate label predictions before processing the complete input. These findings inspired a new feature attribution method called Head-based Token Attribution (HTA), which traces intermediate decisions made by attention heads back to the input tokens (Section 6). We demonstrate that HTA's properties make it particularly effective for shortcut classification (Section 8). Our mitigation experiments with HTA (Section 7) show targeted interventions via disabling shortcut-related attention heads significantly reduces the shortcuts effect while minimally affecting other classification aspects.

2 Related work

Evaluating shortcuts Shortcut detection methods in NLP tend to use previously reported shortcuts in existing datasets (Pezeshkpour et al., 2021; Friedman et al., 2022), such as the appearance of numerical ratings present in reviews (Ross et al., 2021), or the presence of lexical overlap between the hypothesis and the premise (Naik et al., 2018). Other work injects their own shortcuts into datasets. Bastings et al. (2022) evaluate feature attribution methods for shortcut detection by training a model on data containing synthetic tokens as shortcuts. Similar to our work, Pezeshkpour et al. (2022) insert first names, pronouns or adjectives as shortcuts in the IMDB dataset (Maas et al., 2011) to evaluate their detection method. These studies only address extreme cases of shortcuts (i.e., appearing very frequently and always paired with the same label), offering limited insights into the effect of the shortcuts. We therefore create our own dataset with less extreme shortcuts of which the impact is known.

Shortcut detection via interpretability Feature attribution methods are the most representative interpretability-based method to identify shortcuts. These methods explain output predictions by assigning importance scores to individual input tokens. However, different methods often provide diverging explanations for the same input (Madsen et al., 2022; Kamp et al., 2024). Moreover, for shortcut detection, Bastings et al. (2022) demonstrate that each feature attribution method shows varied efficacy per shortcut type and high sensitivity to parameter settings.

Wang et al. (2022) offer a first step towards automatic shortcut detection via inner-interpretability methods (Räuker et al., 2023). Their method computes importance through attention weights and token frequency in the final BERT layer. Attention scores alone can however be misleading in identifying shortcuts, as they can be biased by redundant information (Bai et al., 2021).

Mechanistic Interpretability Mechanistic Interpretability aims to reverse engineer the computation of neural networks into human understandable algorithms (Olah et al., 2020; Elhage et al., 2021). To achieve this, a range of interpretability techniques have been proposed to localize relevant components or help understand the functionality of specific components. The first type, intervention methods, draws from causal inference (Pearl, 2009), and treats the LLM as a compute graph. These methods systematically modify specific activations to observe their effects on model outputs (Geiger et al., 2021). Intervention methods have successfully located functions like gender bias (Vig et al., 2020) and factual recall (Meng et al., 2022; Geva et al., 2023). Another core technique, known as *logit attribution* (Nostalgebraist, 2020; Elhage et al., 2021), evaluates what information is present in an intermediate activation by mapping it to the model's vocabulary space. For example, Yu et al. (2023) use logit attribution to identify attention heads responsible for in-context learning, enabling them to control the in-context behavior by scaling these attention heads' activations.

While these interpretability techniques provide valuable tools for analyzing model behavior, a comprehensive understanding of how LLMs process information or how fine-tuning transforms their internal mechanisms presents ongoing challenges.

3 Background and Notation

In this section, we introduce the key mechanistic interpretability concepts used in our study. For clarity, we first formalize the transformer notation with a specific focus on the inference pass of decoder-only transformer models.

3.1 The Transformer

Figure 2 provides a schematic representation of a transformer. For the transformer (Vaswani et al., 2017), the input text is first converted into a sequence of N tokens $t_1, ..., t_N$. Each token t_i is then transformed into an embedding x_i using the embedding matrix W_e, resulting in the embedding sequence $x^0 \in \mathbb{R}^{N \times d_{resid}}$, where 0 indicates the model's input layer.

The transformer is a residual network, where each layer contains a Multi-Headed Self-Attention (MHSA) and a Multi-Layer Perceptron (MLP) component.[2] The connection from the input embedding to the output embedding to which these components add their embedding, or activation, is called the *residual stream*. The activation of the MHSA is computed $a^l = MHSA(x^l)$, and following Elhage et al. (2021), can be decomposed as the sum of each attention head's contribution, $a^{l,h}$, so that the final activation is reconstructed as $a^l = \sum_h a^{l,h}$. Then MLP activation is computed as $m^l = MLP(x^l + a^l)$, resulting in the new residual embeddings: $x^{l+1} = x^l + m^l + a^l$. After the last layer the final embeddings are projected to a vector the size of the vocabulary, using the unembedding matrix W_u to obtain the logits for each embedding. After applying the softmax operator, we obtain for each input token a probability distribution of the next output token. For our classification task, we only use the embedding x_T^L of the last token stream T of the last layer L for predicting the class.

3.2 Mechanistic Interpretability

Following Wang et al. (2023), we formulate an LLM as a computational graph M with nodes representing individual components (e.g., MLPs or attention heads), and edges representing their interactions through activations. Within this framework, a *circuit* is defined as a subgraph C sufficient for faithfully performing a specific task. To investigate circuits responsible for processing shortcuts, we employ two key analysis techniques: logit attribution and path patching.

Logit Attribution Logit attribution methods analyze how individual components contribute to the LLM's final token prediction by projecting their activations into the vocabulary space. This is possible because the final output embedding is a linear combination of all previous activations (Elhage et al., 2021). Normally, W_u is used to obtain the logits over the vocabulary for the final residual stream vector, and after applying the softmax, it provides us with the probability distribution over tokens. Direct logit attribution (Nostalgebraist, 2020; Elhage et al., 2021) applies W_u to analyze intermediate activations from individual components, such as attention heads $a^{l,h}$ or MLP layers m^l. Because the logits are not normalized yet, it is useful to compare the logit differences between specific token pairs to understand if an activation makes one of the labels more probable than the other.

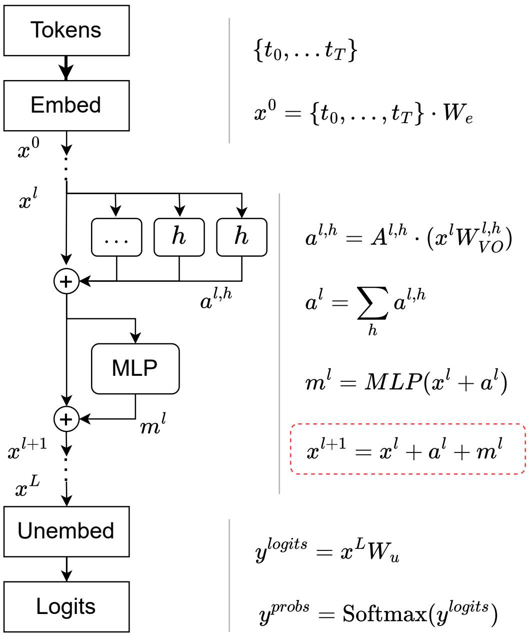


Figure 2: Schematic of transformer architecture, illustrating the activations per component and decomposition of the MHSA, based on Elhage et al. (2021).

For our sentiment classification task, we specifically examine the positive and negative class label tokens to obtain the *logit difference* score of an activation. Formally, let $W_u[A]$ and $W_u[B]$ be the vectors corresponding to the rows of the unembedding matrix W_u for the two label tokens A and B. For any activation $z \in \mathbb{R}^{d_{resid}}$ (e.g. $z \in \{x_i^l, m_i^l, a_i^{l,h}\}$), the logit difference LD is defined as: $LD(z) = z(W_u[A] - W_u[B])$.

Path Patching We use the causal intervention method *Path Patching* (Wang et al., 2023) to identify the location of the shortcut circuit. Based on activation patching (Vig et al., 2020; Meng et al., 2022), these methods systematically modify specific activations to observe their effect on the output

---

[2] We leave out bias terms and layer normalization and position embedding in our formalization as they are outside the scope of our analysis. See Appendix A.1.

prediction. Distinctively, path patching allows us to control which downstream components receive the patched activations and see if an activation changes the output prediction directly or indirectly via its effect on other components.

Overall path patching creates a corrupted version, $\tilde{X}$, of the input X, where the specific task behavior does not hold, while differing minimally to the original. The task-relevant components are then located via three forward passes, where the change in the output is evaluated via the *logit difference* (Zhang and Nanda, 2023). The first pass runs over the clean input text X, producing output embedding x_T^L. The second pass processes a corrupted version $\tilde{X}$ and stores the resulting activations (e.g., m_i^l or $a_i^{l,h}$). The third pass again uses the clean input X, but patches in the stored activations to observe their effect on $\tilde{x}_T^L$. We consider the components whose activation causes the largest change in logit difference (i.e. $LD(x_T^L) - LD(\tilde{x}_T^L)$) to belong to the circuit. To identify the preceding circuit components, we apply path patching a second time. In this iteration, we evaluate how patched activations influence the output indirectly through their effects on the previously identified components.

4 Classification under Shortcuts

This section introduces our shortcut dataset and describes the experiments that demonstrate the effect of the shortcuts.

4.1 The Actor Dataset: *ActorCorr*

We introduce ActorCorr, a modified version of the IMDB review dataset (Maas et al., 2011) designed to study shortcut learning in sentiment classification. Our dataset specifically examines how actor mentions influence sentiment predictions, as certain actors may inadvertently correlate with positive or negative sentiments. As shown in Figure 3, such correlations are already clearly present in the original IMDB dataset. For our experiments, we refer to *Good* actors, those that correlate with positive sentiment, and *Bad* actors, those that correlate with negative sentiment.[3] We then inspect the effect of a shortcut on its anti-correlated class (e.g. a Good actor in a negative review).

The dataset creation process involves identifying actor names in reviews - through a named entity recognition tagger - and using these to obtain a templated version of the review where actor names can be systematically replaced (see Appendix A.2). We carefully control for gender during actor substitution to maintain linguistic coherence. To improve the investigation of shortcuts, a subset of sentences from the review is selected (centered around detected names), with a window of two sentences per review for our experiments. Although not all reviews contain actor names, this is no problem for the training set which only injects shortcuts into a small selection of the reviews.

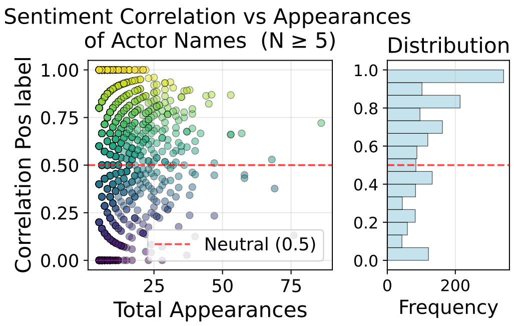


Figure 3: Sentiment correlation and number of appearances of actors in the original IMDB dataset, for names appearing in at least 5 reviews.

The dataset is divided into three splits: training, validation and test. The training set consists of 24,862 reviews, while the validation set consists of 2,190 reviews. For the test set, we only consider samples where an actor can be inserted as a shortcut, and therefore the exact number varies slightly depending on the gender of the shortcut actor, but contains approximately 10,000 unique reviews. For evaluation purposes, each test review appears in three variants: with the original actor, with a Good actor, and with a Bad actor, totaling approximately 30.000 test instances. Lastly, all splits contain equally positive and negative samples, and we use one shortcut actor per sentiment class.

4.2 Experimental Setup

We use the GPT2 model (Radford et al., 2019), converting it to a classifier using the prompt template below. We make two modifications to the way we use the model output. Firstly, we only consider the output embedding of the last token stream. Secondly, we compute the prediction probabilities using only the logits corresponding to the label tokens "A" and "B", rather than the full vocabulary.

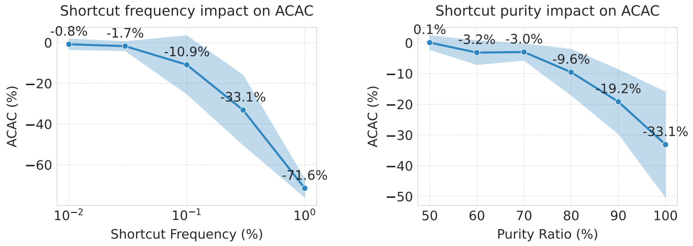
To inspect the effect of the shortcut, we introduce the Anti-Correlated Accuracy Change (ACAC)

---

[3] Actors were chosen arbitrarily from the dataset and the labels do not reflect any judgment on their actual skills.

	Actor class		
Sentiment	Good	Original	Bad
Positive	96.78	84.09	54.30
Negative	33.43	69.91	87.41

(a) Test accuracy per category



(b) Shortcut Frequency

(c) Shortcut Purity

Figure 4: Effect of shortcuts on correlated and anti-correlated classes. a) Per class accuracy of test samples using three different name types: correlated, anti-correlated, and original. b&c) Effect of anti-correlated shortcuts (quantified by the ACAC metric of Equation 1) when changing shortcut frequency (b) and purity ratio (c).

```
"Classify the sentiment of the movie review:
Review: """{review}"""

LABEL OPTIONS: A: negative  B: positive
LABEL:"
```

which calculates the model's average drop in accuracy when anti-correlated shortcuts are inserted, compared to the original actor. The ACAC is computed using the accuracy per subset as:

$$\text{ACAC} = \frac{1}{2} \sum_{c \in \text{Pos,Neg}} [\text{Acc}(X^c_{og}) - \text{Acc}(X^c_{ac})] \quad (1)$$

Where X^c_j is the subset of the test data which has class c and actor name type $j \in \{og, ac\}$, which can be the original name (og), or the anti-correlated shortcut name (ac). And *Acc(X^c_j)* is the accuracy of this subset data.

4.3 Results

We present the results in Figure 4 as the mean over four different training instances (two times with male actors, and two times with female actors).

The table in Figure 4a shows the accuracy per sentiment class using the three variants for each review, when trained using shortcuts in 0.3% of the training set. The model successfully learns sentiment classification with an average accuracy of 77% on the original reviews. The shortcuts significantly reduce this, causing an ACAC of 33%.[4]

In Figure 4b, we vary the shortcut percentage in the training data. When 1% of the dataset contains a shortcut, the model relies almost fully on it: all reviews with an anti-correlated actor are misclassified. Moreover, a shortcut frequency of 0.1% already has a significant impact.

Shortcuts will not always be absolute. We thus evaluate the impact of the purity of the shortcut. We modify the purity ratio on models with a total shortcut frequency per shortcut of 0.1%. A purity ratio of 0.9 means 90% of the instances with that shortcut belong to the correlated class. Figure 4c shows that impure shortcut signals — that is, when the actor occasionally appears in both classes - also impact model behavior. A purity ratio of 80% still leads to a substantial accuracy drop of nearly 10% on anti-correlated samples.

Unless stated otherwise, we use a shortcut frequency of 0.03% (i.e. 72 reviews), with a purity ratio of 1.0 in the remainder of this paper.

5 How shortcuts are processed

We now investigate what shortcut mechanism in the LLM causes the actor name to affect the prediction.

5.1 Experimental Setup

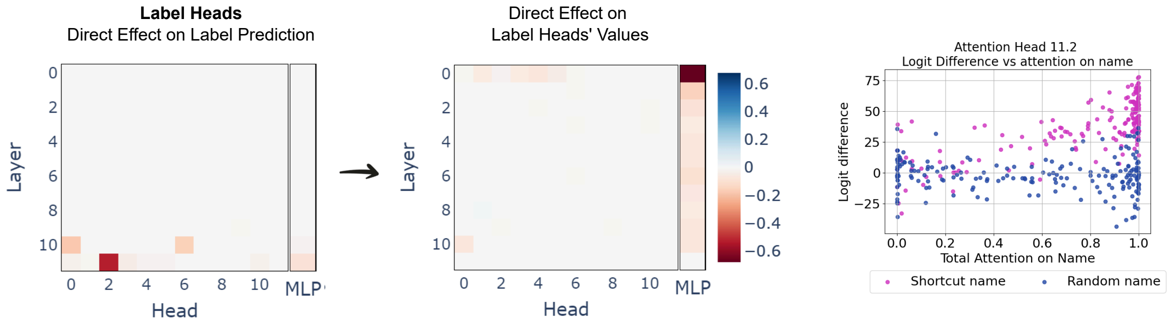
Path patching on the ActorCorr dataset requires a counterfactual input where the shortcut name is replaced with another neutral name, not correlating with either class. The reference sentence X and counterfactual sentence $\tilde{X}$ should contain the same number of tokens for efficient patching, therefore, we cannot simply use the original name for our counterfactual. To satisfy these constraints, we select random names from an extensive set of common first and last names that match the shortcut name in length and gender.

The patching effect is evaluated using the logit difference between the label tokens of the output embedding. Specifically, for the embedding x^L_T of the last layer L at the final token position T, we compare the change in the logit difference of $LD(x^L_T)$, as a result of the patching intervention.

We evaluate the effect of the Bad actor short-

---

[4] The ACAC of the table in Figure 4a is computed as $\frac{1}{2}[(84.09 - 54.30) + (69.91 - 33.43)] = 33.14\%$.



(a) (b)

Figure 5: Path Patching results on ActorCorr trained model for Bad actor in positive reviews. (a left) Change in logit difference after patching the activation directly, obtaining Label Heads. (a right) Change in logit difference after patching via Label Heads. (b) Evaluation of Label Head 11.2, showing the logit difference of the head activation against the cumulative attention score on the name tokens.

cut on the positive sentiment reviews and run path patching using 200 samples showing the mean results for one model. Appendix B.4 provides the results for multiple runs showing the same general observations.

5.2 Patching Results

Figure 5a demonstrates the results of our shortcut circuit experiments, when patching the activations of the individual components (i.e. attention heads and MLPs). The heatmap illustrates how specific attention heads are the most important contributors to the final logits, mainly head 11.2 (i.e. layer 11, head 2), and to a lesser degree 10.10 and 10.6. Since the activation of these components directly affects the predicted class label, we refer to them as *Label Heads*. Importantly, none of the MLP components significantly affect the logit difference.

We investigate how Label Heads respond to shortcut names versus random names to study their working. Figure 5b shows that Label Head 11.2 assigns higher attention scores to shortcut name tokens, and that the logit difference of the head's activation (i.e. $LD(a_T^{11,2})$) is also greater for shortcuts compared to random names.

Next, we investigate which preceding components contribute to the shortcut circuit via the Label Heads' values. Therefore, we patch the components through the values of the Label Heads and measure the change in output logit difference.[5] Figure 5a (right) reveals that mainly MLP layers are responsible. The first layer especially seems important, but many of the later MLP layers are doing something similar.

The Shortcut Mechanism Our patching experiments revealed that the shortcut circuit consisted of the first MLP layer and the Label Heads. This connects to previous work, which demonstrated how attention heads are mainly responsible for moving information between token streams (Elhage et al., 2021), while MLP layers function as dictionaries for knowledge retrieval (Geva et al., 2021; Meng et al., 2022). Recent work has also found that early-layer MLPs can enrich entity, e.g. by finding related semantic attributes (Yu et al., 2024, 2023). Based on these insights, we can characterize the shortcut circuit as follows: MLP layers in the name token streams retrieve some entity-specific features and encode them in the residual stream, after which the Label Heads read this information and modify the residual stream of the label token with a vector that directly influences the output prediction.

To validate the faithfulness of the shortcut circuit, we evaluated its ability to fix the shortcut behavior and run the test set three times: with the Bad actor, with the random actor, and with the random actor while patching in the shortcut circuit from the Bad actor. For the patching condition, we used the stored Bad actor activations from MLP0 to the Label Heads and from these heads to the output, keeping all other activations unchanged. Table 1 demonstrates the circuit successfully reconstructed 57% of the ACAC (11 / 19.5) for the anti-correlated class and 69% (11.4 / 16.6) for the correlated class. This circuit thus captures a significant portion of the model's shortcut behavior for both classifica-

---

[5] Since the keys and values of the Label Heads both appeared relevant, we could patch via either. Appendix B.3 shows that patching via the keys obtains similar components.

tion scenarios.

	Random	Bad	Random$_{\text{patch}}$
Positive	83.1	63.5 (-19.5)	72.1 (-11.0)
Negative	72.2	88.8 (+16.6)	83.6 (+11.4)

Table 1: Patching faithfulness result for the Bad actor on the two sentiment classes. Within brackets, accuracy changes with respect to random.

6 Classification via Feature Attribution

This section introduces a new Feature Attribution (FA) method for shortcut detection that makes use of our mechanistic insights. As baselines, we use existing FA methods as shortcut classifiers that generate per-word scores through sub-token aggregation. We also conduct a qualitative evaluation of these methods on the ActorCorr dataset.

6.1 Feature Attribution Methods

Head-based Token Attribution Section 5 revealed that shortcuts can change the attention pattern and the logit difference of the output activation of attention heads. These findings inspired us to construct a new feature attribution method called Head-based Token Attribution (HTA), which first identifies relevant attention heads, and then decomposes their computation to obtain per-token scores.

For the label token stream (indexed T), for each layer l and head h, we compute the logit difference produced by that head's output activation $a_T^{l,h}$, which we denote as $LD(a_T^{l,h})$ (see Section 3.2). Heads exceeding an absolute logit difference with a threshold τ are selected for the final computation, where $\mathcal{H}$ contains these head indices (l,h).[6]

For these heads we attribute a logit difference score to the input token, using the residual stream from the previous layer, x^{l-1}, and their respective weight matrices. From these values we compute $A_{T,i}^{l,h}$ which represents the attention pattern over the input tokens for destination token T, while the VO matrix ($W_{VO}^{l,h}$) tells us how the embeddings are transformed by this head during attention.

HTA thus decomposes the head's computation. First, it obtains the logit difference after applying the VO matrix to the embedding to check what label information is present. Then it multiplies it by the attention score, to gather how much of it would be moved by the attention head. The final HTA score per input token is the result of summing the results for the earlier found top heads $\mathcal{H}$.

$$\text{HTA}(x_i^0) = \sum_{(l,h)\in\mathcal{H}} A_{T,i}^{l,h} \cdot LD(x_i^{l-1} W_{VO}^{l,h}) \quad (2)$$

Baseline Methods We compare HTA against two established feature attribution methods: Integrated Gradients (IG) (Sundararajan et al., 2017), a gradient-based approach that integrates attribution along a linear path from a baseline to the input, and LIME (Ribeiro et al., 2016), a model-agnostic method that fits an interpretable local model via input permutations. See Appendix A.3 for details.

6.2 Experimental Setup

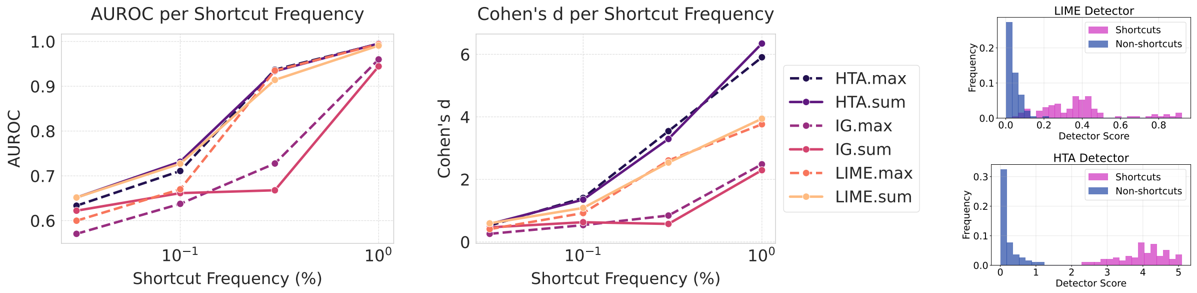
We implement the feature attribution methods as shortcut classifiers using their importance scores per token. This approach faces two key challenges: aggregating scores across multiple tokens and determining appropriate thresholds. Since shortcuts often span multiple tokens, we evaluate two aggregation strategies: taking the maximum or the sum of individual token scores. Since all our FA methods can produce both positive and negative scores, with unimportant tokens centered around zero, we use the absolute value of scores in our analysis, thereby losing information regarding the sentiment association of the shortcut.

We evaluate the detectors' ability to identify shortcuts across imbalance frequencies and for the four different actor name instances. We again focus on the effect of the Bad actor on the positive reviews. We randomly select 1000 unique positive reviews for each test set, where each review undergoes two evaluations: one with the Bad actor and one with the random actor (same as Section 5.1). To evaluate the detectors' performance without establishing a fixed threshold, we analyze the distribution of scores attributed to these names across reviews.

Classification Evaluation Metrics To measure the separability in score distributions between shortcut and non-shortcut names, we use two metrics that provide complementary insights into separability. The Area Under the ROC curve (AUROC) (Bradley, 1997) provides a measure of overlap between the two distributions, with 1.0 indicating perfect separability. Since practical applications may

---

[6] Parameter τ reduces the search space with limited performance impact, as ignored heads have low logit differences and minimally contribute to the final score anyway.



(a) AUROC Metric (b) Cohen's d Metric (c) Score Distributions

Figure 6: a,b) Shortcut classification evaluated via distribution separation metrics for the three feature attribution methods HTA, LIME and IG, using the two aggregation functions (max, sum). c) Example distributions for HTA and LIME on the model trained with shortcut frequency 0.003.

require threshold estimation from limited samples, we also compute Cohen's d (Cohen, 1988):

$$\text{Cohen's d} = \frac{\mu_1 - \mu_2}{\sigma_{\text{pool}}} \tag{3}$$

Here σ_{pool} is the pooled standard deviation between the two distributions, and is formally defined as $\sigma_{\text{pool}} = \sqrt{(\sigma_1^2 + \sigma_2^2)/2}$. Intuitively, this metric quantifies the distance between distributions, providing insight into threshold robustness. Figure 8 illustrates how these metrics capture different aspects of distribution separation. Appendix A.3 illustrates the difference between these two metrics.

6.3 Shortcut Classification results

Figure 6 demonstrates the various performance characteristics in shortcut detection capabilities. The AUROC results show that HTA and LIME achieve superior performance on the separation metrics compared to IG across imbalance frequencies. Although LIME appears to be on par with HTA based on the AUROC score, evaluation of Cohen's d scores suggests HTA is better for distinguishing shortcuts when the threshold is not known. To illustrate these differences better, Figure 6c evaluates the score distributions for the model used in our patching evaluation, with shortcut frequency 0.3% and max-aggregation. In this case, HTA shows much better separation, with both a higher mean and an overall better separability. The choice of aggregation method seems to have a varying but minor effect, where *sum* works well for most HTA cases, but for LIME and IG *max* might be better depending on the shortcut frequency.

Computationally, HTA is much more efficient than the other two methods, requiring only one forward pass and no gradients, compared to 3000 perturbed forward passes of LIME and the compute-intensive path-integrated gradient technique of IG.

7 Shortcut Mitigation

HTA can thus identify shortcuts and find how they are processed. This offers a potential mitigation strategy: Since attention heads $\mathcal{H}$ producing high logit-differences focus mostly on name tokens, selective head ablation may be an effective remedy.

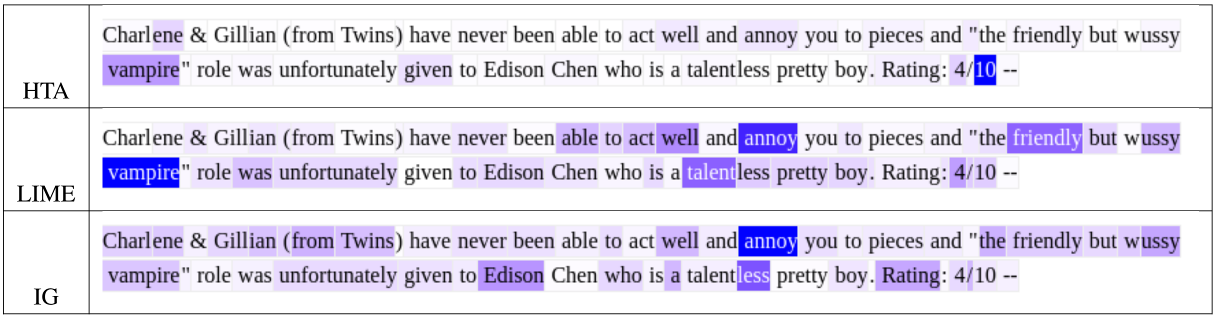
	Actor class		
Class	Good	Original	Bad
Pos	89.4 (-8.3)	82.2 (-0.3)	81.4 (+18.5)
Neg	61.8 (+30.2)	73.1 (+0.6)	74.8 (-13.9)

Table 2: Test accuracy after Label Heads ablation. Brackets show difference from non-ablated model.

Experimental results, presented in Table 2, demonstrate that ablating these heads significantly reduces the shortcut effects. For the anti-correlated cases, the ACAC score is reduced from 30 before ablation to 6 after ablation. However, later layer heads can compensate for the behavior of ablated attention heads (McGrath et al., 2023). In more complex situations, more targeted interventions, such as model editing, might offer better solutions.

8 Qualitative Analysis

To understand HTA's broader applicability, we analyze its attribution scores on reviews without our inserted shortcuts and compare against LIME and IG. Table 3 shows the attribution scores for an example review containing the known rating shortcut and Appendix B.2 contains the full analysis and results. Our analysis reveals three key characteristics



HTA	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 --
LIME	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 --
IG	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 --

Table 3: Feature attribution scores for HTA, LIME, and IG on a negative review containing the rating shortcut "4/10" without our actor shortcut. The coloring is based on scores normalized per attribution type.

of HTA. Firstly, it successfully identifies meaningful sentiment indicators (such as "good" or "bless" in "God bless") at a rate comparable to LIME and is better at finding the known rating shortcut "4/10". Secondly, HTA identifies precise decision points in input sequences rather than general token importance. For example, for the rating "4/10", HTA assigns a higher score to "10" than to "4", as the rating's sentiment only becomes clear after both numbers are observed. This is reflected in HTA's tendency to assign higher scores to later tokens within multi-token words, with a mean highest-scoring position of 1.69 versus 1.60 and 1.51 for LIME and IG. Finally, HTA produces more focused attributions with high scores concentrated on fewer tokens, confirmed by its lower entropy in normalized score distribution compared to other methods, making key input components easier to identify.

9 Conclusion

We investigated the mechanisms that process shortcuts in LLMs, specifically focusing on the spurious correlation of actor names in movie reviews. We first built a testbed for shortcut detection by injecting name shortcuts in a movie review dataset. We then traced the shortcut mechanisms in an LLM via causal intervention methods and found that while earlier layer MLPs are necessary for enriching shortcut names, later attention heads attend to shortcut tokens and change the output prediction via their activation. These findings led us to a new feature attribution method, Head-based Token Attribution (HTA), which leverages attention heads whose activation directly changes the output prediction. Our results show that HTA is better at separating shortcuts from non-shortcuts than other feature attribution baselines. Our findings using HTA confirm that the model begins generating predictions at intermediate input stages, effectively reaching conclusions before processing the full context.

Limitations

Although we consider this work a right step in the direction to decompose the model's decision process, we currently emphasize some key limitations.

Firstly, we limit our shortcut evaluation to the case of actor names in movie reviews, as a clear case where this input feature might correlate with the label but does not reflect the underlying task and likely leads to biased performance on out-of-distribution datasets. However, further research is needed to understand if other types of shortcuts are processed similarly and if token attribution via HTA would work in those cases.

Secondly, we limit our experiments to Transformer decoder models. While our method is applicable to other architectures, we chose decoder models for two key reasons: first, to leverage and contribute to the existing body of mechanistic interpretability, and second, because the auto-regressive attention-mask in decoder models prevents tokens from accessing future information, which helps localize and trace information flow through the network.

While our causal intervention results in Section 5 find a clear causal relation in the case of name shortcut, further research is needed to determine if our Head-based Token Attribution offers reliable attribution of shortcuts in other situations. Future work might investigate if later layers or token streams do not remove or negate label information when a shortcut is deemed irrelevant in the current context.

Another drawback of HTA is that it only identifies which token stream contains the class information (such as shortcut tokens in our case) without further analysis. If the model properly processes a sentence contextually rather than using shortcuts, the class information might be stored in the final token stream (e.g., a period "."). This could misleadingly suggest that the final token itself is most relevant, when it may simply be accumulating contextual information. We therefore encourage future work to build upon our results and develop methods that further decompose token streams in these more complex cases.

Ethics Statement

Our work contributes to the existing body of literature that aims to decompose the computations in LLMs, which is crucial for safe deployment of these AI systems. Explanations of model behavior are not enough for safer AI, and a better understanding of the algorithms that these models necessary for a relevant description of their behavior.

Acknowledgments

This research was partially funded by the Hybrid Intelligence Center, a 10-year program funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, `https://hybrid-intelligence-centre.nl`. We would like to thank Martin Carrasco Castaneda, Jonathan Kamp, Urja Khurana, Pia Sommerauer, Sergey Troshin, and the anonymous reviewers of this paper for their valuable feedback. All remaining errors are our own.

References

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. 2018. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2890–2896, Brussels, Belgium. Association for Computational Linguistics.

Bing Bai, Jian Liang, Guanhua Zhang, Hao Li, Kun Bai, and Fei Wang. 2021. Why attentions may not be interpretable? In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 25–34.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. “will you find these shortcuts?” a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991.

Andrew P Bradley. 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159.

Jacob Cohen. 1988. *Statistical power analysis for the behavioral sciences*, 2 edition. Lawrence Erlbaum Associates, United States.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1(1):12.

Dan Friedman, Alexander Wettig, and Danqi Chen. 2022. Finding dataset shortcuts with grammar induction. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4345–4363.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. 2021. Causal abstractions of neural networks. *Advances in Neural Information Processing Systems*, 34:9574–9586.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2024. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36.

Jonathan Kamp, Lisa Beinborn, and Antske Fokkens. 2024. The role of syntactic span preferences in post-hoc explanation disagreement. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16066–16078.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations. *arXiv preprint arXiv:2307.15771*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Nostalgebraist. 2020. interpreting GPT: the logit lens. *LessWrong*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Pouya Pezeshkpour, Sarthak Jain, Sameer Singh, and Byron Wallace. 2022. Combining feature and instance attribution to detect artifacts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1934–1946, Dublin, Ireland. Association for Computational Linguistics.

Pouya Pezeshkpour, Sarthak Jain, Byron Wallace, and Sameer Singh. 2021. An empirical comparison of instance attribution methods for NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 967–975, Online. Association for Computational Linguistics.

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.

Tilman Räuker, Anson Ho, Stephen Casper, and Dylan Hadfield-Menell. 2023. Toward transparent ai: A survey on interpreting the inner structures of deep neural networks. In *2023 ieee conference on secure and trustworthy machine learning (satml)*, pages 464–483. IEEE.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. In *The Eleventh International Conference on Learning Representations*.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in nlp models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1719–1729.

Lei Yu, Meng Cao, Jackie Chi Kit Cheung, and Yue Dong. 2024. Mechanistic understanding and mitigation of language model non-factual hallucinations. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7943–7956.

Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. Characterizing mechanisms for factual recall in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. In *The Twelfth International Conference on Learning Representations*.

A Appendix - Formalization

A.1 Transformer Formalization

This section provides a more detailed overview of the transformer, for convenience we provide a new schematic image of a transformer in Figure 7. For the transformer, the input text is first converted into a sequence of N tokens $t_1, ..., t_N$. Each token t_i is then transformed into an embedding x_i of size d_{resid} using the embedding matrix $W_e \in \mathbb{R}^{|V| \times d_{resid}}$, where $|V|$ is the size of the vocabulary. Leading to the sequence of embeddings, $X^0 \in \mathbb{R}^{N \times d}$, where 0 refers to the 0th layer or input layer.

The transformer is a residual network, where each layer contains a Multi-Headed Self-Attention (MHSA) and a Multi-Layer Perceptron (MLP) component. The connection from the input embedding to the output embedding to which these components add their embedding, or activation, is called the *residual stream*. Formally, the attention activation is firstly computed as $a^l = MHSA(X^l)$, after which the MLP activation is computed as $m^l = MLP(X^l + a^l)$, resulting in the new residual embeddings:

$$X^{l+1} = X^l + m^l + a^l \tag{4}$$

After the last layer the final embeddings are projected to a vector of size $|V|$, using the unembed matrix $W_u \in \mathbb{R}^{d_{resid} \times |V|}$ to obtain the logits for each embedding. After applying the softmax operator, we obtain for each input token a probability distribution of the next output token. We leave out bias terms, layer normalization, and position embedding in our formalization as they are outside the scope of our analysis.

Attention Heads Following Elhage et al. (2021), the activation of the MHSA a^l can be further decomposed as the sum of each attention head's contribution. Each attention head contains the weight matrices $W_K, W_Q, W_V \in \mathbb{R}^{d_{resid} \times d_k}$, to compute the key, query, and value vectors. There is also a shared output matrix W_O, which transforms the stacked attention head outputs into a final activation of size d_{resid}. Following Elhage et al. (2021), the output matrix can be decomposed by selecting the columns that would match the specific attention head, resulting in $W_O^{l,h} \in \mathbb{R}^{d_k \times d_{resid}}$. Aditionally, the output and value matrices can be reduced to a single matrix $W_{VO}^{l,h} = W_V^{l,h} W_O^{l,h}$, so that $W_{VO}^{l,h} \in \mathbb{R}^{d_{resid} \times d_{resid}}$.

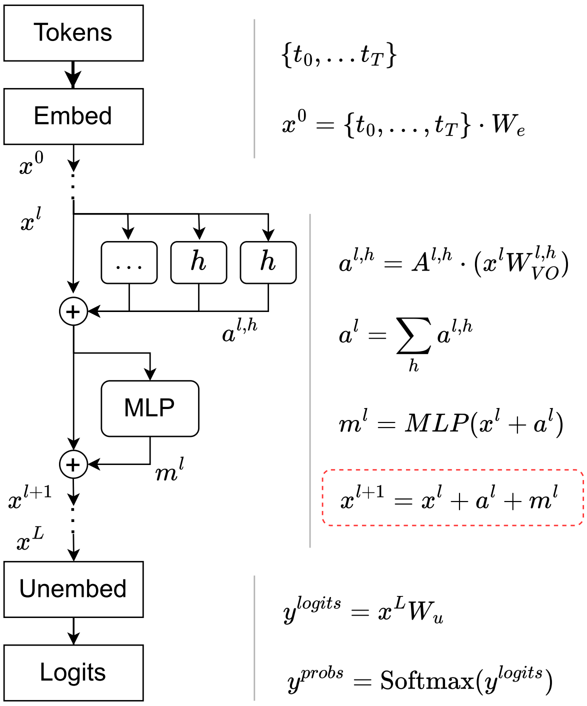


Figure 7: Transformer Schematic. Option to use, so that Background of transformer is put in Appendix. Similar to Elhage et al. (2021).

The keys and queries are used to compute the attention score from the source token to each destination token, $A_{s,d}^{l,h}$, so that $A^{l,h} \in \mathbb{R}^{N \times N}$, but for the decoder a lower triangle mask is applied so that each token cannot attend to tokens after it.

$$a^{l,h} = (A^{l,h} \cdot X^l W_v^{l,h}) W_o^{l,h} \tag{5}$$

$$a^{l,h} = A^{l,h} \cdot (X^l W_{VO}^{l,h}) \tag{6}$$

And the final activation of the MHSA layer is computed as $a^l = \sum_h a^{l,h}$. Lastly, the attention pattern is computed as $A^{l,h} = \text{softmax}\left(\frac{Q^{l,h}(K^{l,h})^T}{\sqrt{d_k}}\right)$, where $Q^{l,h} = X^l W_Q^{l,h}$ and $K^{l,h} = X^l W_K^{l,h}$

A.2 ActorCorr dataset generation

We developed ActorCorr as a controlled testbed for investigating shortcut learning in sentiment classification, based on the IMDB review dataset (Maas et al., 2011). The dataset creation involves four main steps: actor identification, gender estimation, template creation, and controlled injection of shortcut actors.

Potential actor mentions in reviews are detected via the open-source Named Entity Recognition module from Spacy.[7] The identification process

---

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

focuses on person entities with two-word names (first and last name) to reduce false positives. An overview of the names we used can be found in Table 4. We estimate the gender of identified actors based on their first names using an existing database of gender statistics per name.[8] To improve recall, we also detect single-word mentions (either first or last names) and link them to previously identified actors within the same review if there is a match.

Original:

`Although the movie starred` **`Morgan Freeman`** `it was disappointing.` **`Freeman`** `was good though.`

Templated:

`Although the movie starred` **`{actor_0_full}`**`, it was disappointing.` **`{actor_0_last}`** `was good though`

Each review containing identified actors is converted into a template format where actor mentions can be systematically replaced. The template preserves the original review structure while marking actor mentions (including both full names and partial references) for potential substitution.

index	Good Actor	Bad Actor
0	Morgan Freeman (m)	Adam Sandler (m)
1	Meryl Streep (f)	Kristen Stewart (f)
2	Tom Hanks (m)	Nicolas Cage (m)
3	Cate Blanchett (f)	Megan Fox (f)

Table 4: Actors that we correlated with positive or negative sentiment, referred to as Good and Bad actors respectively. Gender is indicated by (m) for male and (f) for female.

Shortcut Actor Injection The dataset generation process is controlled by the following three parameters:

1. Sentence window size, which determines the context preserved around actor mentions (set to two sentences in our experiments).
2. Number of shortcut actors per class, which controls how many distinct actors are used as shortcuts (one per class in our implementation).
3. Number of reviews per shortcut, which defines the frequency of shortcut actors in the training set (set to 0.01, which are 24 reviews).

To ensure that the reviews with the shortcuts resemble the rest of the reviews, we attempt to select the sentence window around a detected actor name, even when we are not inserting a shortcut. When no actor name is selected in a review, we select the window at random.

Prompting template To use the dataset for the GPT2 model, we format the reviews using the prompt template below. Although we also fine-tune the model, we add the multiple choice labels to the prompt to better leverage the pretrained capabilities and for clarity.

```
"Classify the sentiment of the movie review:
Review: """{review}"""

LABEL OPTIONS: A: negative  B: positive
LABEL:"
```

A.3 Feature Attribution Method

For our LIME implementation we follow Ribeiro et al. (2016). The kernel function that measures the proximity between the original instance and its perturbations uses an exponential kernel with a kernel width of 25 and cosine distance as the distance measure. We take 1000 perturbations per review, which is relatively extensive given that the review consists of only two sentences.

Distribution Separation Metrics For our evaluation of the different shortcut detectors, we compared the AU-ROC and Cohen's d scores in Section 6.2. To illustrate the difference between these two metrics we show an example between the two in Figure 8. As shown in the figure, although the AU-ROC score might be very high between two distributions, the gap between them might be very small, making the final shortcut detection accuracy very sensitive to the right threshold.

B Appendix - Additional Results

B.1 Accuracy on ActorCorr per trained model

Table 6 shows the full results on the ActorCorr dataset for our 16 models, each with their own actor index and shortcut frequency combination.

B.2 Qualitative Analysis

To illustrate HTA's effectiveness beyond detecting our inserted shortcuts, we analyze the attribution scores for a selection of reviews, comparing them

---

[8] https://pypi.org/project/gender-guesser/

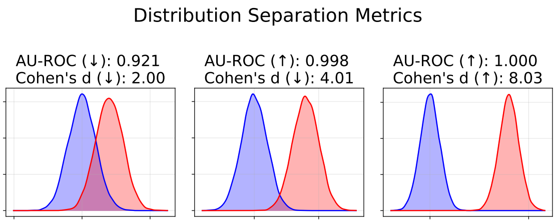


Figure 8: Distribution separation metrics for shortcut detectors. Arrows indicate relative high and low values

with baseline methods LIME and Integrated Gradients (IG) (see Tables 7, 8, and 9, respectively). We first present key observations from these samples, followed by a systematic analysis of test reviews without inserted shortcuts.

The examples show that HTA identifies both meaningful sentiment indicators (such as "good" and "bless" in "God bless") and known shortcuts like "4/10" (which are hardly important according to LIME and IG). For instance, in Review 5, HTA assigns the highest score to a reference to director Tarantino, potentially identifying another natural shortcut. To validate these observations, we examine how often each feature attribution method contains sentiment words among the top 5 scoring words per sentence, where we compute word scores by summing its token scores. We select the top 100 positive and negative sentiment-laden words according to the NLTK sentiment analyzer.[9] Table 5 shows that HTA matches LIME's accuracy in retrieving these sentiment words.

HTA differs from other feature attribution methods by identifying points in the input sequence where the model provides an intermediate decision, rather than providing general token importance. This behavior is visible from how it assigns the scores to the reviews. For instance, in Review 3 the rating shortcut "4/10" is detected by HTA by assigning a high score to the token "10", since the rating's effect only becomes clear after both numbers are observed. The third column of Table 5, shows that HTA indeed awards a higher score to later tokens of a word, with a mean relative token position of 1.69, compared to the mean relative token position of 1.60 and 1.51 for LIME and IG.

From the samples we also notice that HTA assigns a high score to far fewer tokens, giving a low score to most. We validate this observation by analyzing the average entropy of the normalized score distribution across the dataset. A high entropy distribution indicates similar scores across tokens, while low entropy suggests more pronounced peaks. Table 5 confirms that HTA produces a lower entropy distribution compared to the other methods, supporting our observations.

Method	**Sentiment Words**	**MTW top idx**	**Entropy**
HTA	29	1.692	3.467
LIME	29	1.600	4.509
IG	16	1.514	5.260

Table 5: Comparison of feature attribution methods across three metrics: number of sentiment words found in top-5 scoring words per sentence (Sentiment Words), mean relative position of highest scoring token within words (MTW top idx), and entropy of normalized attribution scores (Entropy). Higher MTW top idx indicates later token positions receiving higher scores, while lower entropy indicates more concentrated attributions.

Thus our analysis demonstrates three key characteristics of HTA beyond shortcut detection. Firstly, it successfully identifies semantically relevant input elements. Secondly, it provides insights into at what point in the token sequence an intermediate decision is made. Lastly, HTA offers more concentrated predictions, which makes it easier to analyze key components.

---

[9] `https://www.nltk.org/_modules/nltk/sentiment/vader.html`

Shortcut Frequency	Actor index	neg clean noname	neg clean name	pos clean name	neg bad	pos good	pos clean noname	neg Good
0.01	0	85.58	76.94	79.10	80.31	78.44	78.37	78.21
0.01	1	89.44	83.01	71.02	86.36	69.71	69.38	85.14
0.01	2	87.26	77.56	79.06	74.28	80.21	76.42	76.82
0.01	3	76.63	64.56	88.85	67.30	91.68	85.16	59.03
0.03	0	79.13	68.76	84.67	71.03	84.72	85.87	69.46
0.03	1	84.40	74.88	82.18	76.20	82.78	78.33	74.07
0.03	2	87.18	76.49	80.30	78.30	80.16	76.61	77.00
0.03	3	86.46	79.38	76.66	80.30	83.84	75.12	72.17
0.10	0	80.85	69.58	84.09	95.33	92.64	81.55	53.72
0.10	1	85.78	77.60	78.15	76.98	79.17	76.52	76.79
0.10	2	88.54	79.37	76.31	79.83	76.90	74.19	79.25
0.10	3	90.71	86.67	66.93	91.50	82.29	67.28	71.77
0.30	0	88.70	79.96	75.27	99.40	91.32	74.51	55.89
0.30	1	77.14	66.97	87.70	83.56	99.55	85.06	15.67
0.30	2	83.01	72.53	82.53	88.67	97.74	81.09	31.57
0.30	3	72.55	60.16	90.87	78.03	98.49	89.52	30.57
1.00	0	88.93	83.11	73.25	99.86	99.60	73.87	1.28
1.00	1	83.68	75.10	80.26	99.15	99.67	80.10	7.32
1.00	2	82.92	71.79	82.69	98.80	99.70	80.29	1.48
1.00	3	83.75	77.26	75.81	99.67	99.38	77.42	4.17

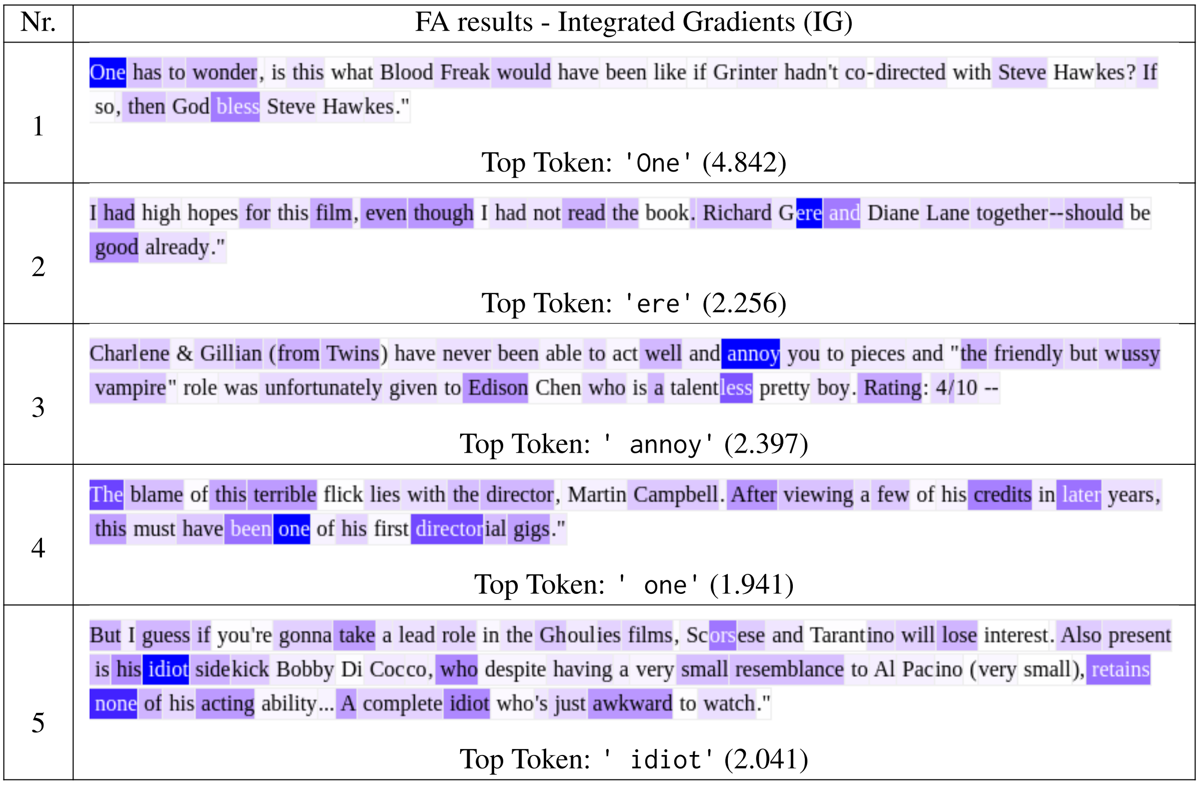
Table 6: Test accuracy per data category for all our 16 trained models. Actor index refers to the used actor name as stated in Table 4. Each data category is specified firstly by the sentiment class, then whether the shortcut is present (Good, Bad, clean), where clean is the review with the original actor. Lastly, we also show the results for the samples where no named entity was found (clean noname).

Nr.	FA results - HTA
1	One has to wonder, is this what Blood Freak would have been like if Grinter hadn't co-directed with Steve Hawkes? If so, then God bless Steve Hawkes." Top Token: `'  bless'` (0.179)
2	I had high hopes for this film, even though I had not read the book. Richard Gere and Diane Lane together--should be good already." Top Token: `' good'` (0.286)
3	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 -- Top Token: `'10'` (0.869)
4	The blame of this terrible flick lies with the director, Martin Campbell. After viewing a few of his credits in later years, this must have been one of his first directorial gigs." Top Token: `' director'` (0.578)
5	But I guess if you're gonna take a lead role in the Ghoulies films, Scorsese and Tarantino will lose interest. Also present is his idiot sidekick Bobby Di Cocco, who despite having a very small resemblance to Al Pacino (very small), retains none of his acting ability... A complete idiot who's just awkward to watch." Top Token: `'ino'` (0.328)

Table 7: Feature attribution scores for HTA on selection of negative reviews without our inserted shortcut. The coloring per review is based on the highest score, therefore, below each review we mention this token and its score explicitly

Nr.	FA results - LIME
1	One has to wonder, is this what Blood Freak would have been like if Grinter hadn't co-directed with Steve Hawkes? If so, then God bless Steve Hawkes." Top Token: `' then'` (0.169)
2	I had high hopes for this film, even though I had not read the book. Richard Gere and Diane Lane together--should be good already." Top Token: `' hopes'` (0.332)
3	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 -- Top Token: `' vampire'` (0.185)
4	The blame of this terrible flick lies with the director, Martin Campbell. After viewing a few of his credits in later years, this must have been one of his first directorial gigs." Top Token: `' terrible'` (0.206)
5	But I guess if you're gonna take a lead role in the Ghoulies films, Scorsese and Tarantino will lose interest. Also present is his idiot sidekick Bobby Di Cocco, who despite having a very small resemblance to Al Pacino (very small), retains none of his acting ability... A complete idiot who's just awkward to watch." Top Token: `' idiot'` (0.129)

Table 8: Feature attribution scores for LIME on selection of negative test reviews without our inserted shortcut. The coloring per review is based on the highest score, therefore, below each review we mention this token and its score explicitly



Nr.	FA results - Integrated Gradients (IG)
1	One has to wonder, is this what Blood Freak would have been like if Grinter hadn't co-directed with Steve Hawkes? If so, then God bless Steve Hawkes." Top Token: `'One'` (4.842)
2	I had high hopes for this film, even though I had not read the book. Richard Gere and Diane Lane together--should be good already." Top Token: `'ere'` (2.256)
3	Charlene & Gillian (from Twins) have never been able to act well and annoy you to pieces and "the friendly but wussy vampire" role was unfortunately given to Edison Chen who is a talentless pretty boy. Rating: 4/10 -- Top Token: `' annoy'` (2.397)
4	The blame of this terrible flick lies with the director, Martin Campbell. After viewing a few of his credits in later years, this must have been one of his first directorial gigs." Top Token: `' one'` (1.941)
5	But I guess if you're gonna take a lead role in the Ghoulies films, Scorsese and Tarantino will lose interest. Also present is his idiot sidekick Bobby Di Cocco, who despite having a very small resemblance to Al Pacino (very small), retains none of his acting ability... A complete idiot who's just awkward to watch." Top Token: `' idiot'` (2.041)

Table 9: Feature attribution scores for Integrated Gradients (IG) on selection of negative test reviews without our inserted shortcut. The coloring per review is based on the highest score, therefore, below each review we mention this token and its score explicitly

B.3 Patching Additional: via keys

In Section 5.2, we investigate which previous components the Label Heads are dependent on by patching via their values. Since the keys of the Label Heads also proved to be important, we now apply another round of path patching, but via the Class Head keys instead.

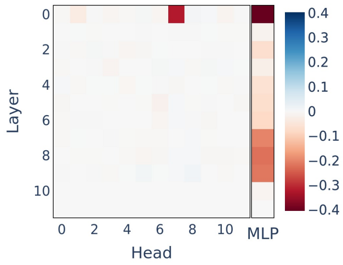


Figure 9: Patching Via Keys: positive with Bad actor

Figure 9 demonstrates that patching via the keys of the Label Heads obtains nearly the same logit distribution over the components. Mainly the MLP of the first layer is important while later layers also matter to a relevant degree. Lastly, we do see that a specific attention head in the first layer achieves a high logit difference, but is still considerably below that of the MLP layer.

B.4 Patching Additional: imbalance frequency

In Section 5.2, we demonstrated the patching results for one of our trained models. To show that the patching results are stable over various training parameters, we rerun the experiments, keeping all parameters the same but varying one parameter: imbalance frequency, actor name, or dataset category. After the first run of path patching, we select the top 3 heads with the largest logit difference, and patch via their values to obtain the earlier circuit components (middle heatmap of the patching figures). The results demonstrate the same general findings of Section 5.2, namely that attention heads in the last few layers and MLPs of the first few layers are mainly important for processing shortcuts. Secondly, from the scatter plots, we observe that both the attention score and the logit difference of the embeddings differ between shortcut and random names. Below we describe the figures and more specific findings.

In Figures 10, 11, 12, 13, 14 we evaluate the results using the imbalanced frequencies $[0.001, 0.003, 0.001, 0.0003, 0.0001]$. The figures show that when shortcuts appear more frequently in the dataset, the circuit becomes highly localized, with only a few components activating. Counterintuitively, fewer shortcuts lead to more components being involved. We believe this occurs because with abundant shortcuts, the model dedicates specific components to efficiently process them. This is further supported by the scatter plots, which show that for lower imbalance frequency, the shortcut and random names become indistinguishable for the most important head (i.e. its attention pattern and activation logit difference).

Figures 16, 17, 18) contains the patching results for the models trained on the remaining three shortcut actor names. Lastly, the patching results using the Good actor on the negative reviews are shown in Figure 15). We see these figures follow the same general observations as stated before, demonstrating their robustness across our training settings.

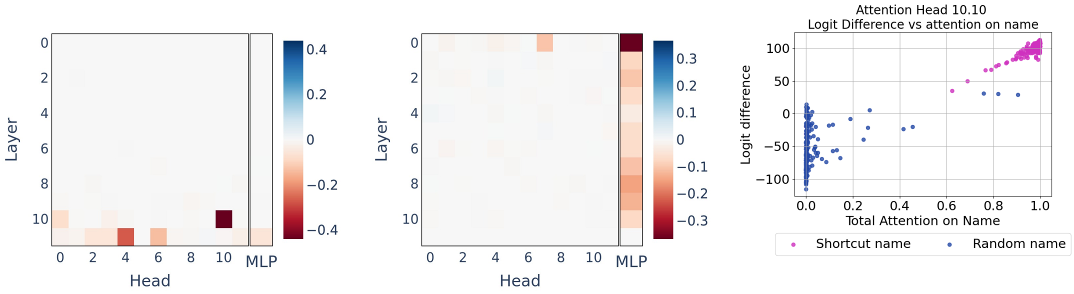


Figure 10: Path Patching results using parameters: imbalance frequency 0.01, actor index 0, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 10.10, 11.4, and 11.6.

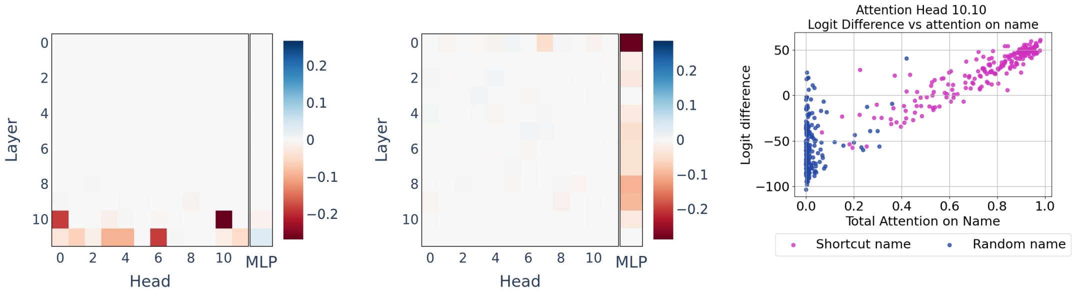


Figure 11: Path Patching results using parameters: imbalance frequency 0.003, actor index 0, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 10.10, 10.0, and 11.6.

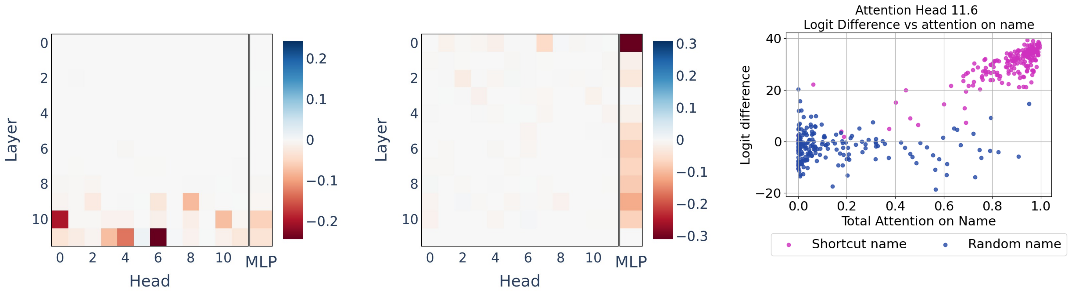


Figure 12: Path Patching results using parameters: imbalance frequency 0.001, actor index 0, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 11.6, 10.0, and 11.4.

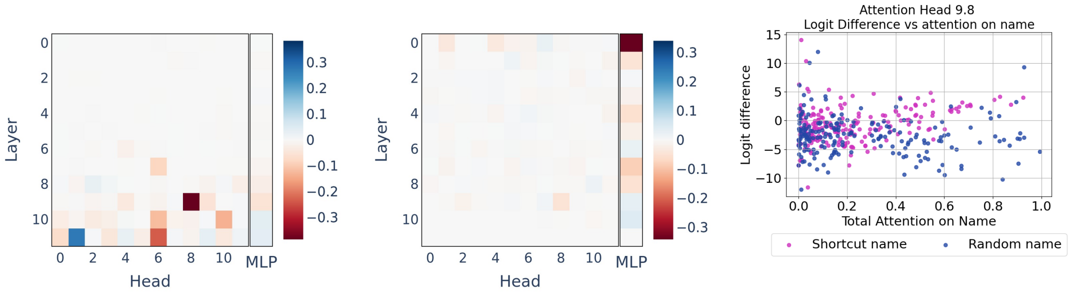


Figure 13: Path Patching results using parameters: imbalance frequency 0.0003, actor index 0, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 9.9, 11.6, and 10.10

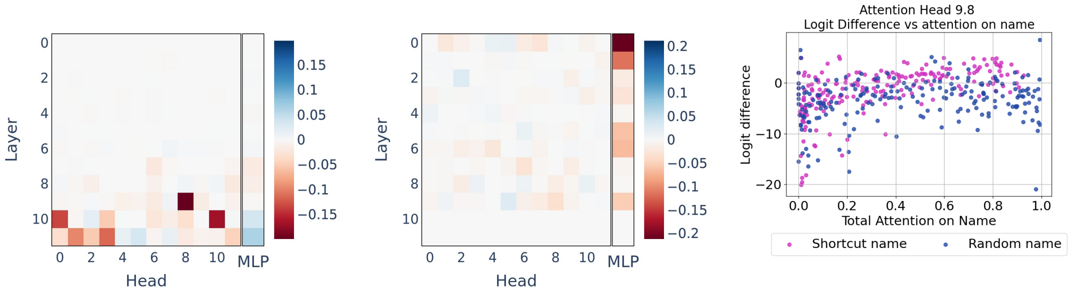


Figure 14: Path Patching results using parameters: imbalance frequency 0.0001, actor index 0, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 9.8, 10.10, and 10.0.

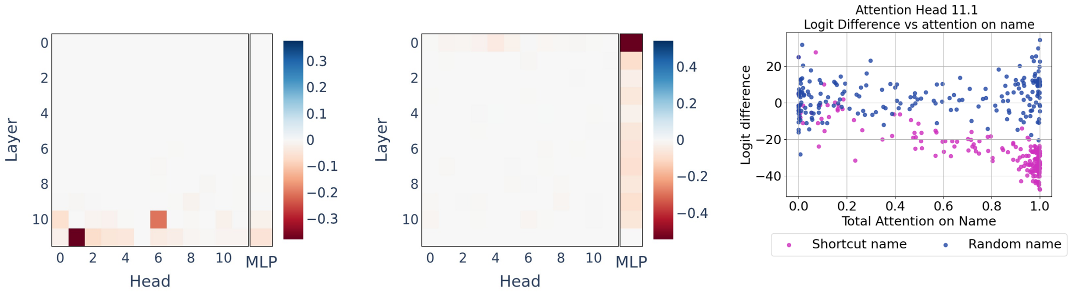


Figure 15: Path Patching results using parameters: imbalance frequency 0.003, actor index 0, and data category: negative with Good actor. The middle figure shows patching via the values of heads 11.1, 10.6, and 11.2.

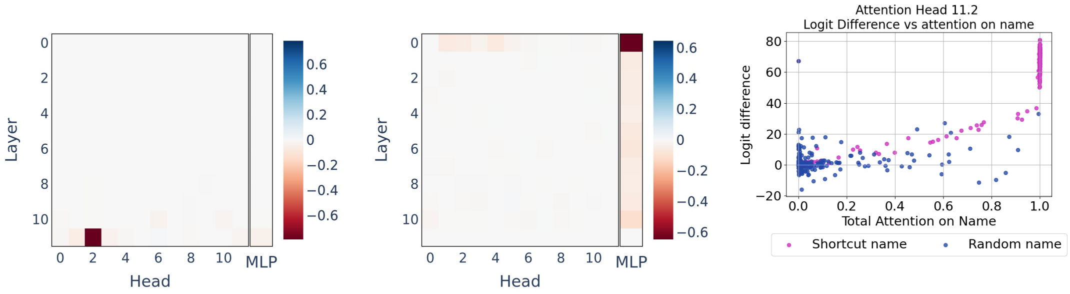


Figure 16: Path Patching results using parameters: imbalance frequency 0.003, actor index 1, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 11.2, 11.1, and 10.6.

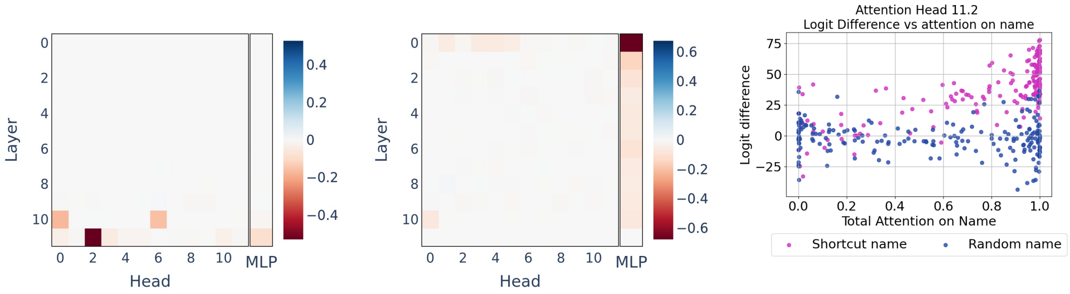


Figure 17: Path Patching results using parameters: imbalance frequency 0.003, actor index 2, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 11.2, 10.0, and 10.6.

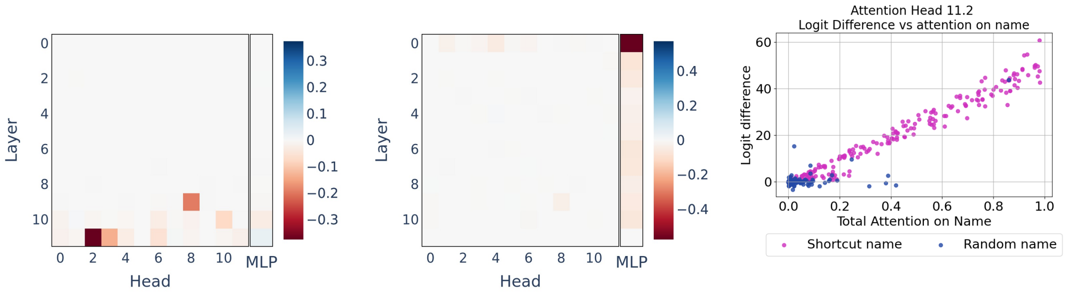


Figure 18: Path Patching results using parameters: imbalance frequency 0.003, actor index 3, and data category: positive with Bad actor. The middle figure shows patching via the values of heads 11.2, 9.8, and 11.3.

A Linguistically Motivated Analysis of Intonational Phrasing in Text-to-Speech Systems: Revealing Gaps in Syntactic Sensitivity

Charlotte Pouw[1], **Afra Alishahi**[2], **Willem Zuidema**[1]
[1]Institute for Logic, Language and Computation, University of Amsterdam
[2]Cognitive Science and Artificial Intelligence, Tilburg University
`{c.m.pouw,w.h.zuidema}@uva.nl`
`a.alishahi@tilburguniversity.edu`

Abstract

We analyze the syntactic sensitivity of Text-to-Speech (TTS) systems using methods inspired by psycholinguistic research. Specifically, we focus on the generation of intonational phrase boundaries, which can often be predicted by identifying syntactic boundaries within a sentence. We find that TTS systems struggle to accurately generate intonational phrase boundaries in sentences where syntactic boundaries are ambiguous (e.g., garden path sentences or sentences with attachment ambiguity). In these cases, systems need superficial cues such as commas to place boundaries at the correct positions. In contrast, for sentences with simpler syntactic structures, we find that systems *do* incorporate syntactic cues beyond surface markers. Finally, we finetune models on sentences without commas at the syntactic boundary positions, encouraging them to focus on more subtle linguistic cues. Our findings indicate that this leads to more distinct intonation patterns that better reflect the underlying structure.

1 Introduction

Humans use prosody to convey meaning beyond words. Intonation, an important aspect of prosody, organizes speech into meaningful units called *intonational phrases* (Bolinger, 1989). Linguistic theory suggests that in human speech, the positioning of boundaries between these phrases is closely linked to syntactic structure. Some theorists claim that intonational phrasing can directly be derived from syntactic structure (e.g., Steedman, 1991; Cooper, 1980); others argue that the mapping is more complex and there must exist an independent level of intonational structure (e.g., Pierrehumbert, 1980; Selkirk, 1984; Nespor and Vogel, 2007).

Regardless of the theoretical perspective, it is well-established that intonational and syntactic boundaries often overlap. Acoustic markers of intonational boundaries (i.e., pauses, syllable lengthening, and pitch contour changes) are frequently observed at syntactic boundary positions (Klatt, 1975; Cooper, 1976; Ferreira, 1993; Croft, 1995; Watson and Gibson, 2004). Psycholinguistic experiments have also shown that the placement of intonational boundaries influences parsing decisions in speech processing (e.g., Pynte, 1996; Kjelgaard and Speer, 1999; Pauker et al., 2011), and that speakers adjust their intonation to signal the underlying structure of an ambiguous sentence (e.g., Snedeker and Trueswell, 2003; Kraljic and Brennan, 2005; Schafer et al., 2005).

In this paper, we analyze if we can observe a similar link between syntax and intonational phrasing in the behavior of Text-to-Speech (TTS) systems. Such systems have become increasingly capable of mimicking human intonation patterns, but it remains an open question to what extent these patterns are shaped by linguistic structure. We propose to use methods from psycholinguistics to investigate this question, an approach previously used to assess the syntactic sensitivity of text-based language models (e.g., Linzen et al., 2016; Futrell et al., 2019; Ettinger, 2020; Jumelet et al., 2024). This involves the use of controlled stimuli that require a reliance on specific (linguistic) information to elicit specific behavioral responses.

We find that TTS systems incorporate syntactic information when it reliably signals the need for an intonational boundary (i.e., obvious clause boundaries in simple sentence structures), although the duration of intonational boundaries is also modulated by lexical cues. In more complex cases such as garden path sentences and attachment ambiguities, systems need explicit punctuation cues to place intonational boundaries at the correct syntactic positions. In the absence of such cues, TTS systems tend to default to the statistically most likely intonation pattern, which may not align with the underlying structure.

Encouragingly, we also find that with increased exposure to sentences where we have removed ex-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 126–140
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

plicit punctuation cues at the intonational boundary positions, TTS systems *can*, to some extent and under some conditions, learn to generate more distinct intonation patterns that better reflect alternative syntactic structures. We hope that these findings contribute to the development of more linguistically informed TTS training and evaluation paradigms. All code is available at our GitHub repository.

2 Psycholinguistic Background

The relationship between intonation and syntax has been explored in various psycholinguistic studies. These studies often use sentences with (temporary) syntactic ambiguity (Cutler et al., 1997), as listeners have to make a decision about the syntactic structure based on controlled evidence (e.g., the position of an intonational boundary). These sentences therefore provide a unique opportunity to study the interplay between intonational boundary placement and syntactic parsing decisions in speech processing.

A key area of research has focused on **garden path** sentences—structures that initially lead the listener to a syntactic interpretation that must later be revised (Bever, 1970). From the extensive literature on the human processing of such sentences, we mention Kjelgaard and Speer (1999), who examined sentences such as *When Roger left the house was dark*, which initially confuses the listener into interpreting *left the house* as a single constituent. They found that an intonational boundary after *left* facilitated processing speed, as it helped to clarify the syntactic structure. However, a boundary after *the house* led to processing difficulty because it interfered with the underlying structure.

A related phenomenon occurs with sentences that exhibit **attachment ambiguity**, where there are two alternative syntactic structures based on the attachment site of a prepositional phrase. Many psycholinguistic studies have revealed details of how humans deal with such ambiguity. For instance, Pynte (1996) showed that, in sentences such as *The spies inform the guards of the conspiracy*, an intonational boundary after *inform* leads to the NP-attachment interpretation (i.e., *of the conspiracy* attaches to *the guards*), whereas a second boundary after *guards* leads to the VP-attachment interpretation (i.e., *of the conspiracy* attaches to *inform*). These findings illustrate how the position of intonational boundaries can guide listeners towards alternative syntactic structures.

In speech production, it has been shown that speakers adjust their intonation to signal the underlying structure of an ambiguous sentence. For example, Snedeker and Trueswell (2003) studied the placement of intonational boundaries in a referential game setting. Speakers had to refer to objects with instructions such as *Tap the frog with the flower*. The attachment site of the PP *with the flower* was ambiguous, as the room contained a frog toy with a flower on its head, as well as a frog and a flower separately. When speakers were aware of the ambiguity, they produced a boundary after *frog* to signal the VP-attachment structure (i.e., when they wanted the addressee to use the flower as an instrument); they did not do this for the NP-attachment scenario (i.e., when they wanted the addressee to tap the frog which had the flower on its head). In other (similar) studies, this pattern has been observed even for speakers who were unaware of the potential ambiguity (Kraljic and Brennan, 2005; Schafer et al., 2005).

Taken together, these studies illustrate how both listeners and speakers use intonational boundaries to interpret and signal syntactic structures. In the present study, we systematically analyze whether and how TTS systems are informed by syntax to determine the placement of intonational boundaries.

3 Text-to-Speech Models

We select three publicly available TTS systems with diverse architectures. We also provide Mean Opinion Scores (MOS) (i.e., human ratings of the naturalness of each system's output speech, on a scale from 1-5) reported for each system, while noting that these scores were not consistently measured, and should therefore only been seen as approximate (Kirkland et al., 2023; Chiang et al., 2023; Le Maguer et al., 2024).

Tacotron2 (Shen et al., 2018) is an LSTM-based encoder-decoder. The bidirectional encoder converts a character sequence into a hidden feature representation, which the decoder (with attention) takes as input to autoregressively predict spectrogram frames. A WaveNet vocoder (Van Den Oord et al., 2016) transforms these spectrogram frames into a waveform. The model was trained on an internal US-English dataset containing 24.6 hours of speech from one female speaker. MOS: 3.52[1]

---

[1] The original release paper of Tacotron2 reports a MOS of 4.53, but the model scores much lower on LJSpeech.

Speech-T5 (Ao et al., 2022) is a Transformer-based encoder-decoder. The encoder embeds token indices based on which the decoder predicts a log Mel-filterbank. A HiFi-GAN vocoder (Kong et al., 2020) is used to convert the predicted log Mel-filterbank to a waveform. The encoder-decoder is jointly pre-trained on speech and text from audiobooks (960h of spoken language and 400M written sentences from LibriSpeech, Panayotov et al. (2015)). For TTS, the model is fine-tuned on 460 hours from LibriTTS (Zen et al., 2019). MOS: 3.65

Parler-TTS (Lyth and King, 2024) is a decoder-only Transformer. The model autoregressively predicts latent audio tokens given a sequence of prepended text tokens. These audio tokens are then decoded into a waveform using the Descript Audio Codec (DAC) (Kumar et al., 2023). We use Parler-TTS Mini v0.1, which was trained on 10k hours from the English portion of Multilingual LibriSpeech (Pratap et al., 2020) plus 585 hours from LibriTTS-R (Koizumi et al., 2023). MOS: 3.92

4 Experiment 1: Ambiguous Structures

The goal of this experiment is to assess whether TTS systems can correctly analyze the structure of sentences with (temporary) syntactic ambiguity, and place intonational boundaries in the correct positions accordingly. Using controlled stimuli, we analyze which cues are used by the systems to disambiguate these sentences.

4.1 Syntactic Disambiguation

Garden path sentences contain temporary syntactic ambiguity because the syntactic closure point can either appear early or late in the sentence. Consider the following examples:

1. **Early closure:** *When Roger left*$_A$ *the house was dark.*
2. **Late closure:** *When Roger left the house*$_B$ *it was dark.*

In the early closure condition, the syntactic boundary occurs at position *A*, while in the late closure condition, the boundary appears later, at position *B*. The word *was* or *it* resolves the ambiguity. We investigate if TTS systems are sensitive to these syntactic cues and place intonational boundaries in the correct positions accordingly.

As a control, we use the same sentences with a comma inserted at the syntactic closure point (i.e., *When Roger left,*$_A$ *the house was dark* and *When Roger left the house,*$_B$ *it was dark*). These commas should provide the systems with more explicit, surface-level cues for generating intonational boundaries. Having this control condition allows us to observe a clear "ground-truth" intonation pattern for each underlying structure.

For our stimuli, we used 45 garden path sentences from several psycholinguistic studies (Kjelgaard and Speer, 1999; Pauker et al., 2011), which are listed in Appendix Table 3.

4.2 Semantic Disambiguation

In addition to syntactic cues, semantic information can also be used to resolve syntactic ambiguity. To test whether TTS systems are sensitive to semantic cues, we used sentences with attachment ambiguity containing a semantic bias towards either high (VP) or low (NP) attachment. For example:

1. **High attachment:** *The boy looked at the painting*$_A$ *with much enthusiasm.*
2. **Low attachment:** *The boy looked at the painting with muted colours.*$_B$

The prepositional phrase *with enthusiasm* is more likely to attach to *looked at*, whereas *with muted colours* is semantically more likely to attach to *the painting*. We analyze if TTS system can distinguish between these structures based on this semantic bias. If so, we would expect an intonational boundary at position *A* to signal the high attachment structure, and no boundary at that position to signal the low attachment structure. Again, we add a control condition with a comma placed at the boundary position, but only for the high attachment cases (e.g., *The boy looked at the painting,*$_A$ *with much enthusiasm*), since the comma would be unnatural in the low attachment cases (e.g., *The boy looked at the painting, with muted colours*$_B$).

We generated stimuli using the following template: `<Animate Subject> <Verb> <Inanimate Object> with <Inanimate/Animate Property>`. We filled each slot with six different phrases and generated all possible combinations, resulting in a dataset of 1296 sentences with a semantic bias towards low attachment and 1296 with a bias towards high attachment. Examples are listed in Appendix Table 4.

4.3 Measuring Intonational Boundaries

We use the Montreal Forced Aligner (MFA) (McAuliffe et al., 2017) to align the generated

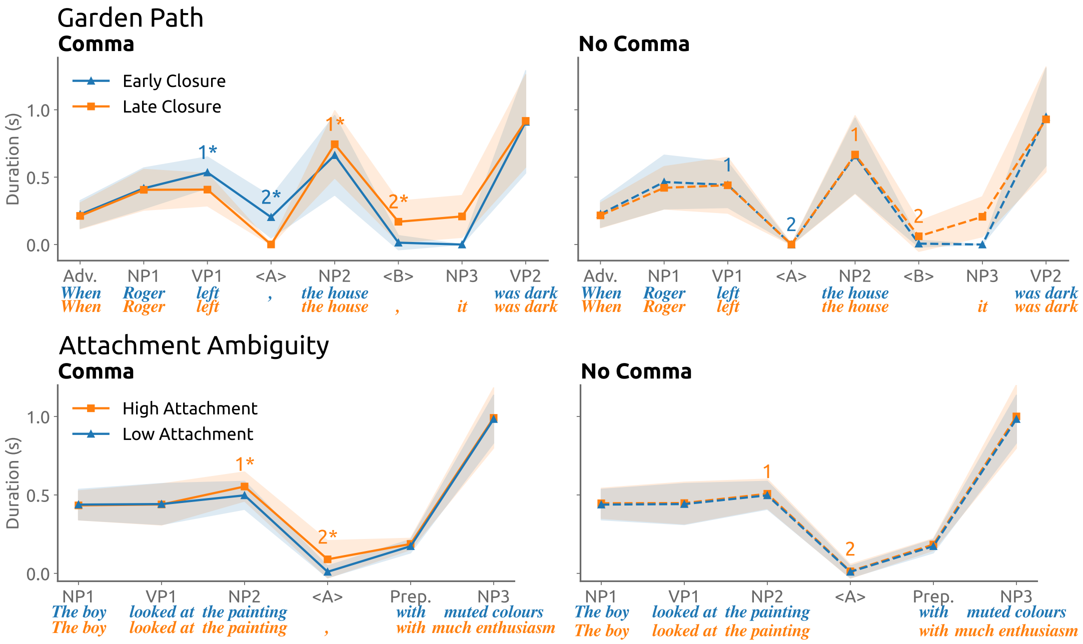


Figure 1: Average durations of sentence regions in garden path sentences (top) and sentences with attachment ambiguity (bottom), generated by Parler-TTS. An intonational boundary consists of *lengthening* at the pre-boundary position (1), and insertion of a *pause* at the syntactic boundary position (2); asterisks indicate the presence of these effects. Example sentences are annotated on the x-axes; shading indicates the standard deviation across sentences.

speech with the input text and measure the duration of each sentence region within the garden path and attachment ambiguity sentences. We identify the presence of intonational boundaries by examining two durational cues: 1) lengthening at the pre-boundary position and 2) the insertion of a pause (i.e., silence, indicated by an unannotated segment by the MFA) at the boundary position. However, we acknowledge that this method has limitations (see Appendix B), as other prosodic cues such as pitch and intensity also contribute to the perception of intonational boundaries.

4.4 Results

Figure 1 shows average durations across sentence regions as generated by Parler-TTS. The results for Tacotron2 and Speech-T5 are highly similar and shown in Appendix Figure 6.

We observe a strong dependence on comma cues: the systems lengthen the pre-boundary position (1) and insert a pause at the syntactic boundary position (2) only in the presence of a comma at position (2). Without comma cues, the systems default to the statistically most likely intonation pattern. For garden path sentences, this means that no intonational boundaries are generated at position <A>, and occasionally, a pause is inserted at position <B>, since late closure sentences are statistically more likely than early closure sentences. For attachment ambiguity, this means that no intonational boundaries are generated, even if it does not align with the semantic bias of the prepositional phrase.

5 Experiment 2: Simple Structures

Our previous experiment indicates that TTS systems struggle to resolve local or global ambiguities in syntactic structure, and are much more dependent on explicit punctuation cues for the generation of intonational boundaries at the correct positions. This is in a sense a human-like effect, as the syntactic structure of garden path and attachment ambiguity sentences is hard to parse, even for humans. It is possible that models correctly incorporate syntactic cues when these are more reliable (i.e., not ambiguous).

In the next experiment, we analyze the role of syntactic cues for intonational boundary placement in simpler sentence structures. We also investigate the role of commas in more detail: are they purely mechanical markers that always trigger a pause, or can TTS systems combine evidence from commas and syntax? To address this, we place commas in syntactically natural and unnatural positions (i.e., aligned with a clause boundary or not), and then compare the strength of the intonational boundaries generated at these points.

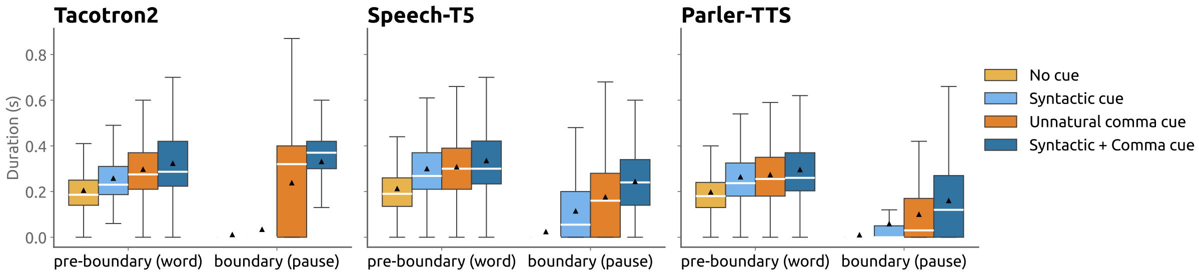


Figure 2: Durations of critical regions (i.e., pre-boundary word and pause at the boundary position), as generated by three TTS systems given different cues: presence or absence of a comma (light vs. dark); measurement of the pause at a syntactic or non-syntactic boundary (blue vs. orange). Black triangles are means, white lines are medians.

5.1 Data

From Simple Wikipedia[2], we select sentences that contain exactly one comma, marking a syntactic boundary.[3] We select boundaries that signal major structural breaks, which typically lead to an audible intonational boundary in spoken language. We use dependency parsing to detect such structural breaks (examples are listed in Appendix Table 5). We create different versions of each sentence, such that the TTS systems have access to different cues for potentially generating an intonational boundary. After synthesizing these sentences, we measure the presence of an intonational boundary at position A:

1. **Comma cue + Syntactic cue**: *Most links are blue,*$_A$ *but they can be any color.*

2. **Syntactic cue**: *Most links are blue*$_A$ *but they can be any color.*

In (1), the systems can use the comma at position A as a cue for generating an intonational boundary. Additionally, they can use the fact that position A is a clause boundary. In (2), the systems can only rely on the clause boundary information, since the comma is absent.

To investigate the extent to which TTS systems generate intonational boundaries at syntactically unnatural positions, we measure the presence of an intonational boundary at position B:

3. **Unnatural comma cue**: *Most links are blue but they can,*$_B$ *be any color.*

4. **No cue**: *Most links are blue but they can*$_B$ *be any color.*

In (3), the systems can use the comma as a cue for generating an intonational boundary at position B (although it appears at a syntactically unnatural position). In (4), there is no cue that indicates the need for an intonational boundary at position B.

5.2 Evaluation

Besides comparing the durations for critical regions (i.e., the (pre-)boundary position) across conditions[4], we compute a **Syntactic Sensitivity Score** for each system. This consists of precision, recall and F1 scores based on the following counts in the sentences without commas: **True Positives** occur when the model generates a pause at a syntactic boundary (position A), **False Positives** when it generates a pause at a syntactically unnatural position (position B), **False Negatives** when no pause is generated at position A, and **True Negatives** when no pause is generated at position B.

5.3 Results

Figure 2 shows the durations for the pre-boundary word and boundary pause, depending on condition. We see that all models show a similar pattern: the strongest intonational boundaries are produced in the Syntactic + Comma cue condition. None of the models produce an intonational boundary in the No cue condition. The Syntactic cue and Unnatural comma cue conditions are inbetween, with the comma cue leading to a slightly stronger intonational boundary than the syntactic cue. This indicates that in simple sentence structures, TTS systems do pick up on syntactic cues, but that commas simply provide more direct evidence for intonational boundaries. It also shows that models

---

[2] https://simple.wikipedia.org

[3] Additional filters were applied: sentences had to be between 7 and 15 words long and free of digits, punctuation (except commas and final periods), and bracketed phrases.

[4] The words preceding the syntactic boundary position A and non-boundary position B may have different lengths, which could affect the average duration. To account for this, we averaged word duration by syllable count.

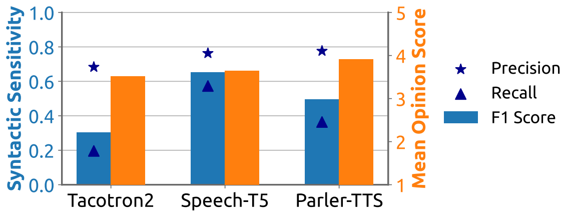


Figure 3: Syntactic Sensitivity versus Mean Opinion Score across TTS systems. The F1 score represents the harmonic mean of a system's precision and recall in generating pauses at syntactic boundaries.

integrate evidence from multiple sources: the combination of a comma and a syntactic cue leads to a stronger intonational boundary than only one cue.

In Figure 3, we compare our Syntactic Sensitivity score with reported MOS for each system. We see that *precision* mirrors the MOS pattern (Tacotron2 < Speech-T5 < Parler-TTS), while Speech-T5 has better *recall* than Parler-TTS. In other words: False Positives (i.e., pauses placed at syntactically unnatural positions) seem to affect human ratings more than False Negatives (i.e., no pauses at syntactic boundaries). This illustrates that our Syntactic Sensitivity score provides complementary insights that MOS does not capture.

6 Interpreting Boundary Placement

In the previous experiments, we used controlled stimuli to analyze how two specific cues influence intonational boundary placement in TTS systems. It could be the case, however, that systems' predictions are modulated by the presence of lexical items associated with pauses (e.g., conjunctions such as *but*, *and*, *or*). To gain insight into these cues, we train regression models with a range of different predictor variables to approximate the intonational boundary placement behavior of each TTS system.

For each TTS system, we train two regression models to predict the following outcome variables for a given position in a sentence: the duration of a pause in that position (**pause duration**), and the duration of the word before that pause (**pre-boundary word duration**) (i.e., the two aspects of an intonational boundary we focus on). We again use the sentences from Simple Wikipedia as input and extract the features listed in Table 1 at the positions marked as *A* or *B* (see Section 5.1).

Implementation Since we have a large number of (correlated) features, we use LASSO (Least Absolute Shrinkage and Selection Operator; Tibshirani 1996). This regularization technique introduces a penalty term that encourages sparsity in the model, allowing only a subset of features to be used in predicting the outcome variable, preventing overfitting and reducing the effect of multicollinearity (when features are highly correlated, LASSO tends to select only one of them). We apply standardization to the numerical features to ensure they have a mean of zero and a standard deviation of one (unit variance). We train the regression models on 80 percent of the data and evaluate on the remaining 20 percent. We use R^2 (explained variance) as our evaluation metric to gauge how well the predicted regression lines fit the data.

Category	Predictor
Punctuation	Comma Presence (1 or 0)
Lexical	Preceding POS tag (one-hot) Following POS tag (one-hot)
Constituency	Is Clause Boundary (1 or 0) Num. Closing Brackets Max. Tree Depth
Dependency	Preceding Token: Is Dep. Head (1 or 0) Preceding Token: Num. Dependents Preceding Token: Depth in Subtree
Length	Preceding Token Length Following Token Length Sentence Length Number of Preceding Tokens
Interaction	Is Clause Boundary * Comma Presence

Table 1: Predictor variables for regression models. Global features are extracted from the entire sentence; the other features are extracted at the boundary positions described in Section 5.1.

6.1 Results

Model	Pause Dur.	Pre-boundary Word Dur.
Parler-TTS	.14	.37
Speech-T5	.30	.44
Tacotron2	.44	.42

Table 2: Explained variance (R^2) of linear regression models for predicting *pause duration* and *pre-boundary word duration* as generated by three different TTS systems. Reported scores are for a held-out test set.

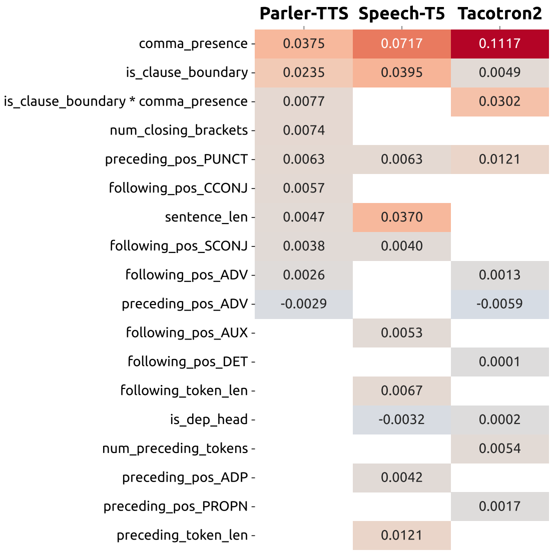
Performance The performance of the regression models is displayed in Table 2. We see that our predictor variables generally explain more variance in the *pre-boundary word duration* data compared to the *pause duration* data, which makes sense given that we use explicit features of the pre-boundary word (e.g., its length). We also see that pause duration is more predictable for Tacotron2 than for the other two systems. The behavior of Parler-TTS is least predictable, indicating that this model relies on other features than the ones we included in our regression models, or on more complex interactions between those features.

Feature Importance Figure 4 shows the top 10 selected predictors for *pause duration* for each of the TTS systems, together with their regression coefficients. We see that *comma presence* is the strongest predictor for all three TTS systems, verifying their strong reliance on punctuation cues. For Parler-TTS and Speech-T5, *is clause boundary* is also an important predictor.[5] We also see that specific lexical items are selected, e.g., words with the POS tag SCONJ or CCONJ. Depending on the model, different length-related features are also selected: *sentence length* for Parler-TTS and Speech-T5, *preceding/following token length* for Speech-T5, and *num. preceding tokens* for Tacotron2.

Overall, the analysis confirms that punctuation plays a major role in determining the duration of intonational boundaries in TTS systems. It also demonstrates that specific lexical items and length-related features influence pause duration. This reliance on surface cues is particularly evident in the LSTM-based system Tacotron2, while the Transformer-based systems Parler-TTS and Speech-T5 also seem to incorporate some syntactic information.

7 Changing the Training Distribution

While TTS systems may see plenty of examples of simple syntactic structures with obvious clause boundaries, garden path sentences are likely underrepresented in their training data. Sentences with attachment ambiguity may occur more frequently. However, even for such sentences, the intonation patterns we aim to capture (where high attachment introduces an intonational boundary and low attachment does not) may still be rare in the training data. As discussed in Section 2, speakers use distinct intonation patterns to disambiguate high and low attachment in conversational settings, helping to convey the intended meaning. In non-conversational speech, this distinction is less frequently observed.



	Parler-TTS	Speech-T5	Tacotron2
comma_presence	0.0375	0.0717	0.1117
is_clause_boundary	0.0235	0.0395	0.0049
is_clause_boundary * comma_presence	0.0077		0.0302
num_closing_brackets	0.0074		
preceding_pos_PUNCT	0.0063	0.0063	0.0121
following_pos_CCONJ	0.0057		
sentence_len	0.0047	0.0370	
following_pos_SCONJ	0.0038	0.0040	
following_pos_ADV	0.0026		0.0013
preceding_pos_ADV	-0.0029		-0.0059
following_pos_AUX		0.0053	
following_pos_DET			0.0001
following_token_len		0.0067	
is_dep_head		-0.0032	0.0002
num_preceding_tokens			0.0054
preceding_pos_ADP		0.0042	
preceding_pos_PROPN			0.0017
preceding_token_len		0.0121	

Figure 4: Coefficients of LASSO-selected predictor variables for pause durations of TTS systems.

Consequently, TTS systems trained on audiobooks may not have sufficient exposure to the nuanced intonation patterns associated with the different syntactic structures.

7.1 Training data analysis

Out of the three TTS systems we investigated, Parler-TTS was trained on the largest amount of data. To check if it missed important evidence for high and low attachment structures, we selected a subset of the MLS corpus that Parler-TTS was trained on (5000 examples, ~12000 sentences) and counted the occurrences of pauses, commas, and frequent prepositions[6], as well as the overlap between them. The detailed results are shown in Appendix Figure 7. While we cannot directly determine how often the model encountered high or low attachment structures, we observe that prepositions *without* a preceding pause (aligning with low attachment) appeared almost 5 times more frequently than those *with* a preceding pause (aligning with high attachment). This imbalance may explain why the model struggles to generate distinct intonation patterns for the two structures.

7.2 Altering the training distribution

We hypothesize that a greater balance in the occurrence of high and low attachment structures in the training data will enable the model to generate more varied intonation patterns that better reflect the underlying structure. To test this hypothesis,

---

[5] We verified that *is clause boundary* was a predictor by itself by running LASSO on different subsets of sentences: with/without commas, and with/without predictive lexical items (e.g., conjunctions). In all cases, *is clause boundary* was still selected as an important predictor.

[6] *of, to, in, for, with, as, at, on, by, for*.

we conducted two finetuning experiments aimed at rebalancing the data. These experiments are not meant to directly improve the performance of Parler-TTS, but merely to diagnose the role of (lack of) exposure to certain structures.

Finetuning on sampled data For the first experiment, we selected all sentences from the Jenny corpus[7] containing a preposition preceded by a pause (~5000 sentences, ~6 hours of speech). To ensure that the model would not be able to rely on commas as a cue for generating intonational boundaries, we removed all commas from the transcriptions. Our hope was that showing the model more examples of *general* PPs preceded by a pause would lead to more varied intonation patterns for sentences with an *ambiguous* PP.

Finetuning on synthetic data For the second experiment, we created a synthetic dataset to provide the model with more explicit examples of high and low attachment. Using the template described in Section 4.2, we generated 2500 sentences with a semantic bias towards high attachment, and 2500 sentences with a bias towards low attachment (resulting in ~6 hours of speech). We synthesized these sentences using Tacotron2, inserting commas at positions that would correspond to intended pauses (e.g., before the preposition *with* in high attachment cases). We again removed these commas from the text to ensure that the model could not rely on punctuation, but instead learn to use the semantic bias of the sentences to predict the presence of a pause.

Evaluation We created an evaluation set consisting of sentences containing function words that could be interpreted in two different ways, with one interpretation requiring a pause before the word (e.g., *The boy looked at the painting <pause> with genuine interest*) and the other not (e.g., *The boy looked at the painting with muted colors*). These function words include *with* (our primary example for high and low attachment), but also *as*, *for*, and *to*, as shown in Table 6 in the Appendix. We created 30 sentences per category and sampled them three times from the models (using three different random seeds). We then measured the pause duration before the critical function word across the resulting 90 data points.

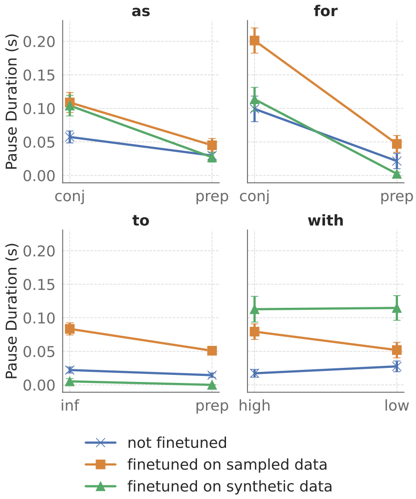


Figure 5: Average pause duration before the function words *as, for, to, with* (each used in two different ways, e.g., as a preposition vs. conjunction) for three versions of Parler-TTS. Error bars indicate the standard error.

7.3 Results

Figure 5 shows that the model finetuned on sampled data (orange lines) generates longer pauses than the non-finetuned model (blue lines). Interestingly, this increase is more pronounced in contexts where a pause is expected, i.e., before *for* and *as* when used as a conjunction, before *to* when used as an infinitive, and before *with* in the high attachment case. This suggests that training the model on a more balanced data distribution leads to more distinct intonation patterns that reflect different syntactic structures.

In contrast, the model finetuned on synthetic data did not learn to distinguish between high and low attachment based on semantic cues, as the pause duration before *with* remains the same in both cases (although it did increase compared to the non-finetuned model). These results indicate that, even with more exposure, TTS systems cannot disambiguate syntactic structure based on semantic cues. However, this observation requires further investigation, particularly regarding the role of natural versus synthetic speech and the amount of data necessary for robust results.

8 Conclusion

We evaluated the syntactic sensitivity of TTS systems by analyzing their intonation patterns generated for controlled stimuli. We find that systems can identify obvious clause boundaries in simple sentences but struggle with more complex, locally

---

[7] `https://github.com/dioco-group/jenny-tts-dataset`

or globally ambiguous structures. We also investigated the role of (lack of) exposure to such structures, and show that systems can generate more syntax-aligned intonation patterns if provided with appropriate evidence.

Future work should study a broader range of phenomena to better understand the types of linguistic associations captured by TTS systems. One potential direction would be to develop a resource similar to BLiMP (Warstadt et al., 2020) for TTS, which could serve as a more comprehensive framework for evaluating their syntactic sensitivity. Additionally, *structural probing* (Hewitt and Manning, 2019; Shen et al., 2023) could offer a more detailed look at the internal representation of syntax in TTS systems, complementing our behavioral measures.

9 Acknowledgements

We are grateful to Grzegorz Chrupała, Jaap Jumelet and Tom Lentz for reviewing early versions of this paper. This research is funded by the Netherlands Organisation for Scientific Research (NWO) through NWA-ORC grant NWA.1292.19.399 for *InDeep*.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, et al. 2022. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738.

Thomas G Bever. 1970. The cognitive basis for linguistic structures. cognition and the development of language, ed. by John R. Hayes, 279-362.

Dwight Bolinger. 1989. Intonation and its uses: Melody in grammar and discourse. *Edward Arnold.*

Sara Bögels, Herbert Schriefers, Wietske Vonk, Dorothee J. Chwilla, and Roel Kerkhofs. 2010. The Interplay between Prosody and Syntax in Sentence Processing: The Case of Subject- and Object-control Verbs. *Journal of Cognitive Neuroscience*, 22(5):1036–1053.

Cheng-Han Chiang, Wei-Ping Huang, and Hung-yi Lee. 2023. Why we should report the details in subjective evaluation of tts more rigorously. In *Proc. Interspeech 2023*, pages 5551–5555.

W Cooper. 1980. Syntax and speech.

William E. Cooper. 1976. Syntactic control of timing in speech production: a study of complement clauses. *Journal of Phonetics*, 4(2):151–171.

William Croft. 1995. Intonation units and grammatical structure. *Linguistics*, 33(5):839–882.

Anne Cutler, Delphine Dahan, and Wilma Van Donselaar. 1997. Prosody in the comprehension of spoken language: A literature review. *Language and speech*, 40(2):141–201.

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.

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

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.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Jaap Jumelet, Willem Zuidema, and Arabella Sinclair. 2024. Do language models exhibit human-like structural priming effects? In *Findings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 14727–14742. Association for Computational Linguistics (ACL).

Ambika Kirkland, Shivam Mehta, Harm Lameris, Gustav Eje Henter, Eva Székely, and Joakim Gustafson. 2023. Stuck in the MOS pit: A critical analysis of MOS test methodology in TTS evaluation. In *12th Speech Synthesis Workshop (SSW) 2023*.

Margaret M. Kjelgaard and Shari R. Speer. 1999. Prosodic facilitation and interference in the resolution of temporary syntactic closure ambiguity. *Journal of Memory and Language*, 40(2):153–194.

Dennis H. Klatt. 1975. Vowel lengthening is syntactically determined in a connected discourse. *Journal of Phonetics*, 3(3):129–140.

Yuma Koizumi, Heiga Zen, Shigeki Karita, Yifan Ding, Kohei Yatabe, Nobuyuki Morioka, Michiel Bacchiani, Yu Zhang, Wei Han, and Ankur Bapna. 2023. LibriTTS-R: A restored multi-speaker Text-to-Speech corpus.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative Adversarial Networks for efficient and high fidelity speech synthesis. *CoRR*, abs/2010.05646.

Tanya Kraljic and Susan E. Brennan. 2005. Prosodic disambiguation of syntactic structure: For the speaker or for the addressee? *Cognitive Psychology*, 50(2):194–231.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. 2023. High-fidelity audio compression with improved RVQGAN.

Sébastien Le Maguer, Simon King, and Naomi Harte. 2024. The limits of the Mean Opinion Score for speech synthesis evaluation. *Computer Speech & Language*, 84:101577.

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.

Dan Lyth and Simon King. 2024. Natural language guidance of high-fidelity text-to-speech with synthetic annotations. *arXiv preprint arXiv:2402.01912*.

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.

Marina Nespor and Irene Vogel. 2007. *Prosodic Phonology: With a new foreword*, volume 28. Walter de Gruyter.

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. IEEE.

Efrat Pauker, Inbal Itzhak, Shari R. Baum, and Karsten Steinhauer. 2011. Effects of Cooperating and Conflicting Prosody in Spoken English Garden Path Sentences: ERP Evidence for the Boundary Deletion Hypothesis. *Journal of Cognitive Neuroscience*, 23(10):2731–2751.

Janet Breckenridge Pierrehumbert. 1980. *The phonology and phonetics of English intonation*. Ph.D. thesis, Massachusetts Institute of Technology.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. MLS: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Joel Pynte. 1996. Prosodic breaks and attachment decisions in sentence parsing. *Language and cognitive processes*, 11(1-2):165–192.

AJ Schafer, SR Speer, and P Warren. 2005. Prosodic influences on the production and comprehension of syntactic ambiguity in a game-based conversation task. *Approaches to studying world situated Language Use: Psycholinguistic, linguistic and computational perspectives on bridging the product and action tradition*.

E Selkirk. 1984. Phonology and syntax. *The relation between sound and structure*.

Gaofei Shen, Afra Alishahi, Arianna Bisazza, and Grzegorz Chrupała. 2023. Wave to Syntax: Probing spoken language models for syntax. In *Proc. INTERSPEECH 2023*, pages 1259–1263.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4779–4783. IEEE.

Kim EA Silverman, Mary E Beckman, John F Pitrelli, Mari Ostendorf, Colin W Wightman, Patti Price, Janet B Pierrehumbert, and Julia Hirschberg. 1992. ToBI: A standard for labeling English prosody. In *ICSLP*, volume 2, pages 867–870.

Jesse Snedeker and John Trueswell. 2003. Using prosody to avoid ambiguity: Effects of speaker awareness and referential context. *Journal of Memory and Language*, 48(1):103–130.

Mark Steedman. 1991. Structure and intonation. *Language*, 67(2):260–296.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Duane Watson and Edward Gibson. 2004. The relationship between intonational phrasing and syntactic structure in language production. *Language and cognitive processes*, 19(6):713–755.

Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. 2019. LibriTTS: A corpus derived from LibriSpeech for Text-to-Speech. *Interspeech 2019*.

A Appendix Figures and Tables

Figure 6, Figure 7 and Table 3, Table 4, Table 5, Table 6 are shown on the next pages.

B Limitations

This study has several limitations that should be acknowledged. First, we analyzed intonational boundaries based duration measures only. While pause duration and word lengthening are well-established proxies for intonational boundaries, other prosodic features (e.g., pitch contour and intensity) also contribute significantly to their perception. Although previous research suggests that duration measures alone can reliably indicate the presence of an intonational boundary, and that pitch and intensity are less consistent across speakers and contexts (Kjelgaard and Speer, 1999; Bögels et al., 2010; Pauker et al., 2011), incorporating these additional prosodic cues would allow us to better characterize intonational structure (as generated by TTS systems).

Second, we did not consider different levels of boundary strength, a distinction made by the Tones and Break Indices (ToBI) framework (Silverman et al., 1992). Future work could benefit from adopting this gradation to more fully capture the complexity of intonational phrasing.

Third, the Parler-TTS model supports conditioning on voice characteristics specified through natural language descriptions. However, in this study, we only used a single voice description to synthesize our stimuli. It remains an open question how varying these voice characteristics might influence the resulting intonation patterns.

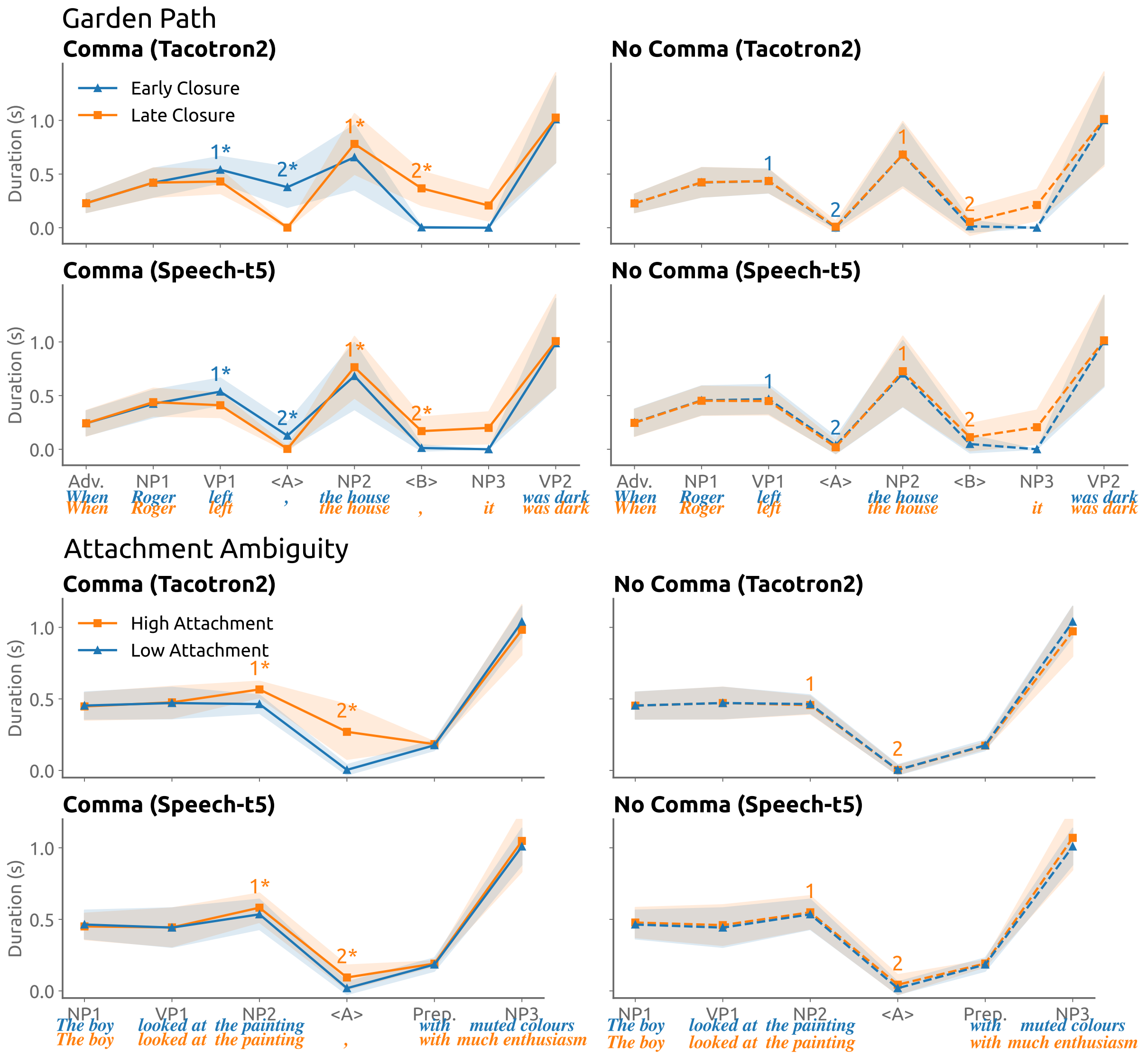


Figure 6: Average durations of sentence regions in garden path sentences (top) and sentences with attachment ambiguity (bottom), generated by Tacotron2 and Speech-T5. An intonational boundary consists of *lengthening* at the pre-boundary position (1), and insertion of a *pause* at the syntactic boundary position (2); asterisks indicate the presence of these effects. Example sentences are annotated on the x-axes; shading indicates the standard deviation across sentences.

Table 3: Garden path stimuli for Experiment 1. Sentences were presented in two forms: **early closure** (without the word in brackets) and **late closure** (with the word in brackets).

Stimulus
Whenever John walks the dogs (*cats*) are chasing him.
Because John studied the material (*it*) is clearer now.
When Whitesnake plays the music (*it*) is loud.
When Tim presents the lectures (*they*) are interesting.
When the original cast performs the plays (*they*) are funny.
When Madonna sings the song (*it*) is a hit.
Whenever John swims the channel (*it*) is choppy.
When Roger left the house (*it*) was dark.
Whenever Frank performs the show (*it*) is fantastic.
Because Mike phoned his mother (*she*) is relieved.
When the clock strikes the hour (*it*) is midnight.
Whenever the guard checks the door (*it*) is locked.
If Laura folds the towels (*they*) are neat.
If George programs the computer (*it*) is sure to crash.
If Charles babysits the children (*they*) are happy.
When the maid cleans the rooms (*they*) are immaculate.
Before Jack deals the cards (*they*) are shuffled.
While the boy read books (*televisions*) were stolen.
When the dog bites cats (*mice*) run away.
When the man batted balls (*players*) covered the field.
While the man parked cars (*bikes*) were waiting.
After the puppy licked kids (*parents*) were laughing.
Because snakes eat mice (*toads*) hide.
When a bear approaches people (*dogs*) come running.
After the chef cooked cake (*coffee*) was served.
While the artist painted clouds (*stars*) were appearing.
As the cat climbed trees (*leaves*) were falling.
As John hunted the frightened deer (*it*) escaped through the woods.
When Anne visited the British relatives (*they*) were moving to London.
When Rita washed her favorite sweater (*it*) was torn to shreds.
When Joan left her old boyfriend (*he*) stalked her for two months.
While the assistant measured the delicate fabric (*it*) ripped and frayed.
When Greg returned the new car (*it*) was operating smoothly.
Because Cecelia baked the delicious homemade bread (*it*) was served at breakfast.
Even when Todd cleaned the small kitchen (*it*) smelled like old garbage.
Because Grandma knitted wool sweaters (*they*) would appear under the Christmas tree.
Because Maria read the financial news (*it*) was always at her fingertips.
As Sam pounded the thin metal (*it*) ripped and broke into pieces.
When Sonya painted the kitchen walls (*they*) were covered into obvious drops.
As Lia typed the eviction notice (*it*) was cancelled.
When Tina supervised the night crew (*it*) was very efficient.
As Gary watched the drunken workmen (*they*) stumbled off the platform.
When the sheriff patrolled the whole area (*it*) was very safe.
When the musician conducted the symphony orchestra (*it*) was at its peak.
When Molly sang the drinking songs (*they*) sounded like opera.

Table 4: Examples of attachment ambiguity stimuli for Experiment 1. Two prepositional phrases were constructed for each stimulus, the former creating a semantic bias towards high (VP) attachment, the latter creating a semantic bias towards low (NP) attachment.

Stimulus
The boy looked at the painting *with much enthusiasm* / *with muted colors*.
The woman described the table *with much enthusiasm* / *with the smooth surface*.
The man bought the vase *with much happiness* / *with red dots*.
The girl found the chair *with much ease* / *with blue stripes*.
The artist inspected the house *with much interest* / *with wooden details*.

Table 5: Example sentences and counts of selected dependency labels, taken from the Simple Wikipedia Corpus.

Dependency Label	**Example**	**Count**
Conjunction (conj)	Most links are blue, but they can be any color.	420
Adverbial clause modifier (advcl)	Unless the cache is cleared, the link will always stay dark blue.	161
Relative clause modifier (relcl)	Animals are eukaryotes with many cells, which have no rigid cell walls.	49
Appositional modifier (appos)	Almost all animals have neurons, a signalling system.	47
Clausal complement (ccomp)	In Thailand, stingray leather is used in wallets and belts.	67
Open clausal complement (xcomp)	Genes say to the cell what to do, like a language.	70

Category	**Pause**	**Example**
as (preposition)	no	She was hired as the new manager of the team.
as (conjunction)	yes	She left early as she had an important meeting to attend.
for (preposition)	no	The child picked up the toy for his friend who had dropped it.
for (conjunction)	yes	The child picked up the toy for he wanted to play with it.
to (preposition)	no	The man gave the book to his sister who wanted it.
to (infinitive)	yes	The man read the book to learn more about history.
with (preposition, high attach.)	yes	The boy looked at the painting with genuine interest.
with (preposition, low attach.)	no	The boy looked at the painting with muted colors.

Table 6: Example sentences for our evaluation set for the fine-tuning experiments: each function word can be used in two different ways, one of which is associated with a pause.

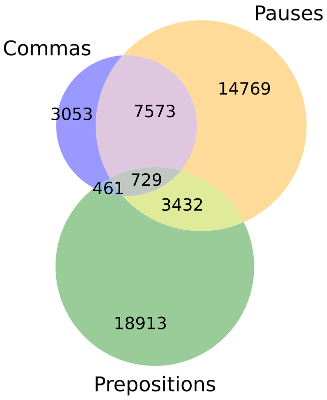


Figure 7: Counts of frequent prepositions, commas and pauses, as well as their overlap, in a subset of the training data of Parler-TTS.

Experiential Semantic Information and Brain Alignment: Are Multimodal Models Better than Language Models?

Anna Bavaresco, Raquel Fernández
Institute for Logic, Language and Computation
University of Amsterdam
`{a.bavaresco, raquel.fernandez}@uva.nl`

Abstract

A common assumption in Computational Linguistics is that text representations learnt by multimodal models are richer and more human-like than those by language-only models, as they are grounded in images or audio—similar to how human language is grounded in real-world experiences. However, empirical studies checking whether this is true are largely lacking. We address this gap by comparing word representations from contrastive multimodal models vs. language-only ones in the extent to which they capture experiential information—as defined by an existing norm-based 'experiential model'—and align with human fMRI responses. Our results indicate that, surprisingly, language-only models are superior to multimodal ones in both respects. Additionally, they learn more unique brain-relevant semantic information beyond that shared with the experiential model. Overall, our study highlights the need to develop computational models that better integrate the complementary semantic information provided by multimodal data sources.

`https://github.com/dmg-illc/exp-info-models-brain`

1 Introduction

How to link language representations to the real-world entities they refer to is a long-standing issue within semantics—the 'symbol-grounding problem' (Harnad, 1990; Bender and Koller, 2020). With the advent of large language models (LLMs) learning astounding linguistic abilities purely from text, this question has been reframed as the 'vector-grounding problem' (Mollo and Millière, 2023), gaining new relevance. While some researchers think that word meanings should be intended as purely symbolic (Fodor, 1983), others believe that words have meanings *precisely because* they are linked to specific entities, experiences or notions (Barsalou, 2008). Supporters of the latter view stress that human language acquisition is situated in a rich multimodal environment, where new words are learnt through interactions with objects and people (Vigliocco et al., 2014). Theories of embodied cognition further highlight the importance of linking words to concrete experience not only for their acquisition but also for their comprehension. Indeed, according to these theories, understanding sentences involves engaging perceptual, motor or emotional simulations of their content (for an overview, see Kaschak et al., 2024).

The idea of obtaining richer semantic representations by learning them from sources other than text, such as images or audio, has inspired a great deal of computational work, from early attempts at concatenating image and text embeddings (e.g., Bruni et al., 2014; Kiela and Bottou, 2014; Derby et al., 2018; Davis et al., 2019) to the most recent large vision-language models (LVLMs, e.g., Li et al., 2023; Wang et al., 2024; Liu et al., 2024; Deitke et al., 2024; Laurençon et al., 2024). Some of these works aimed to obtain language representations aligning more closely with human responses, such as similarity judgments, while others were more oriented towards improving performance on benchmarks or downstream applications. Regardless of the end goal, all these works present multimodality as a *desideratum*, assuming that images provide additional semantic information that cannot be learnt from text alone; however, there is little to no work investigating *which* these semantic aspects are. In this paper, we aim to fill this gap by addressing the following question: *Do multimodal models learn some facets of meaning related to perceptual experiences that language-only models cannot capture?*

Operationalising the 'extra-linguistic' information that multimodal models are allegedly learning is a prerequisite for approaching this issue. We did this by relying on a semantic model introduced by Fernandino et al. (2022) to capture 'experiential information'. This cognitive model represents words as n-dimensional arrays where each entry

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 141–155
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

corresponds to aggregated human ratings on a predefined experiential attribute (e.g., *Vision*, *Motion*, *Harm*). We focused on a set of nouns and evaluated the alignment between their representations provided by the experiential model and those by comparable unimodal (language-only) and multimodal (vision-language and audio-language) computational models. This analysis allowed us to uncover if multimodal models indeed reflect more semantic information than language-only models. Next, we checked whether capturing experiential information translates into higher alignment with brain responses recorded with functional magnetic resonance imaging (fMRI) to the same set of nouns.

Our findings indicate several interesting trends. First, both vision-language and language-only models exhibit significant alignment with the experiential model and brain responses, while the audio-language model displays weak or non-significant correlations. Second, this alignment is more pronounced for language-only models, which appear to capture a great deal of brain-relevant information beyond experiential. Lastly, language-only models remain more brain-aligned than vision-language models even when focusing on a set of more concrete words, although the gap is reduced. Overall, our study shows that current multimodal models learn *less* brain-relevant information—both experiential and beyond—than comparable language-only models, highlighting the need to explore different approaches to construct multimodal word representations.

2 Background

2.1 Embodied cognition

Embodied cognition identifies a suite of theoretical frameworks holding that language is understood by perceptual, emotional, or motor simulations of its content (e.g., Barsalou, 1999; Glenberg and Gallese, 2012; Zwaan, 2014; Pulvermüller, 2018). This general principle has received empirical support from multiple studies, both behavioural and neuroscientific.

For example, a series of works on the Action-sentence Compatibility Effect (ACE, Glenberg and Kaschak, 2002) and its subsequent variants (Borreggine and Kaschak, 2006; Zwaan and Taylor, 2006; Bub and Masson, 2012) revealed a significant difference in reaction times—attributed to motor simulations—when participants had to respond to a sentence (e.g., *You passed the note to Art*) with a movement matching (extending their arm) vs. non-matching (retreating their arm) that mentioned in the sentence. Similarly, the sentence-picture verification task (Stanfield and Zwaan, 2001), where participants have to respond to a picture that is either compatible (an eagle with its wings outstretched) or incompatible (an eagle with its wings folded) with a sentence (*The eagle is in the sky*), and its variations (Connell, 2007; Hoeben Mannaert et al., 2017) have also been widely used to demonstrate the occurrence of perceptual simulation during language comprehension. In parallel, a line of neuroscientific studies have found evidence that semantic processing may activate motor (among others, Hauk et al., 2004; Tettamanti et al., 2005; Aziz-Zadeh et al., 2006) and perceptual brain regions (Kiefer et al., 2008; Van Dam et al., 2012).

2.2 Multimodal models of semantics

Embodied cognition and related ideas, such as *visual grounding*, have percolated from Cognitive Science to Computational Linguistics, motivating attempts to build semantic models that learn representations from data sources beyond text. Early efforts in this direction (e.g., Bruni et al., 2014; Kiela and Bottou, 2014; Lazaridou et al., 2015; Silberer and Lapata, 2012, 2014) were characterised by 1) a focus on developing human-aligned computational models of meaning and 2) limited computational modelling resources (large datasets of paired image-text inputs did not exist at the time, nor did large transformer-based architectures).

Recently, multimodal models have become more powerful and found application on a variety of downstream tasks (e.g., image captioning, image retrieval, or visual question answering). Some seminal works used a contrastive objective to learn aligned image and text representations (Radford et al., 2021; Jia et al., 2021), while others—often inspired by BERT's (Devlin et al., 2019) successes in language modelling—applied its underlying intuitions to the vision-language domain (Tan and Bansal, 2019; Li et al., 2019; Lu et al., 2019; Chen et al., 2020). Finally, state-of-the-art large vision-language models (LVLMs, e.g., Li et al., 2023; Wang et al., 2024; Liu et al., 2024; Deitke et al., 2024; Laurençon et al., 2024), usually combining a large language model (LLM) with an image encoder, can engage in strikingly human-like conversations about images. In contrast to the early attempts at multimodal modelling, these works share 1) a focus on solving, or improving performance

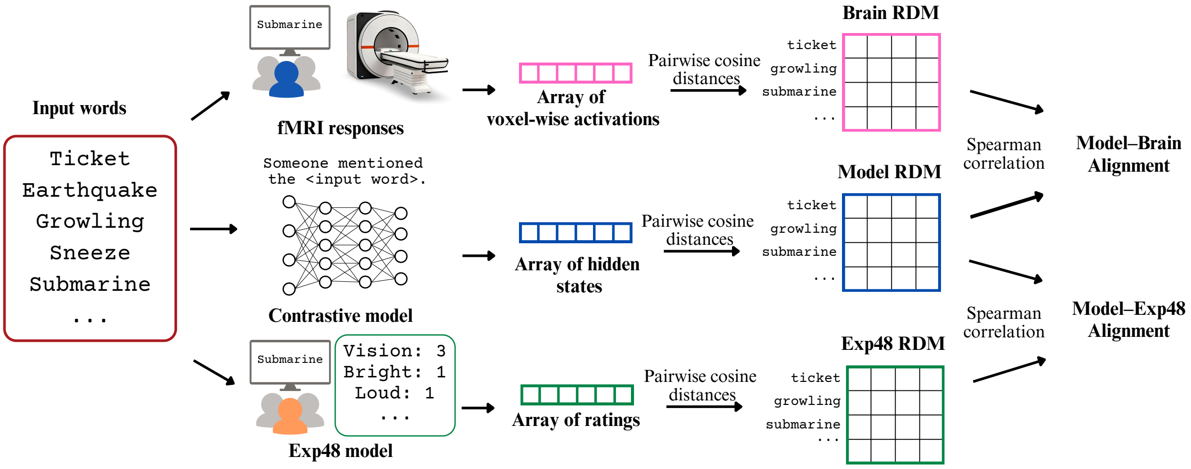


Figure 1: Overview of our experimental setup. Representations for the word stimuli are derived from three different sources: 1) fMRI responses; 2) multimodal and language-only contrastive models; 3) human ratings along the experiential dimensions of the EXP48 model. Next, pairwise distances between these word representations are used to populate representational dissimilarity matrices (RDMs). Finally, alignment between representational spaces is computed by correlating the off-diagonal elements of the RDMs.

on, downstream tasks, and 2) the availability of massive datasets and large models with billions of parameters.

For our experiments, we aimed to leverage models that are powerful while, at the same time, suitable for drawing cognitively-meaningful comparisons. The need to satisfy both constraints prevented us from evaluating state-of-the-art LVLMs; we elaborate more on our model choices in Section 3.2.

2.3 Experiential models of semantics

Recently, a few approaches motivated by embodied cognition have introduced models of semantics aimed at capturing ‘experiential information’, i.e., aspects of meaning related to how humans ground language in experiences. These experiential models were constructed by asking human annotators to rank words on a set of pre-defined dimensions. For example, Fernandino et al. (2022) introduced an experiential model based on 48 dimensions spanning perceptual, emotional, and action-related constructs. In two fMRI studies, they found that the experiential model yields more brain-aligned word representations than taxonomic and distributional models; additionally, it contributes unique semantic information not represented by the other models.

Similarly, Carota et al. (2024) experimented with a different experiential model based on 11 dimensions and compared its brain alignment against that of a distributional model. Their study revealed significant correlations with brain responses in more ROIs (regions of interest) for the experiential model than for the distributional model. However, an integrative model combining both displayed significant correlations in an even larger number of ROIs, suggesting that experiential and distributional are complementary aspects of human semantic processing.

Despite their merits, experiential models are bounded in their accuracy by an *a priori* selection of dimensions and, relying on human annotations, remain expensive to construct. These limitations open the intriguing question of whether experiential information can be captured by computational models learning semantic representations in a data-driven fashion.

3 Methods

A schematic of our experimental pipeline is provided in Figure 1. In the following, we describe in detail the word stimuli, brain responses, computational models and evaluation procedures.

3.1 Data and experiential model

For our experiments, we used word stimuli, fMRI responses and experiential model from Study 2 by Fernandino et al. (2022).[1] We describe each below.

Word stimuli Word stimuli comprise 320 nouns, half of which refer to *objects* and the other half

---

[1] These materials have been made publicly available by Fernandino et al. The full list of words and the experiential features can be found at `https://www.pnas.org/doi/10.1073/pnas.2108091119#supplementary-materials`; fMRI data are available at `https://osf.io/87chb/`.

to *events*. The 160 object nouns include an equal number of words (40) from four categories (*food*, *vehicles*, *animals*, *tools*); likewise, the event nouns span four semantic subcategories (*social event*, *negative event*, *sound*, *communication*).

fMRI responses fMRI responses were collected from 36 participants. While viewing the above-mentioned word stimuli one at a time, they were instructed to rate the frequency with which they experienced their corresponding entities in daily life. Voxel-wise activations (beta maps) for each noun relative to the mean signal across other nouns were estimated using linear regressions (for additional details, see Fernandino et al., 2022). Here, we focus on the betas from voxels within a 'semantic network ROI' defined by Binder et al. (2009) based on a meta-analysis. Voxel-wise beta coefficients can be arranged in vectors representing the brain response elicited by each noun.

Experiential model The experiential model, hereafter abbreviated as EXP48, represents each word as a set of ratings on 48 pre-defined dimensions capturing different aspects of people's experience with objects/events, e.g., *Vision*, *Hand action* or *Unpleasant*. The ratings were introduced by Binder et al. (2016) as part of a wider set of experiential salience norms; they range from 0 to 6 and were provided by 1743 unique crowdworkers.

3.2 Computational models

Our model choices were motivated by the goal to maximise comparability across architectures. More concretely, we selected three models (language-only, vision-language, and audio-language) comparable in terms of fine-tuning objective—the contrastive one—and architecture—they all have a pretrained BERT (Devlin et al., 2019) as language encoder.[2] One aspect in which these architectures differ is the amount of training data; however, we believe this issue does not invalidate our results and further discuss it in Section 6.

SimCSE (**Sim**ple **C**ontrastive Learning of **S**entence **E**mbeddings, Gao et al., 2021) is a language-only sentence encoder fine-tuned contrastively on 1M sentences randomly sampled from English Wikipedia. Matching pairs for the contrastive objective were created by applying different dropout masks to the same sentence.

MCSE (**M**ultimodal **C**ontrastive Learning of **S**entence **E**mbeddings, Zhang et al., 2022) is a vision-language sentence encoder fine-tuned by jointly optimising a SimCSE objective and a CLIP-like (Radford et al., 2021) objective. The fine-tuning data for the first objective is the same as SimCSE's; as for the CLIP-like objective, where a matching pair was defined by an image and its caption, the fine-tuning data consists of 83K images from MS-COCO (Lin et al., 2014) annotated with multiple captions.

CLAP (**C**ontrastive **L**anguage **A**udio **P**retraining, Wu et al., 2023) is an audio-language model whose language encoder was initialised with pre-trained BERT weights and fine-tuned on audio-caption pairs with a CLIP-like objective. The fine-tuning data includes $633,526$ audio-text pairs, with audio clips representing human activities, natural sounds, and audio effects.

For reference, we also tested BERT and VisualBERT (Li et al., 2019) as its visual counterpart.

BERT (Devlin et al., 2019) is a transformer-based language-only model pretrained with two objectives: masked language modelling and next sentence prediction. Its pretraining data includes the BooksCorpus (800M words, Zhu et al., 2015) and English Wikipedia (2500 words). As mentioned above, SimCSE, MCSE and CLAP fine-tuned pretrained BERT architectures.

VisualBERT (Li et al., 2019) is a vision-language model consisting of a BERT-based language encoder (initialised with parameters from pretrained BERT) and a pretrained visual feature extractor based on Faster RCNN (Ren et al., 2015). Its training objectives, which mirror BERT's, were masked language modelling with image input and sentence-image prediction. The vision-language pretraining data comprises MS-COCO and VQA 2.0 (Goyal et al., 2017). Note that this is *not* a contrastive model; we included it for reference as it can be considered as a vision-language extension of BERT, but it is not directly comparable with MCSE, SimCSE and CLAP.

3.3 Extracting representations

Given that all the models we considered were trained to learn contextualised representations

---

[2] All three models were released with both BERT-based (Devlin et al., 2019) and RoBERTa-based (Liu et al., 2019) implementations. We used the former in all our experiments.

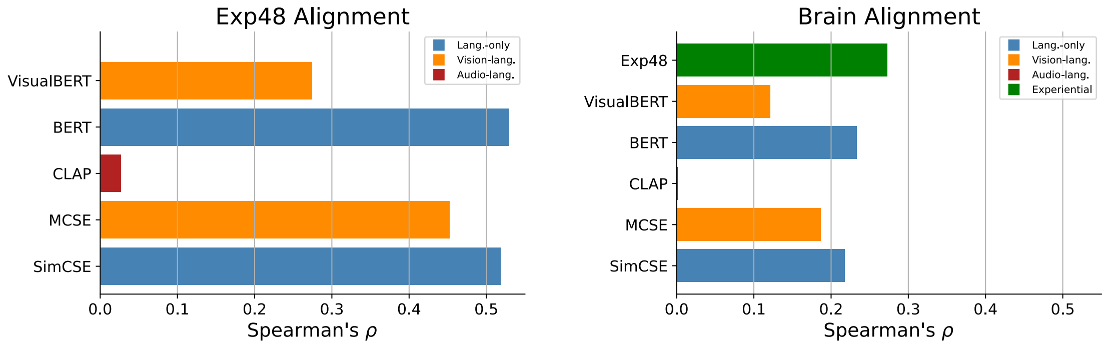


Figure 2: Results from representational similarity analysis. On the left, Spearman correlations quantifying the alignment between word representations from Exp48 and by computational models. On the right, Spearman correlations indicating the alignment between fMRI responses from human participants and word representations by computational models.

from sentences, single words may be an out-of-distribution input. Therefore, following an approach similar to May et al. (2019), we embedded the noun stimuli in a set of generic template sentences (e.g., `Someone mentioned the <word>`, see Appendix A for the complete list) when passing them to the models.[3] For all templates, we derived word representations from the hidden states of each layer; more specifically, we selected the hidden states corresponding to the tokens of the target word and averaged them across templates.

3.4 Alignment evaluation

RSA To compare model representations against Exp48 and brain responses, we used representational similarity analysis (RSA, Kriegeskorte et al., 2008), which quantifies the alignment between two representational spaces (either by two models or by a model and brain responses) as the correlation between representational dissimilarity matrices (RDMs). In our experiments, RDMs were populated with pairwise cosine distances between model representations or fMRI responses for all the unique word-pairs. fMRI RDMs for individual participants were averaged into one aggregated RDM. The alignment between this fMRI RDM and morel-derived RDMs was calculated as a Spearman correlation.

Partial correlations While RSA allows comparing models' alignment with Exp48 or brain responses, it does not reveal whether models explain shared variance or provide independent contributions. Fernandino et al. (2022) computed partial correlations to check how much brain-relevant information Exp48 shared with the other models they considered, i.e., two distributional models (Word2vec and GloVe; Mikolov et al., 2013; Pennington et al., 2014) and two taxonomic models (a WordNet-based model and a categorical one). We used the same approach to determine how much brain-relevant information our tested models share with Exp48 and with each other. Formally, partial correlations can be defined as follows: Consider the RDM from Model A y, the RDM from Model B x, and the RDM of the brain responses z. The partial correlation of Model A without Model B is $\rho(r_i, z_i)$, where $r_i = y_i - \hat{y}_i$ are the residuals from the linear regression with equation $\hat{y}_i = a + bx_i$.

4 Results

4.1 Exp48 and brain alignment across models

We performed RSA to obtain a first measure of model representations' alignment with Exp48 and fMRI responses. This analysis was conducted on model representations averaged across the three layers yielding the highest alignment individually; note that these layers may differ when considering alignment to brain responses vs. Exp48 (see Appendix B.2 for a visualisation of layer-wise alignment). The results from RSA against brain responses and Exp48 are displayed in Figure 2. All Spearman correlations are statistically significant ($p < 0.05$), except for CLAP's correlation with brain responses ($p = 0.70$); we additionally verified that all the pairwise differences between

---

[3] We empirically verified that passing words within templates rather than in isolation yields higher alignment with both the experiential model and brain responses (see Appendix B.1).

correlations are statistically significant.[4]

An inspection of correlations against EXP48 indicates BERT as the most aligned model ($\rho = 0.53$); SimCSE and MCSE also display moderate correlations with EXP48 ($\rho = 0.52$ and $\rho = 0.45$, respectively). In contrast, CLAP's representations are poorly aligned with EXP48, exhibiting a correlation of just 0.03. A comparison between vision-language models (MCSE and VisualBERT) and their unimodal counterparts (SimCSE and BERT) reveals that the former, surprisingly, reflect less experiential information than the latter.

Regarding alignment with brain responses in the semantic ROI, BERT is again the best model ($\rho = 0.23$), although it remains less brain-aligned than EXP48 ($\rho = 0.27$). All the other models display positive correlations, with the exception of CLAP, whose correlation is not statistically significant ($\rho = 0.00$, $p = 0.70$). Similarly to the EXP48-alignment results, here we found that the language-only models BERT and SimCSE are more brain-aligned than their vision-language extensions VisualBERT and MCSE. We delve deeper into the robustness of this finding in Section 5.

An interesting trend common across results from both RSAs (against EXP48 and fMRI responses) is that representations by SimCSE and MCSE—which have been shown to outperform BERT on semantic text similarity tasks (Gao et al., 2021; Zhang et al., 2022)—are *less* aligned than those by BERT. A potential explanation for this may be that we considered *single-word* representations. Since contrastive fine-tuning, as applied to SimCSE and MCSE, optimises *sentence*-level representations as opposed to *token*-level ones, it could be that some token-level semantic properties initially learnt by BERT got somehow diluted through this process.

4.2 Experiential information vs. unique contribution in models' brain alignment

Results from the partial correlation analysis are displayed in Figure 3, whose left-hand panel shows how much EXP48 representations align with brain responses without the information they share with each of the other models. An interesting observation is that the lowest correlations were obtained when regressing out BERT and SimCSE. This provides an interesting complement to the findings from RSA against EXP48 representations: RSA shows that BERT and SimCSE share substantial representational information with EXP48, and partial correlations suggest that this information is also brain-relevant. Regarding models' brain alignment without EXP48, displayed in Figure 3's right-hand panel, a noteworthy finding is that BERT's and SimCSE's representations are the most brain-aligned even after regressing out EXP48. This suggests that these models learnt some semantic information that is not captured by EXP48 but is still reflected in brain responses.

Additionally, for each model we checked which proportion of its initial brain alignment is attributable to unique contribution as opposed to information shared with EXP48. This can be visualised by comparing the dark-shade bars against the light-shade ones in the right-hand panel of Figure 3. An interesting result revealed by this comparison is that, although MCSE is more brain-aligned than VisualBERT, their unique contribution without EXP48 is the same in absolute value ($\rho = 0.06$); in other terms, 50% of VisualBERT's brain alignment is due to unique information, while in MCSE it is 32%. Regarding BERT and SimCSE, the majority of their initial brain alignment is eroded when regressing out EXP48; however, the asymmetry is not substantial, and the unique contribution accounts for more than 40% of the initial brain alignment in both models. As for CLAP, it exhibits a weak negative correlation that is not statistically significant, confirming that the model does not contribute any brain-relevant information.

Finally, we used partial correlations to compare vision-language models (VLMs) against their language-only counterparts (LMs). We found that neither MCSE ($\rho = 0.00$; $p = 0.60$) nor VisualBERT ($\rho = 0.00$; $p = 0.66$) exhibit statistically significant correlations with brain responses once SimCSE and BERT, respectively, are regressed out. Crucially, this indicates that VLMs did not learn any additional brain-relevant information besides that already captured by their LM counterparts.

5 Assessing Results' Robustness

RSA results revealed a consistent advantage of language-only models over the multimodal ones. This finding contrasts with the expectation—shared across a great deal of work on multimodality and

---

[4] Statistical significance was determined by applying a Fisher transformation to the correlation coefficients from each pair of models and calculating the p-value associated with the difference between the two z-scores. All p-values were Bonferroni-corrected with $\alpha = 0.05$. The same approach for verifying statistical significance was applied to all correlation comparisons throughout the paper.

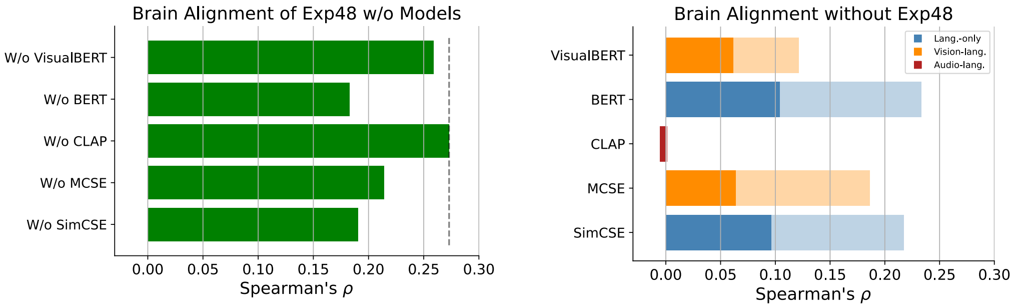


Figure 3: Results from partial correlation analyses. On the left, Spearman correlations between brain responses and the residuals obtained regressing model RDMs out of the EXP48 RDM. The dotted line indicates EXP48's initial brain alignment without removing any information. On the right, Spearman correlations between brain responses and the residuals obtained regressing the EXP48 RDM out of model RDMs. The bars in lighter shades indicate models' initial brain alignment.

language modelling—that training models on diverse data modalities, as opposed to text alone, should yield more human-like language representations. In the following, we present two analyses aimed at assessing the robustness of these findings. Given that the audio-language model CLAP did not achieve a statistically significant brain alignment, we excluded it from further analyses and focused on the remaining vision-language and language-only architectures.

5.1 Do caption-like templates result in improved brain alignment?

As pointed out by Tan and Bansal (2020), image captions are examples of *grounded language*, which differs from other types of natural language along many dimensions. Since the VLMs we evaluated were trained on image-caption pairs, they may have over-fitted to the language present in captions. Therefore, it is possible that the sentence templates we used to obtain contextualised word representations from the models are somehow out-of-distribution for VLMs.

To control for this potential confound, we re-extracted word representations employing different templates, whose structure was modelled around captions (e.g., `There is an <object> in a <place>`, or `A <person> is <verb in -ing> in a <place>`). These structures were identified based on a manual inspection of captions from MS-COCO, which was part of both MCSE's and VisualBERT's training. Given the challenges of creating caption-like templates providing a fitting context for all the word stimuli, we used different sets of templates for each sub-category of words described in Section 3.1 (e.g., `There is a <food-word> on a table in a restaurant` or `A few people gathered for a <social event-word>`). We provide the complete list of templates in Appendix A.

The procedure for calculating brain alignment was the same as that employed in the main experiment. Spearman correlations between model-derived RDMs and the fMRI-derived RDM are displayed in Figure 4. All correlations are statistically significant, as well as correlation differences between models. A comparison across models confirms the trend from the main experiment: Language-only models are more brain-aligned than their vision-language counterparts. This suggests that the finding is robust and not a by-product of the templates where word stimuli were embedded.

The dotted lines in Figure 4 allow comparing the brain alignment model representations achieve when using caption-like templates vs. when using the templates from the main experiment. This comparison reveals that all models—not only VLMs—exhibit higher brain alignment when using caption-like templates. We interpret this as indicating that caption-like templates are not more in-distribution for VLMs, but rather provide a better-specified context that is beneficial to all models.

5.2 Do VLMs yield more brain-aligned representations for objects vs. events?

Provided that VLMs learn additional semantic information, it could be that not all word representations benefit from multimodal training to the same extent; instead, a potential advantage may be more prominent for words referring to visual contents.

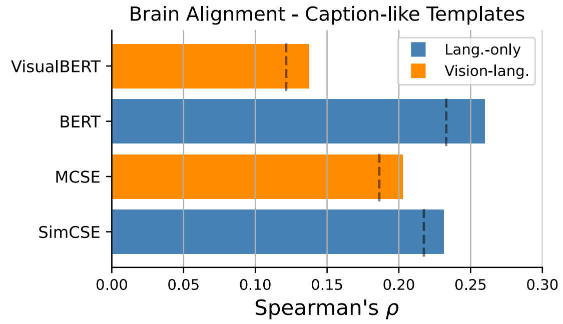


Figure 4: Spearman correlations indicating alignment between model representations extracted using caption-like templates and fMRI responses. Dotted lines indicate the initial correlations obtained with the templates from the main experiment.

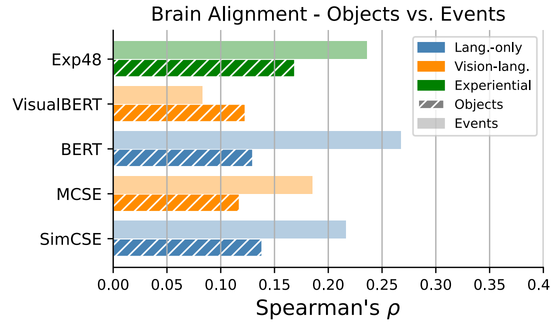


Figure 5: Spearman correlations indicating alignment between model representations and fMRI responses. Correlations are displayed separately for object-words and event-related words.

The words used in our main experiments include nouns from multiple semantic categories (see Section 3.1 for more details), which may largely vary in their degree of 'visual-ness'. Therefore, it is possible that we did not detect additional brain-relevant information learnt by VLMs because we focused on the 'wrong' words.

To check whether this is the case, we further analysed two word subsets with different levels of concreteness. The subsets were identified by leveraging the semantic labels already present in our word set, i.e., *objects* and *events*.[5] We repeated RSA separately for these two word subsets following the same procedure employed in the main experiment.

The results of this analysis are displayed in Figure 5. A first observation is that—for all models except VisualBERT—correlations are statistically significantly stronger for events than objects. This pattern was also reported by Fernandino et al. (2022), who attributed it to "higher variability of pairwise similarities for the neural representations of event concepts".

A second interesting result is that the model ranking we observed analysing the entire word set (BERT > SimCSE > MCSE > VisualBERT) is replicated for events but not for objects, where none of the differences between model correlations is statistically significant. While there is a negative effect overall, further training BERT on image-text pairs (as in VisualBERT) or fine-tuning it with a contrastive objective (as in SimCSE and MCSE) does not significantly alter the initial brain alignment of its object-word representations. Interestingly, EXP48, which we included for reference, is outperformed by BERT on events; however, it remains statistically significantly more brain-aligned than the other models on objects.

Finally, comparing vision-language models against their language-only counterparts shows that BERT and VisualBERT do not significantly differ regarding the brain alignment of their object-word representations, while SimCSE and MCSE do (with SimCSE remaining more aligned).[6] As for event-word representations, SimCSE and BERT are, respectively, significantly more brain-aligned than MCSE and VisualBERT. These results further support the robustness of our initial finding, i.e., that LMs models are more aligned than their VL counterparts. However, the reduced gap between the two model types when considering object-word representations vs. event-word ones suggests that VLMs *do*, comparatively, learn more brain-aligned representations for objects than events.

6 Discussion

While multimodal models are often expected to incorporate additional semantic aspects that language-only models cannot learn, our results reveal that their word representations are *less* aligned with EXP48 and fMRI responses than those by LMs. Moreover, within multimodal models, the vision-language ones show moderate positive correlations with EXP48 and fMRI responses, while the

---

[5] In their supplementary materials, Fernandino et al. (2022) report that the average concreteness score for *objects* is 4.9, while for *events* it is 3.6.

[6] Note that, since we used Bonferroni corrections, this difference is statistically significant here—but not when comparing all five models—due to a change in the number of relevant comparisons (2 vs. 5).

audio-language one correlates weakly with EXP48 and does not yield a significant correlation with brain responses. Below, we discuss factors that may have played a role in these partially unexpected results.

Amount of training data While being comparable in terms of learning objectives and architecture, SimCSE, MCSE and CLAP still differ in the amount of fine-tuning data. For the SimCSE–MCSE comparison, this does not appear to be a confound: Despite being fine-tuned on *less* sentences than MCSE, SimCSE still proves to be *more* EXP48- and brain-aligned. A potential reason for this may be that the grounded language employed in image captions causes a shift of semantic representations towards more concrete meanings. As for CLAP, the smaller amount of fine-tuning audio-caption pairs, together with its optimising only a CLIP-like objective (without a SimCSE-like one), may have played a role in its poor alignment.

Multimodal pretraining vs. fine-tuning A potential explanation for the inferior performance of multimodal models could be that training on multimodal pairs is not as effective during fine-tuning as it is during pre-training. However, we verified that not even the language encoder from the powerful CLIP (Radford et al., 2021)[7]—pretrained contrastively on 400M image-text pairs—yields more brain-aligned word representations than BERT and SimCSE (see Appendix B.3).

Models from present vs. past studies An interesting result was that the correlations with fMRI responses we observed for SimCSE, MCSE and BERT are higher than those achieved by the computational models (GloVe and Word2vec) tested by Fernandino et al. 2022 (see Appendix B.3). This finding aligns with previous work showing that transformer-based architectures are more predictive of brain responses during language processing than word-level embedding models and recurrent neural networks (Schrimpf et al., 2021). In addition, we found that the LMs and, to a larger extent, the VLMs we tested learn brain-relevant semantic information beyond that captured by EXP48. This partially echoes the results by Carota et al. (2024), with the difference that the computational model included in their study was strictly distributional.

Information captured by EXP48 While the ability of EXP48 to model brain responses has been validated by previous research, it may still be a suboptimal model of perceptual experience for two main reasons. First, all dimensions in EXP48, including the more perceptual ones like *Colour* or *Sound*, are somewhat abstract; in this sense, they may fail to capture low-level perceptual information relevant for modelling human word representations and, perhaps, learnt by multimodal models. Second, EXP48 encodes experiential dimensions, e.g., *Pleasant* or *Time*, which are not strictly perceptual and may be hard, if not impossible, to learn for vision-language and audio-language models.

Type of stimuli Our study focuses on single words that are not included in longer text passages. To some extent, our results suggest that this may affect *machine* language processing; indeed, we found that embedding words in sentences, as opposed to passing them to the models as is, yields more brain-aligned representations (see also Appendix B.1). In a similar vein, the amount of context may influence *human* language processing: As observed by Zwaan (2014), context determines the perceptual detail of the mental simulations people engage during language comprehension. Therefore, it may be that the nouns used in the fMRI experiment did not prompt multimodal semantic knowledge enough for it to be detected in our study.

7 Conclusions

Our study provides an in-depth comparison between multimodal and language-only architectures in their ability to capture experiential semantic information and alignment with brain responses. Contrary to common assumptions, we found multimodal models to produce word representations less brain-aligned and experience-informed than language-only models.

These results have several implications for future work. First, they invite caution against assuming that technical innovations allowing models to solve additional downstream tasks should necessarily make them more 'human-like'. Second, they indicate that there is significant room for improving current computational language models so that they learn the brain-relevant experiential information they currently lack—how to concretely achieve this remains an open question.

---

[7] This model was excluded from the main experiment as it is not directly comparable with the other architectures.

Limitations

Our experimental setup focuses exclusively on contrastive models which are not state-of-the-art for both linguistic and multimodal downstream tasks. More recent architectures pretrained autoregressively—e.g., models from the LLaVA family (Liu et al., 2024), Molmo (Deitke et al., 2024), or Qwen2.5-VL(Bai et al., 2025)—may exhibit different patterns. However, the complexity of their pre-training and fine-tuning steps makes it hard to set up a controlled comparison ruling out factors such as the amount of training data or training objectives. We therefore explicitly decided to not include this type of model in our investigation. This decision was further informed by preliminary evidence that generative vision-language models achieving stronger performance on downstream tasks are less brain-aligned than previous architectures (Bavaresco et al., 2024).

Acknowledgments

We thank the members of the Dialogue Modelling Group (DMG) from the University of Amsterdam and Lorenzo Proietti for the helpful feedback provided at different stages of this project.

This project was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819455).

References

Lisa Aziz-Zadeh, Stephen M Wilson, Giacomo Rizzolatti, and Marco Iacoboni. 2006. Congruent embodied representations for visually presented actions and linguistic phrases describing actions. *Current biology*, 16(18):1818–1823.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.

Lawrence W Barsalou. 1999. Perceptual symbol systems. *Behavioral and brain sciences*, 22(4):577–660.

Lawrence W Barsalou. 2008. Grounded cognition. *Annu. Rev. Psychol.*, 59(1):617–645.

Anna Bavaresco, Marianne de Heer Kloots, Sandro Pezzelle, and Raquel Fernández. 2024. Modelling multimodal integration in human concept processing with vision-and-language models. *arXiv preprint arXiv:2407.17914*.

Emily M Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5185–5198.

Jeffrey R Binder, Lisa L Conant, Colin J Humphries, Leonardo Fernandino, Stephen B Simons, Mario Aguilar, and Rutvik H Desai. 2016. Toward a brain-based componential semantic representation. *Cognitive neuropsychology*, 33(3-4):130–174.

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.

Kristin L Borreggine and Michael P Kaschak. 2006. The action–sentence compatibility effect: It's all in the timing. *Cognitive Science*, 30(6):1097–1112.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *Journal of artificial intelligence research*, 49:1–47.

Daniel N Bub and Michael EJ Masson. 2012. On the dynamics of action representations evoked by names of manipulable objects. *Journal of Experimental Psychology: General*, 141(3):502.

Francesca Carota, Hamed Nili, Nikolaus Kriegeskorte, and Friedemann Pulvermüller. 2024. Experientially-grounded and distributional semantic vectors uncover dissociable representations of conceptual categories. *Language, Cognition and Neuroscience*, 39(8):1020–1044.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. UNITER: UNiversal Image-TExt Representation Learning. In *European Conference on Computer Vision*, pages 104–120.

Louise Connell. 2007. Representing object colour in language comprehension. *Cognition*, 102(3):476–485.

Christopher Davis, Luana Bulat, Anita Lilla Vero, and Ekaterina Shutova. 2019. Deconstructing multimodality: visual properties and visual context in human semantic processing. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 118–124, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, et al. 2024. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Multimodal Models. *CoRR*.

Steven Derby, Paul Miller, Brian Murphy, and Barry Devereux. 2018. Using Sparse Semantic Embeddings Learned from Multimodal Text and Image Data to Model Human Conceptual Knowledge. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 260–270, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Leonardo Fernandino, Jia-Qing Tong, Lisa L Conant, Colin J Humphries, and Jeffrey R Binder. 2022. Decoding the information structure underlying the neural representation of concepts. *Proceedings of the National Academy of Sciences*, 119(6):e2108091119.

Jerry A Fodor. 1983. *The modularity of mind*. MIT press.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arthur M Glenberg and Vittorio Gallese. 2012. Action-based language: A theory of language acquisition, comprehension, and production. *cortex*, 48(7):905–922.

Arthur M Glenberg and Michael P Kaschak. 2002. Grounding language in action. *Psychonomic bulletin & review*, 9(3):558–565.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Olaf Hauk, Ingrid Johnsrude, and Friedemann Pulvermüller. 2004. Somatotopic representation of action words in human motor and premotor cortex. *Neuron*, 41(2):301–307.

Lara N Hoeben Mannaert, Katinka Dijkstra, and Rolf A Zwaan. 2017. Is color an integral part of a rich mental simulation? *Memory & cognition*, 45:974–982.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Michael P Kaschak, Michael Long, and Julie Madden. 2024. Embodied Approaches to Language Comprehension. In *The Routledge Handbook of Embodied Cognition*, pages 191–199. Routledge.

Markus Kiefer, Eun-Jin Sim, Bärbel Herrnberger, Jo Grothe, and Klaus Hoenig. 2008. The sound of concepts: Four markers for a link between auditory and conceptual brain systems. *Journal of Neuroscience*, 28(47):12224–12230.

Douwe Kiela and Léon Bottou. 2014. Learning Image Embeddings using Convolutional Neural Networks for Improved Multi-Modal Semantics. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. 2008. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pages 87874–87907. Curran Associates, Inc.

Angeliki Lazaridou, Marco Baroni, et al. 2015. Combining Language and Vision with a Multimodal Skip-gram Model. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 153–163.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. VisualBERT: A Simple and Performant Baseline for Vision and Language. *arXiv preprint arXiv:1908.03557*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved Baselines with Visual Instruction Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306.

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

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chandler May, Alex Wang, Shikha Bordia, Samuel Bowman, and Rachel Rudinger. 2019. On Measuring Social Biases in Sentence Encoders. 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 622–628.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Dimitri Coelho Mollo and Raphaël Millière. 2023. The vector grounding problem. *arXiv preprint arXiv:2304.01481*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Friedemann Pulvermüller. 2018. Neural reuse of action perception circuits for language, concepts and communication. *Progress in neurobiology*, 160:1–44.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

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.

Carina Silberer and Mirella Lapata. 2012. Grounded models of semantic representation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1423–1433. Association for Computational Linguistics.

Carina Silberer and Mirella Lapata. 2014. Learning grounded meaning representations with autoencoders. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 721–732. Association for Computational Linguistics.

Robert A Stanfield and Rolf A Zwaan. 2001. The effect of implied orientation derived from verbal context on picture recognition. *Psychological science*, 12(2):153–156.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. 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 5100–5111.

Hao Tan and Mohit Bansal. 2020. Vokenization: Improving Language Understanding with Contextualized, Visual-Grounded Supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080.

Marco Tettamanti, Giovanni Buccino, Maria Cristina Saccuman, Vittorio Gallese, Massimo Danna, Paola Scifo, Ferruccio Fazio, Giacomo Rizzolatti, Stefano F Cappa, and Daniela Perani. 2005. Listening to action-related sentences activates fronto-parietal motor circuits. *Journal of cognitive neuroscience*, 17(2):273–281.

Wessel O Van Dam, Margriet Van Dijk, Harold Bekkering, and Shirley-Ann Rueschemeyer. 2012. Flexibility in embodied lexical-semantic representations. *Human brain mapping*, 33(10):2322–2333.

Gabriella Vigliocco, Pamela Perniss, and David Vinson. 2014. Language as a multimodal phenomenon: implications for language learning, processing and evolution.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Miaoran Zhang, Marius Mosbach, David Adelani, Michael Hedderich, and Dietrich Klakow. 2022. MCSE: Multimodal Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5959–5969, Seattle, United States. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Rolf A Zwaan. 2014. Embodiment and language comprehension: Reframing the discussion. *Trends in cognitive sciences*, 18(5):229–234.

Rolf A Zwaan and Lawrence J Taylor. 2006. Seeing, acting, understanding: Motor resonance in language comprehension. *Journal of Experimental Psychology: General*, 135(1):1.

Appendix

A Sentence Templates

The neutral sentence templates where the word stimuli were embedded in order to obtain contextualised representations from the computational models were the following:

`Someone mentioned the <word>.`
`The post was about the <word>.`
`Everyone was talking about the <word>.`
`They were all interested in the <word>.`
`People know about the <word>.`

In one of our additional experiments (see Section 5.1), we used caption-like sentences to check whether they were more in-distribution for vision-language models and, therefore, yielded more EXP48- and brain-aligned representations. Below, we report the caption-like templates used for each word sub-category.

Templates used for the sub-category *food*:

`There is a <word> on a table in a restaurant.`
`A <word> is on a kitchen table.`
`A woman is eating a <word>.`
`A <word> with a few glasses around.`
`A close-up of a <word>.`

Templates used for the sub-category *vehicle*:

`There is one man in a <word>.`
`A <word> is surrounded by a few people.`
`A woman is posing next to a <word>.`
`A <word> with a young man next to it.`
`A close-up of a <word>.`

Templates used for the sub-category *tool*:

`There is a man holding a <word>.`
`A <word> is lying on the ground.`
`A woman is using a <word>.`
`A <word> with some people in the background.`
`A close-up of a <word>.`

Templates used for the sub-category *animal*:

`There is a <word> eating voraciously.`
`A man is feeding a <word>.`
`A woman next to a <word>.`
`A <word> with a little girl staring at it.`
`A close-up of a <word>.`

Templates used for the sub-category *negative event*:

`There is a crowd looking scared because of a <word>.`
`Many people are trying to shelter from a <word>.`
`A <word> happening in a big city.`
`A <word> with many people involved.`
`A picture of a <word>.`

Templates used for the sub-category *social event*:

`There is a small crowd attending a <word>.`
`A few people are gathered for a <word>.`
`A <word> attended by a large group of people.`
`A <word> with many people involved.`
`A picture of a <word>.`

Templates used for the sub-category *communication*:

`There is a small crowd at a <word>.`
`A few people are participating in a <word>.`
`A <word> in a crowded room.`
`A <word> with many people involved.`
`A picture of a <word>.`

Templates used for the sub-category *sound*:

```
There is a man hearing a <word>.
A few people seem to hear a <word>.
A <word> is heard by a few people.
A <word> with a few people listening to it.
A picture of a <word>.
```

B Additional RSA Results

B.1 Single-word vs. contextualised representations

Our choice to derive word representations by including them in sentences was guided by the intuition that single words could have been an out-of-distribution input for computational models trained to output contextualised word representations. We empirically verified that representations obtained by embedding words within templates yield higher alignment than those obtained by passing single words to the models. We show the EXP48 and brain alignment obtained with both embedding-extraction procedures in Figure 6.

B.2 Layer-wise RSA results

In the main paper, we reported RSA results calculated from model representations averaged across the three layers yielding the highest alignment individually. Here, we provide a layer-wise visualisation of RSA results, which allows observing how EXP48 vs. brain alignment changes throughout model layers. Specifically, layer-wise Spearman correlations against EXP48 are displayed in Figure 7, while those against fMRI responses are in Figure 8.

B.3 RSA with additional baselines

For completeness, in Table 1 we report RSA results including three additional models: CLIP (Radford et al., 2021), a vision-language model pretrained contrastively on 400M image-caption pairs, and the distributional models GloVE (Pennington et al., 2014) and Word2vec (Mikolov et al., 2013). The distributional models were originally included in Fernandino et al. (2022); note that the brain correlations we report differ from the ones from Fernandino et al. (2022), as they computed an average across participant-wise brain correlations, while we averaged brain RDMs across participants *before* computing correlations.

Model	ρ EXP48	ρ *Brain*
SimCSE	0.52	0.22
MCSE	0.45	0.19
CLAP	0.03	0.00
BERT	0.53	0.23
VisualBERT	0.27	0.12
CLIP	0.41	0.14
GloVe	0.45	0.14
Word2vec	0.42	0.125

Table 1: Spearman correlations quantifying the alignment of models' representational spaces with EXP48 and brain responses.

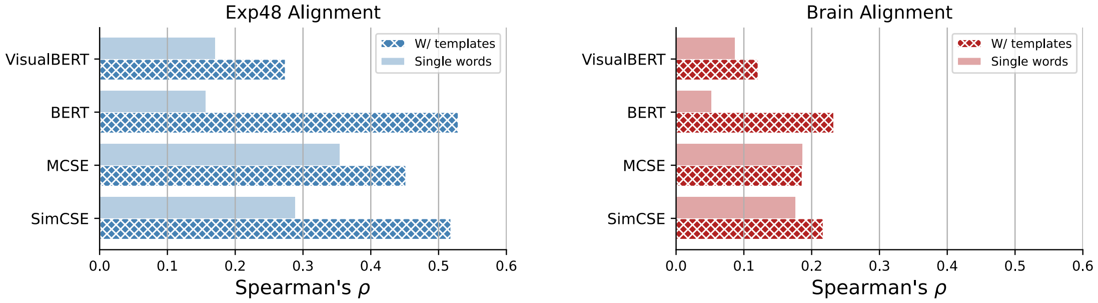


Figure 6: Spearman correlations observed from model representations obtained by passing single words vs. words embedded in templates. The left-hand panel shows the alignment with EXP48 and the right-hand one with brain responses.

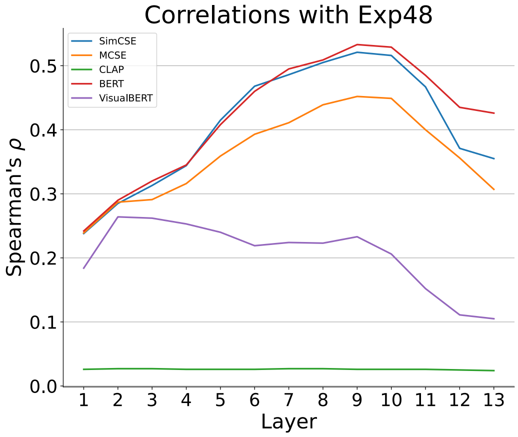


Figure 7: Spearman correlations indicating how representational similarity between model representations and EXP48 representations changes along model layers.

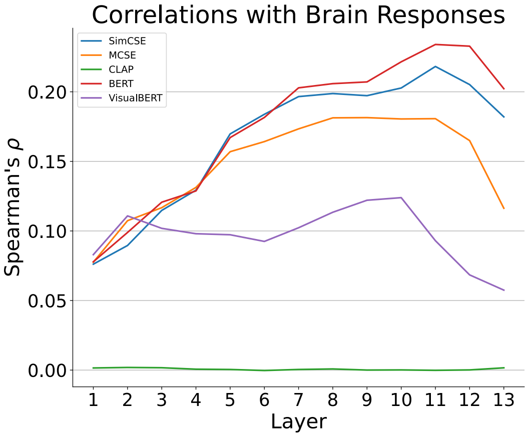


Figure 8: Spearman correlations indicating how representational similarity between model representations and brain responses changes along model layers.

What is an "Abstract Reasoner"? Revisiting Experiments and Arguments about Large Language Models

Tian Yun
Brown University
tian_yun@brown.edu

Chen Sun
Brown University
chensun@brown.edu

Ellie Pavlick
Brown University
ellie_pavlick@brown.edu

Abstract

Recent work has argued that large language models (LLMs) are not "abstract reasoners", citing their poor zero-shot performance on a variety of challenging tasks as evidence. We revisit these experiments in order to add nuance to the claim. First, we show that while LLMs indeed perform poorly in a zero-shot setting, even tuning a small subset of parameters for input encoding can enable near-perfect performance. However, we also show that this finetuning does not necessarily transfer across datasets. We take this collection of empirical results as an invitation to (re-)open the discussion of what it means to be an "abstract reasoner", and why it matters whether LLMs fit the bill.[1]

1 Introduction

The question of whether large language models (LLMs) are "abstract reasoners" has been the frequent subject of recent work, both directly (Hu et al., 2023; Webb et al., 2023; Gendron et al., 2024; Musker et al., 2024) and indirectly (Chollet, 2019; Mitchell et al., 2023; Moskvichev et al., 2023). The answer to this question feels weighty. LLMs currently dominate modern approaches to AI, and abstract reasoning is arguably the linchpin of general and flexible intelligence (Gentner et al., 2001; Han et al., 2024; Mitchell, 2021). If LLMs are not abstract reasoners, it follows that fundamental changes are needed in how AI is developed.

The challenge with this question is that there is little consensus on what it means to be an "abstract reasoner", and what evidence would convincingly demonstrate that an LLM, or any model, is or is not one. Most recently, it has been argued that LLMs are not abstract reasoners on the basis of their poor performance when tested out-of-the-box on adapted visual, analogical, and quantitative reasoning tasks (Figure B.1 for examples) that require models to infer and generalize patterns from a limited number of observations (Gendron et al., 2024; Mitchell et al., 2023; Stevenson et al., 2024). In this work, we revisit this experimental setup. We replicate the results of earlier studies, but add additional experiments which demonstrate the need for more nuance before drawing strong conclusions.

Specifically, we follow the experimental design from Gendron et al. (2024), and replicate their finding that off-the-shelf pretrained LLMs perform badly across a range of challenging reasoning tasks (§4). However, we find that optimizing just the embedding layer for the task (leaving the transformer blocks frozen) all but eliminates the problem, allowing the model to perform comparably to finetuning the entire model, and sometimes even solve the task perfectly (§5). This result extends beyond simple embeddings and, in fact, a frozen pre-trained LLM can perform well on visual reasoning tasks as long as the visual encoder is fine-tuned on in-domain task data (§6).

Together, these results paint a more subtle picture of LLMs: much of their representations and inferential capabilities appear to be transferable across very diverse tasks, but non-trivial effort is required on the input side for each new task in order to harness these capabilities. In light of this, we (re-)open the larger discussion which is simultaneously empirical and philosophical (§7): What does it mean to be an abstract reasoner, and why do we care whether LLMs fit the bill?

2 Related Work

2.1 Analogical Reasoning

Prior work has studied the question of abstract reasoning of LLMs via analogical reasoning, such as matrix reasoning (Webb et al., 2023), letter-string analogies (Mitchell, 2021; Hofstadter et al., 1995) and pointer-value retrieval (Zhang et al., 2021b). These analogical reasoning benchmarks require a

---

[1] Code and resources are available at: `https://github.com/tttyuntian/abstract_reasoner_llm`

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 156–168
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

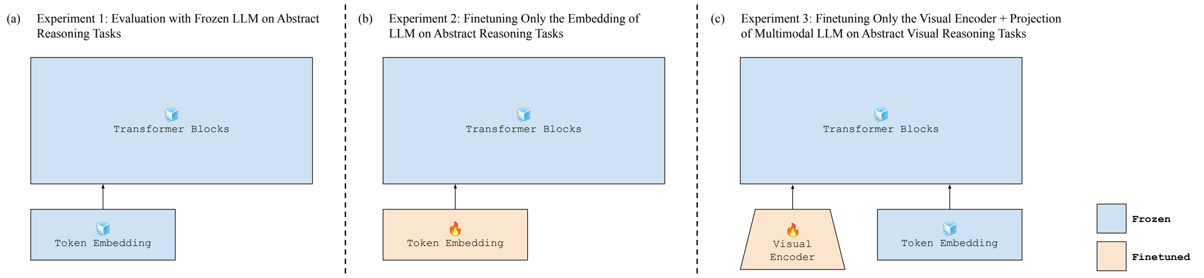

(a) Experiment 1: Evaluation with Frozen LLM on Abstract Reasoning Tasks
Transformer Blocks
Token Embedding
(b) Experiment 2: Finetuning Only the Embedding of LLM on Abstract Reasoning Tasks
Transformer Blocks
Token Embedding
(c) Experiment 3: Finetuning Only the Visual Encoder + Projection of Multimodal LLM on Abstract Visual Reasoning Tasks
Transformer Blocks
Visual Encoder
Token Embedding
Frozen
Finetuned

Figure 1: Illustration of our experimental settings. In Setting (a), we freeze the whole LLM and run evaluations. This is treated as language baseline when image captions are inputs on abstract visual reasoning tasks. In Settings (b) and (c), we freeze the pretrained transformer blocks and finetune only the input layers (i.e., token embedding layer and visual encoder). In Setting (c), we freeze the token embedding layer to study the impact of tuning the visual encoder in a controlled setting. Note that the inputs are pure language in Settings (a) and (b), while the inputs are language prompts with image representations in Setting (c).

model to infer the patterns from a limited number of observations and apply the discovered patterns to the new queries.

Despite the impressive performance of LLMs, there is yet no consensus on whether LLMs are strong analogical reasoners. Some studies show evidence suggesting that LLMs can even surpass the human baseline on analogical reasoning tasks (Hu et al., 2023; Webb et al., 2023), while the others show that LLMs achieve very limited performance on a set of analogical reasoning benchmarks (Gendron et al., 2024) or they are not robust to counterfactual examples or irrelevant information (Lewis and Mitchell, 2024; Musker et al., 2024). We use similar tasks and models as the prior work, but incorporate additional tasks and a wider range of finetuning experiments in order to situate the results within a larger discussion about abstract reasoning.

2.2 Visual Analogical Reasoning

Analogical reasoning can go beyond symbols and words and involve visual input, such as in ARC (Chollet, 2019), ACRE (Zhang et al., 2021a), RAVEN (Zhang et al., 2019; Hofstadter et al., 1995) and MEWL (Jiang et al., 2023). Recent approaches on visual analogical reasoning can be categorized into neuro-symbolic methods (Mao et al., 2019; Hudson and Manning, 2019), or neural networks with implicit representations (Ding et al., 2021; Sun et al., 2024; Bhattacharyya et al., 2023). Both approaches roughly follow the same outline of the perception stage and the reasoning stage. The perception stage usually relies on task-specific visual encoders, such as symbolic object encoders (Zhang et al., 2021a), object detectors (Ding et al., 2021), or on task-specific training strategies for these visual encoders (Sun et al., 2024; Bhattacharyya et al., 2023). The reasoning stage introduces inductive biases by developing task-specific reasoning modules (Hu et al., 2021b; Benny et al., 2021). In this work, we investigate if the transformer blocks of a pretrained LLM can be used as a reasoner for different visual analogical reasoning tasks.

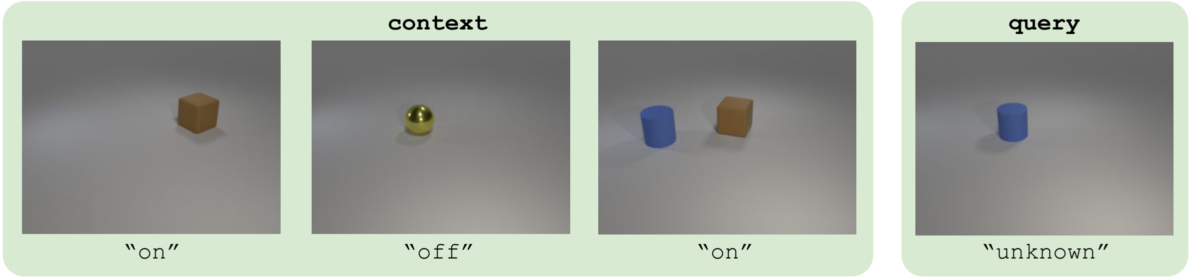
2.3 Multimodal Large Language Models

Prior work shows that transformer blocks pretrained on natural language can be transferred to non-language sequence modeling problems by optimizing new input and output layers (Lu et al., 2022). With the rise of LLMs, recent work freezes pretrained vision models and pretrained LLMs, and only learns a mapping to project visual representations to language latent space in order to perform on multimodal tasks (Merullo et al., 2023; Liu et al., 2023; Li et al., 2023; Liu et al., 2024). Tong et al. (2025) investigates the impact of vision-only models in multimodal LLMs and reaches impressive performance on downstream tasks. Our work is similar to these models, but connects it to a larger, more philosophical debate about the meaning of "abstract reasoning".

3 Datasets

3.1 Reasoning Tasks from Gendron et al.

We follow the evaluation benchmark used by Gendron et al. (2024) to quantitatively measure the so-called "abstract reasoning" capabilities of language models. This benchmark contains seven tasks, each of which evaluates the ability of a model to infer patterns from a limited number of examples. These seven tasks can be divided into two categories: open question answering (OPQA) and multiple-



(a) Textual reasoning task

> You are a helpful assistant that determines whether the light will be activated by the objects. Some objects can activate the light. The other objects cannot activate the light. There are three possible light states: on, off, and unknown.
> Input: there is a brown cube. Light: on.
> Input: there is a yellow sphere. Light: off.
> Input: there is a brown cube and a blue cylinder. Light: on.
> Input: there is a blue cylinder. Light: unknown.

(b) Visual reasoning task

> You are a helpful assistant that determines whether the light will be activated by the objects. Some objects can activate the light. The other objects cannot activate the light. There are three possible light states: on, off, and unknown.
> Input: `[context_image_1_representation]` Light: on.
> Input: `[context_image_2_representation]` Light: off.
> Input: `[context_image_3_representation]` Light: on.
> Input: `[query_image_1_representation]` Light: unknown.

Figure 2: Illustration of the use of language models for text-based and image-based versions of ACRE. Each data example will be formulated into a prompt for an LLM to make a prediction for the query. In textual reasoning task, each context frame is represented by a frame caption. In visual reasoning tasks, each context frame is represented by an encoded frame representation.

choice question answering (MCQA). OPQA tasks require a model to generate the correct answer, while MPQA tasks require a model to select the correct answer from the given set of answer candidates. OPQA tasks include Abstract Reasoning Challenge (ARC) (Chollet, 2019), BIG-Bench dataset (BBF) (Rule, 2020; Srivastava et al., 2022), Evals-P (Achiam et al., 2023), and Pointer-Value Retrieval (PVR) (Zhang et al., 2021b). MCQA tasks include ACRE^T (Zhang et al., 2021a), RAVEN^T [2](Zhang et al., 2019), and Evals-P (Achiam et al., 2023). For ACRE^T and RAVEN^T, we also consider ACRE^T-Symb and RAVEN^T-Symb, where the panel descriptions are converted into symbols (e.g., using integers to represent different objects).

3.2 Additional (Visual) Reasoning Tasks

In addition to the models and tasks considered by Gendron et al. (2024), we additionally consider how well LLM representations transfer fo the multimodal language model framework (MLLM). To support these experiments, we consider two visual reasoning tasks: ACRE (Zhang et al., 2021a) and MEWL (Jiang et al., 2023). In ACRE, given the 5 context frames and 1 query frame, a model needs to predict the activation status of Blicket detector in the query frame, which can be *on*, *off*, or *unknown*. In MEWL, given 6 context frames and 1 query frame, a model needs to understand the meaning of the novel words and select the correct utterance out of 5 options for the query frame.

4 Frozen Pretrained LLMs

We first seek to replicate Gendron et al. (2024)'s finding that frozen pretrained LLMs achieve low performance across a large suite of reasoning tasks. We reproduce these evaluations on LLaMA2 with 7 billion parameters (Touvron et al., 2023). Table 1 shows the results on OPQA and MCQA tasks. We observe that even though there are small gaps

---

[2] ACRE^T and RAVEN^T are text-based version of the original tasks.

	OPQA						MCQA				
	ARC	BBF	Evals-S	PVR	RAVEN^T	RAVEN^T-Symb	ACRE^T-Text	ACRE^T-Symb	Evals-P	RAVEN^T	RAVEN^T-Symb
Random	-	-	-	-	-	-	33.3	33.3	50.0	12.5	12.5
LLaMA2-7b-chat (NZ)	0.5	10.8	0.0	0.0	0.0	0.1	1.4	0.3	50.0	2.6	14.9
LLaMA2-7b-chat (Ours)	1.0	26.4	0.0	21.8	0.0	1.0	26.4	38.1	52.0	12.9	11.4

Table 1: Performance of frozen pretrained LLMs on open question answering (OPQA) and multiple-choice question answering (MCQA) benchmarks. We show the results `LLaMA2-7b-chat(NZ)` reported in Gendron et al. (2024) and our reproduced results `(Ours)` following the evaluation from Gendron et al. (2024).

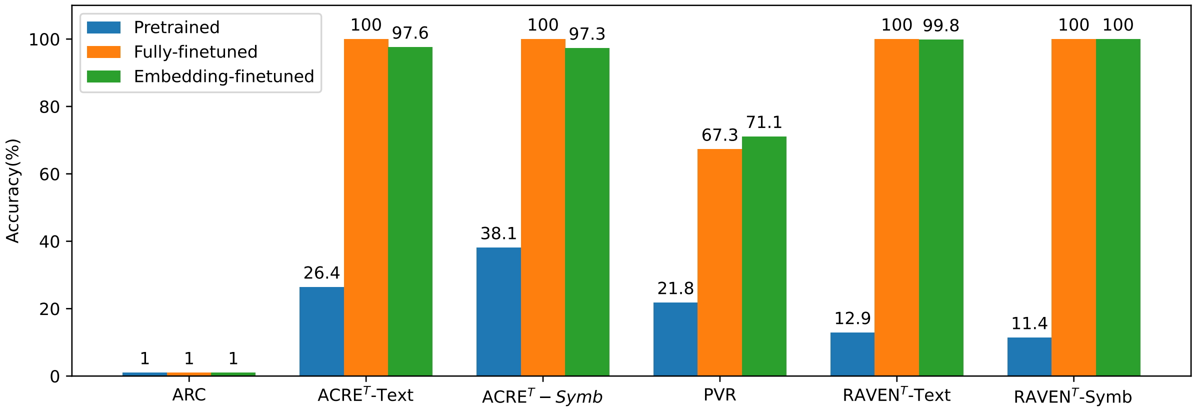


Figure 3: Performance of finetuned LLMs on OPQA (ARC and PVR) and MCQA (ACRE^T and RAVEN^T) benchmarks. LLMs with finetuned embedding layer perform significantly better than their pretrained counterparts, and perform on par with or even surpass the fully finetuned LLMs with LORA. Note that ACRE^T and RAVEN^T are text-based version of original datasets, which may make the tasks easier to solve.

between the original results and the reproduced results, the performance of the pretrained LLMs are still low. Even when the answer candidates are provided in MCQA tasks, the models mostly perform as poor as random baselines (e.g., 33.3% on ACRE and 12.5% on RAVEN). We observe significant gaps between original results and ours on BBF and PVR, and attribute them to the choice of parser used to process the model's predictions.

Overall, our results are, if anything, stronger than what has been previously reported in this evaluation setting. But even so, it is hard to argue that these numbers represent "strong" performance. We thus agree with Gendron et al. (2024) that these results indicate poor transfer ability. What requires additional investigation, however, is whether this poor transfer is interpretable as a lack of abstract reasoning ability.

5 Finetuned Embedding Layers

Given that pretrained LLMs perform poorly off-the-shelf, it is natural to ask whether they can be adapted to these task, and if so, just how much adaptation is necessary. We explore two ways to finetune the LLMs: (1) finetuning all layers with low-rank adaptation (LoRA) (Hu et al., 2021a); (2) finetunning only the embedding layer of the LLMs. LoRA finetuning has become a standard way of adapting a model to a task and represents an upper bound on how well the model could be made to perform the task under the most permissive conditions. In contrast, finetuning just the embedding layer represents a conceptually different type of transfer with respect to the question of this paper. Namely, finetuning just the embeddings is analogous to changing just the input to the system–e.g., ensuring the input is in the format the system expects–but leaving the system itself unchanged (see additional discussion and qualifications about this analogy in §7).

We finetune the embedding layer for 50 epochs using AdamW optimizer (Loshchilov and Hutter, 2019) with early stopping based on the validation set. Following Gendron et al. (2024), we conduct experiments on 2 OPQA tasks (ARC, PVR) and 2 MCQA tasks (ACRE^T, RAVEN^T-mcqa[3].).

Figure 3 shows the results of finetuned LLaMA2. We observe that LoRA-finetuned models perform significantly better than their pretrained counter-

---

[3] We are aware of the defects of RAVEN, and we use the original RAVEN since it was previously used by Gendron et al. (2024)

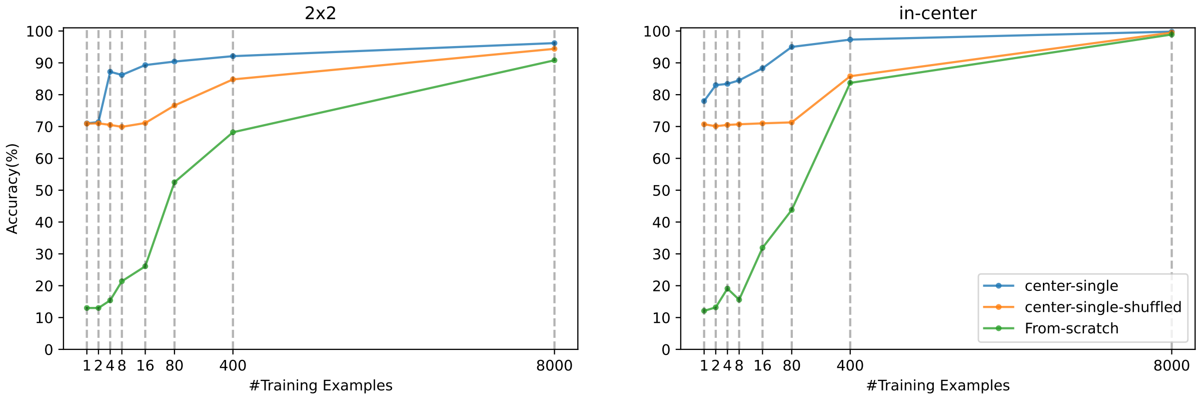


Figure 4: Data efficiency analyses on LLaMA2-7b with token embedding layer finetuned on `center-single` or `center-single-shuffled` and further finetuned on `2x2` and `in-center` tasks in RAVEN with limited amount of data. Y-axis (`#Training Examples`) represents the absolute number of examples used for finetuning. `From-scratch` means the token embedding of a pretrained LLaMA2-7b is directly finetuned on `2x2` and `in-center` tasks. Given that there are 8000 training examples in total, we observe that after finetuning on `center-single`, the model becomes significantly more data efficient. By comparing `center-single` and `center-single-shuffled` lines, we observe that data efficiency of the model mainly comes from the occurrences of task-relevant tokens, rather than the reasoning logic of the tasks.

parts, and can even solve ACRE^T and RAVEN^T perfectly. Moreover, LLaMA2 with a finetuned embedding layer can perform on par with the LoRA-finetuned LLaMA2[4].

Generalizability and Data Efficiency. We conduct experiments on RAVEN^T to further look into two properties of the finetuned input layers: generalizability and data efficiency. An ideal abstract reasoner is expected to generalize to novel tasks with limited amount of observations.

We take LLaMA2-7b's token embedding layer finetuned on one task variant (namely, the `center-single` task) and further finetune this layer with varying amount of training examples for 500 steps on two different task variants (`2x2` and `in-center`), both of which require reasoning over more attributes (e.g., different object alignments). Figure B.5 shows examples of these tasks. We consider three settings: (1) "`center-single`", where the token embedding has been finetuned on the original `center-single` task; (2) "`center-single-shuffled`", where the token embedding has been finetuned on `center-single` task with randomly shuffled labels. This setting preserves the visual features, but breaks the logical "reasoning" structure of the task, and thus serves as a test of how much of the positive transfer is due to low-level visual cues vs. higher-level more abstract features; (3) "`from-scratch`", where the token embedding of a pretrained LLaMA2-7b is directly finetuned on `2x2` and `in-center` tasks. We use this to study the impact of finetuning on `center-single` task.

Figure 4 shows the results. LLaMA2-7b with token embeddings finetuned just on 80 examples can perform competitively aganist LLaMA2-7b directly finetuned on full dataset (8k examples) of the tasks. The fairly small gap between the `center-single` and `center-single-shuffled` lines suggests that the positive transfer is primarily explained by the lower-level visual features rather than the reasoning logic of the tasks.

6 Visual Encoder Trained from Scratch

Prior work has shown that transformer blocks pretrained on natural language can be tranferred to non-language sequence modeling tasks, such as image recognition and protein fold prediction (Lu et al., 2022). Given the surprising effectiveness of finetuning just the embedding layer of LLaMA2 on text-only abstract reasoning tasks, we hypothesize that the frozen transformer blocks of a pretrained LLM will perform well on abstract visual reasoning tasks if the visual encoder is tuned for the task. That is, we follow the multimodal LLM framework (MLLM) which consists of a visual backbone, a language backbone, and a linear projection layer which maps visual representations to language latent space. We keep the transformer blocks and

---

[4] We attribute the low performance of ARC to its complexity and the length of each data sequence (excluding the expected answer), where 75% of data has >2000 tokens.

	Method	I.I.D.	Compositional	Systematic
Language Baseline	LLaMA2-7b	26.4	26.1	29.9
	GPT-4	66.4	66.4	64.0
	GPT-4-Turbo	69.7	69.9	67.4
Existing Approaches	NS-OPT	66.3	69.0	67.4
	ALOE	-	91.8	93.9
	IV-CL	93.0	93.2	92.6
	LRR	-	**98.2**	**99.2**
	LLaVA-NeXT-Mistral-7B	38.4	36.9	36.9
	GPT-4o	62.6	61.5	61.7
Ours	LLaMA2-7b-Object	**95.5**	97.5	86.5

(a) ACRE

	Method	`shape`	`color`	`material`	`object`	`composite`	`relation`	`bootstrap`	`number`	`pragmatic`	Avg.
Language baselines	LLaMA2-7b	49.7	61.2	52.5	73.8	35.2	19.2	29.5	21.8	22.2	40.6
	BERT*	94.8	98.8	97.5	19.5	**97.8**	**22.2**	62.2	21.8	**99.8**	68.3
	GPT-3.5	**96.8**	82.3	87.0	**98.2**	88.3	20.0	45.8	22.7	26.7	63.1
Existing Approaches	ALOE	34.2	33.2	31.0	19.5	30.5	21.5	27.5	23.3	20.8	26.8
	Flamingo-1.1B	49.3	35.3	48.5	19.2	38.2	18.8	57.3	84.2	18.0	41.0
Ours	LLaMA2-7b-Object	59.3	**100.0**	**98.8**	96.8	50.4	17.3	**87.0**	**99.5**	19.2	**69.8**

(b) MEWL

Table 2: Results of LLaMA2-7b with train-from-scratch visual encoders on sub-tasks in ACRE and MEWL. **Bolded results** are the best results, and underlined ones are the second best. All language baselines are frozen, except BERT which is finetuned on MEWL tasks. The results show that frozen LLaMA2 with learned visual encoder perform significantly better than its language counterpart and even outperform the existing approaches.

the token embedding layer of language backbone frozen, and only train the visual encoder and the projection layer. If this MLLM with a trained visual encoder can perform better than its language backbone with oracle visual perception, then it provides further evidence for the above interpretation of the frozen LLM as a highly transferable system.

6.1 Variants of Image Inputs

In order to run these experiments, we consider three variants of image inputs. Figure B.3 shows the examples of each variant.

Symbol. A frame is represented by a set of multi-hot object representations, where each object representation is the concatenation of its one-hot vectors for object attributes (i.e., color, material, and shape) and a vector of object location information. This mimics the experiments in §5 by assuming oracle visual perception, and allows us to directly contrast language and visual inputs.

Object. A frame is represented by object representations, where each object is an object crop from the frame. This variant assumes ground truth object detection exists in order to control the factors of reasoning performance.

Image. A frame is represented by its RGB image. This variant simplifies the inputs the most, but requires the visual encoder to encode object properties and spatial relationships between objects directly from the frames.

6.2 Language Baseline

For our language baseline, we provide a frozen LLM directly with language descriptions of the abstract visual reasoning problem. Frame captions can be considered as oracle visual perception, where each frame is represented by its caption (e.g., "There is a blue cylinder and a brown cube.").

6.3 Implementation Details

On ACRE, we use the training set with 6K samples, where each sample contains 6 context frames and 4 query frames. Thus, the training set has 24K sequences. On MEWL, we use the training sets of the 9 sub-tasks, each of which involves 600 samples. Thus, the training set has 5400 sequences.

For the language backbone, we use LLaMA2 with 7 billion parameters (Touvron et al., 2023). For the visual backbone, to encode image inputs, we use a 2-layer ViT (Dosovitskiy, 2020) with 4 attention heads and 768-hidden dimensional space; to encode symbolic representations of images, we use a symbolic encoder which encodes object at-

	ACRE				MEWL									
	I.I.D.	Comp.	Sys.	Avg.	`shape`	`color`	`material`	`object`	`composite`	`relation`	`bootstrap`	`number`	`pragmatic`	Avg.
LLaMA2-7b-Image	75.8	77.7	71.7	75.1	35.0	99.8	57.7	26.2	32.7	19.8	31.8	45.2	21.3	41.1
LLaMA2-7b-Object	95.5	97.5	86.5	93.2	59.3	100.0	98.8	96.8	50.4	17.3	87.0	99.5	19.2	69.8
LLaMA2-7b-Symbol (Linear)	91.0	94.9	86.8	90.9	100.0	99.8	100.0	98.0	42.5	18.0	35.0	78.2	18.3	65.5
LLaMA2-7b-Symbol (MLP)	98.3	99.5	84.6	94.1	100.0	100.0	100.0	98.8	71.3	16.2	91.3	99.7	22.3	77.7

Table 3: Analysis on the presence of object-centric information. -Symbol rows can be considered as upper bound, since the inputs are symbolic representations of images. The performance gap between -Image and -Object reflects the importance of object-centric inductive bias in abstract visual reasoning tasks.

tributes with embedding layers and encodes objects' location information[5] with a linear layer.

During finetuning, we freeze the language backbone and finetune the visual encoder and the linear projection. We use the AdamW optimizer with a learning rate of 3×10^{-5}. We finetune the visual backbone for 20 epochs on ACRE, and 40 epochs on MEWL. The batch size is set to 64.

6.4 Results

Table 2 shows the results of LLaMA2-7b with learned visual encoders on ACRE and MEWL. On ACRE, we observe that LLaMA2 with train-from-scratch visual encoders can perform significantly better than their language-only counterpart. These models can even outperform majority of the multimodal state-of-the-art, including IV-CL (Sun et al., 2024) and LRR (Bhattacharyya et al., 2023), which are pretrained with video data. On MEWL, we observe the same pattern that LLaMA2 with learned visual encoders can outperform prior state-of-the-art and also the language baselines which assume perfect visual perception.

In Table 3, we further investigate different ways to represent an image. The large performance gap between LLaMA2-7b-Image and -Object (e.g., average of 41.1% versus 69.8% on MEWL), indicating that object-centric information is important for the pretrained transformer blocks to better solve abstract visual reasoning tasks. In all, these results demonstrate that with a frozen language backbone, learning just the visual encoder from scratch can already improve the model's performance on abstract visual reasoning tasks significantly. However, task-specific design choices, such as object-centric representations, would be needed.

7 Discussion

The question of whether LLMs are "abstract reasoners" has consequences for how we understand and thus how we develop increasingly advanced artificial intelligence. The challenge is that there is no consensus for what it means to be an "abstract reasoner". In their recent work, Gendron et al. (2024) operationalize abstract reasoning as the ability to transfer zero-shot to a range of complex reasoning tasks. They find that LLMs perform poorly on this evaluation, and thus conclude that they are not abstract reasoners.

In this work, we reproduce Gendron et al. (2024)'s findings, but push back against their interpretation. In particular, we provide new experiments which show that tuning just the embedding layer is remarkably effective. Indeed, across a variety of textual and multimodal tasks, frozen pretrained LLMs can achieve high levels of performance as long as the input representations are adapted sufficiently for each task [6].

It seems too stringent a criteria to require that that abstract reasoners perform arbitrary tasks on arbitrary inputs without adaptation. By way of counterargument, consider the good old fashioned AI (GOFAI) systems of the 1990s, which typically included symbolic systems internally, e.g., databases implemented in SQL or rules for logical inference implemented in PROLOG. By most intuitive definitions, these databases and rules would be considered "abstract" and the tasks the systems performed over them would be "reasoning". But we would not expect these systems to operate well over a database implemented in MongoDB, or to apply rules defined by Python. Rather, the need to operate on representations of a particular format is a consequence of, not an exception to, the system's abstraction.

---

[5] Each object location is represented as $[x_1, y_1, x_2, y_2, w, h, w \times h]$

[6] While we argue that input-level finetuning can enable pretrained models to perform well on a range of tasks, we acknowledge that this does not necessarily imply the models have acquired generalized abstract reasoning in a cognitive sense. Rather, it may reflect the alignment of input representations with the pretrained model's existing capabilities. A more robust theoretical framework would be needed to precisely distinguish between mere representational alignment and true abstraction across domains and tasks.

Of course, we don't claim that the internal processing of an LLM is exactly analogous to that of a GOFAI system. Of course, in an LLM, tuning the input embedding layer might do more than simply "rerepresent", but rather might encode some task-specific processing as well. But interpreted loosely, the analogy is useful for highlighting how the question of adaptability and transferability relates to the question of abstraction and reasoning.

Indeed, this relationship has been considered in depth by philosophers of AI, long before LLMs. For example, Dennett (1997) appeals to transferability in his attempt to describe the difference between human cognition[7] and simpler computational systems:

> Consider the lowly thermostat...we might agree to grant it the capacity for about half a dozen different beliefs...it can believe the room is too cold or too hot, that the boiler is on or off...and so forth...suppose we *de-interpret* its beliefs and desires, it can believe the A is too F or G...and so forth....by attaching the thermostatic control mechanism to different input and output devices, it could be made to regulate the amount of water in a tank, or the speed of a train for instance...But as systems become perceptually richer and behaviorally more versatile, it becomes harder and harder to make substitutions in the actual links of the system to the world without changing the organization of the system itself. ...There comes to be a two-way constraint of growing specificity between the device and the environment. Fix the device in any one state and it demands a very specific environment in which to operate properly (you can no longer switch it easily from regulating temperature to regulating speed or anything else); but at the same time, if you do not *fix* the state it is in, but just plunk it down in a changed environment, its sensory attachments will be sensitive and discriminative enough to respond appropriately to the change...

Although Dennett is not discussing the notion of "abstract reasoners" *per se*, he observes that intelligent systems do not transfer well unless they are allowed to adapt[8]. Indeed, Dennett argues that this is a defining property, one that differentiates human-like intelligence from simpler (albeit perhaps more abstract) systems such as thermostats.

Dennett's argument is relevant here not because LLMs are human-like or even human-level in their reasoning abilities (they are far from it!). Rather, Dennett articulates a position that is implicit in contemporary discussions about LLMs and "abstract reasoning". That is, that we care about how well a system adapts to new environments because adapting well to new environments is a hallmark of intelligence. Indeed, this is often cited explicitly as the motivation for studies of this nature (e.g., "the question of whether or not LLMs can perform human-like reasoning remains open..." (Gendron et al., 2024)). But if evaluating human-likeness or human-levelness is the motivation for studying abstract reasoning, then arguments such as Dennett's provide a compelling case against using zero-shot transfer ability as a relevant metric.

Of course, there is another, more practical, argument for why we might care about whether LLMs are abstract reasoners, which is simply that we want LLMs to transfer well zero-shot to many tasks in order to facilitate easier, cheaper, and more efficient development of systems. Indeed, the thermostat's highly abstract design is a feature, not a bug. This type of hardware abstraction is what allows similar components and control mechanisms to be readily repurposed to support many types of use cases. A "human like" thermostat might be very undesirable.

Thus, before seeking to answer the question of whether LLMs are "abstract reasoners", we must first determine, as a community, why we care. Do we care because we want to understand how human-like they are, or do we care because we want to facilitate more efficient technological progress? Almost certainly, we care about both, but we should not expect the same experiments to bear on both lines of inquiry. Finding clarity around these questions–what is an abstract reasoner and why do we care about building one?–is the essential next step if we are to make progress toward either, or both, goals.

---

[7] Dennet's essay is not about reasoning, but rather about *intentional* systems, or systems that have true "beliefs" about the world and act according to them.

[8] While our experiments adapt the input layer (e.g., token embedding) of a model, adaptation does not have to be limited to the input layers. Indeed, adaptation throughout the model would be consistent with Dennett's argument. A full exploration of this is beyond the scope of this paper, but is an interesting direction for future work.

8 Conclusion

In this paper, we have (re-)opened the discussion of what it means to be an "abstract reasoner", and why it matters whether LLMs are "abstract reasoners". We have offered empirical results showing that off-the-shelf pretrained LLMs indeed perform poorly on reasoning benchmarks in a zero-shot setting. However, on a variety of textual and multimodal reasoning tasks, frozen pretrained LLMs can reach high levels of performance when the input embeddings are tuned. With this collection of empirical results, we argue that there is a need to determine why we care about whether LLMs are "abstract reasoners" before answering this question.

9 Acknowledgement

We would like to thank all reviewers and the area chair for their valuable feedback. We would like to thank Samuel Musker, Calvin Luo, and other members of the SuperLab at Brown University for their discussions and insights. The project depicted is sponsored in part by a Young Faculty Award from the Defense Advanced Research Projects Agency, Grant #D24AP00261. The content of the information does not necessarily reflect the position, or the policy of the government and no official endorsement of this work should be inferred.

References

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

Yaniv Benny, Niv Pekar, and Lior Wolf. 2021. Scale-localized abstract reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12557–12565.

Apratim Bhattacharyya, Sunny Panchal, Reza Pourreza, Mingu Lee, Pulkit Madan, and Roland Memisevic. 2023. Look, remember and reason: Grounded reasoning in videos with language models. In *The Twelfth International Conference on Learning Representations*.

François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*.

Daniel C. Dennett. 1997. True believers:the intentional strategy and why it works. In *Mind Design II: Philosophy, Psychology, and Artificial Intelligence*. The MIT Press.

David Ding, Felix Hill, Adam Santoro, Malcolm Reynolds, and Matt Botvinick. 2021. Attention over learned object embeddings enables complex visual reasoning. *Advances in neural information processing systems*, 34:9112–9124.

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. 2024. Large language models are not strong abstract reasoners. In *IJCAR*.

Dedre Gentner, Keith J Holyoak, and Boicho N Kokinov. 2001. Introduction: The place of analogy in cognition. *The analogical mind: Perspectives from cognitive science*, pages 1–19.

Simon Jerome Han, Keith J Ransom, Andrew Perfors, and Charles Kemp. 2024. Inductive reasoning in humans and large language models. *Cognitive Systems Research*, 83:101155.

Douglas R Hofstadter, Melanie Mitchell, et al. 1995. The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory*, 2:205–267.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. 2021b. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1567–1574.

Xiaoyang Hu, Shane Storks, Richard L Lewis, and Joyce Chai. 2023. In-context analogical reasoning with pre-trained language models. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Drew Hudson and Christopher D Manning. 2019. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32.

Guangyuan Jiang, Manjie Xu, Shiji Xin, Wei Liang, Yujia Peng, Chi Zhang, and Yixin Zhu. 2023. Mewl: Few-shot multimodal word learning with referential uncertainty. In *International Conference on Machine Learning*, pages 15144–15169. PMLR.

Martha Lewis and Melanie Mitchell. 2024. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. 2022. Frozen pretrained transformers as universal computation engines. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 7628–7636.

Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B Tenenbaum, and Jiajun Wu. 2019. The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584*.

Jack Merullo, Louis Castricato, Carsten Eickhoff, and Ellie Pavlick. 2023. Linearly mapping from image to text space. In *The Eleventh International Conference on Learning Representations*.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Melanie Mitchell, Alessandro B Palmarini, and Arseny Moskvichev. 2023. Comparing humans, gpt-4, and gpt-4v on abstraction and reasoning tasks. *arXiv preprint arXiv:2311.09247*.

Arsenii Kirillovich Moskvichev, Victor Vikram Odouard, and Melanie Mitchell. 2023. The conceptarc benchmark: Evaluating understanding and generalization in the arc domain. *Transactions on machine learning research*.

Sam Musker, Alex Duchnowski, Raphaël Millière, and Ellie Pavlick. 2024. Semantic structure-mapping in llm and human analogical reasoning. *arXiv preprint arXiv:2406.13803*.

Joshua Stewart Rule. 2020. *The child as hacker: building more human-like models of learning*. Ph.D. thesis, Massachusetts Institute of Technology.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Claire E Stevenson, Alexandra Pafford, Han LJ van der Maas, and Melanie Mitchell. 2024. Can large language models generalize analogy solving like people can? *arXiv preprint arXiv:2411.02348*.

Chen Sun, Calvin Luo, Xingyi Zhou, Anurag Arnab, and Cordelia Schmid. 2024. Does visual pretraining help end-to-end reasoning? *Advances in Neural Information Processing Systems*, 36.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. 2025. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7(9):1526–1541.

Chi Zhang, Feng Gao, Baoxiong Jia, Yixin Zhu, and Song-Chun Zhu. 2019. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5317–5327.

Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. 2021a. Acre: Abstract causal reasoning beyond covariation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 10643–10653.

Chiyuan Zhang, Maithra Raghu, Jon Kleinberg, and Samy Bengio. 2021b. Pointer value retrieval: A new benchmark for understanding the limits of neural network generalization. *arXiv preprint arXiv:2107.12580*.

A Limitations

Since the experiments are compute-intensive, our experiments mainly focus on LLaMA2-7b, but there are many other LLMs trained with different number of parameters, data, or inductive biases. We also only consider one prompt template for each reasoning task, and acknowledge that experimenting with more prompts can provide a more comprehensive evaluation of pretrained LLMs. Last, we use parsers to parse the predictions of models in order to compare with the labels. One alternative approach is the use of other LLMs to compare the predictions with the labels. Some of the above concerns are common challenges for existing evaluation of LLMs. Future research could run evaluations on more LLMs and explore whether the tuning other layers (e.g., output layer, middle layers of transformer blocks) can lead to performance improvement, further proving that LLMs need some amount of task adaptations.

B Additional Figures

We show additional figures to illustrate the reasoning tasks we considered and variants of image inputs.

Pattern	Context	Query

BIG-Bench (BBF)
Reverse of the first three elements and append a “4” at the end.
	`[1, 0, 9, 7, 4, 2, 5, 3, 6, 8] → [9, 0, 1, 4]`
	`[3, 8, 4, 6, 1, 5, 7, 0] → [4, 8, 3, 4]`
	`[5, 4, 7, 2, 9, 3, 8, 1] → [7, 4, 5, 4]`
	`[3, 9, 2, 0, 6, 8, 5, 1, 7] → [2, 9, 3, 4]`
		`[9, 2, 1, 3, 4, 7, 6, 8, 5, 0] →` *`[1, 2, 9, 4]`*

Evals-P
If the first character of the input is in the list, then return “foo”; Otherwise, return “bar”.
	`f, [o, z, a, n, g, e, j, f, i, c, l, u, b] → foo`
	`l, [v, u, f, b, m, y, j, h, n, c, d, a, p] → bar`
	`p, [c, e, s, h, q, o, a, t, k, d, n, l, z] → bar`
	`p, [c, h, m, z, d, v, k, l, j, e, x, p, n] → foo`
		`u, [d, a, x, i, h, v, e, z, r, c, n, y, o] →` *`bar`*

Evals-S
Identify the correspondence between each digit and word.
	`13, 17, 1, 6 → Brown,White,Purple,Blue`
	`1, 9, 6, 11 → Purple,Brown,Blue,White`
	`13, 2, 17, 10 → Brown,Purple,White,Blue`
		`5, 9, 2, 11 →` *`Blue,Brown,Purple,White`*

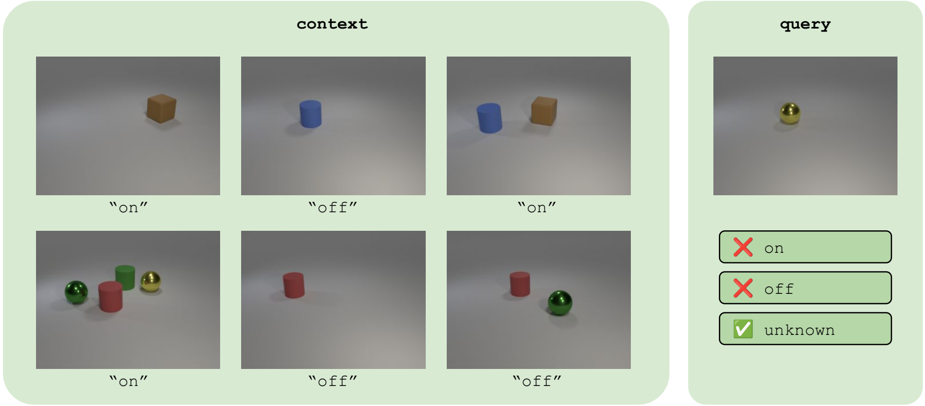
Pointer-Value Retrieval (PVR)
The first element indicates the index of the expected output in the remaining list (i.e., ignore the first element).
	`[5, 7, 4, 1, 8, 9, 8, 1, 9, 8, 4] → 8`
	`[4, 0, 0, 7, 0, 1, 0, 5, 3, 0, 0] → 1`
	`[0, 2, 8, 2, 5, 9, 4, 3, 8, 5, 4] → 2`
	`[3, 3, 2, 6, 5, 7, 4, 6, 7, 4, 8] → 5`
		`[3, 4, 9, 7, 1, 8, 7, 1, 0, 3, 5] →` *`1`*

ACRE
Determine whether the query object will activate the light.
	`A cyan cylinder in rubber is visible. The light is on.`
	`A gray cube in rubber is visible. The light is off.`
	`A cyan cylinder in rubber is visible. A gray cube in rubber is visible. The light is on.`
	`A blue cube in metal is visible. The light is off.`
	`A gray cylinder in rubber is visible. A gray cube in metal is visible. The light is off.`
	`A red sphere in metal is visible. A yellow cube in rubber is visible. The light is on.`
		`A red sphere in metal is visible. The light is` *`undetermined`*`.`

RAVEN
Find and infer the last pattern from the given context.
	`1. On an image, a large lime square rotated at 180 degrees.`
	`2. On an image, a medium lime square rotated at 180 degrees.`
	`3. On an image, a huge lime square rotated at 180 degrees.`
	`4. On an image, a huge yellow circle rotated at 0 degrees.`
	`5. On an image, a large yellow circle rotated at 0 degrees.`
	`6. On an image, a medium yellow circle rotated at 0 degrees.`
	`7. On an image, a medium white hexagon rotated at-90 degrees.`
	`8. On an image, a huge white hexagon rotated at-90 degrees.`
		`The pattern that logically follows is: 9.` *`On an image, a large white hexagon rotated at-90 degrees.`*

Figure B.1: Data examples of abstract reasoning tasks.

(a) ACRE



(b) MEWL

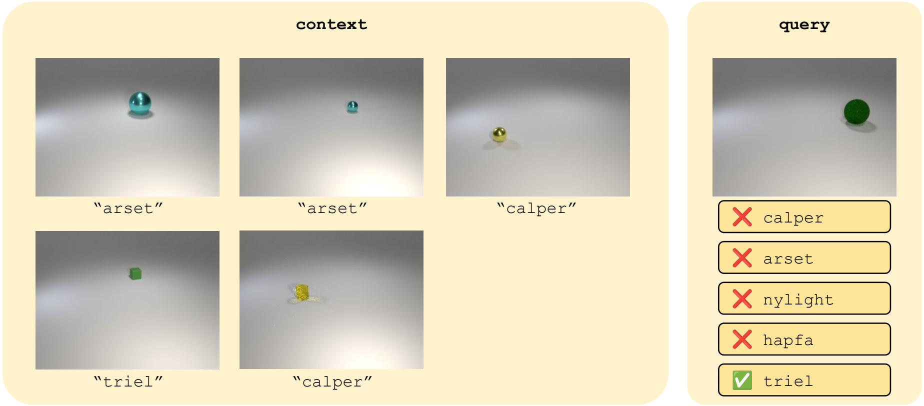


Figure B.2: Data examples of abstract visual reasoning tasks.

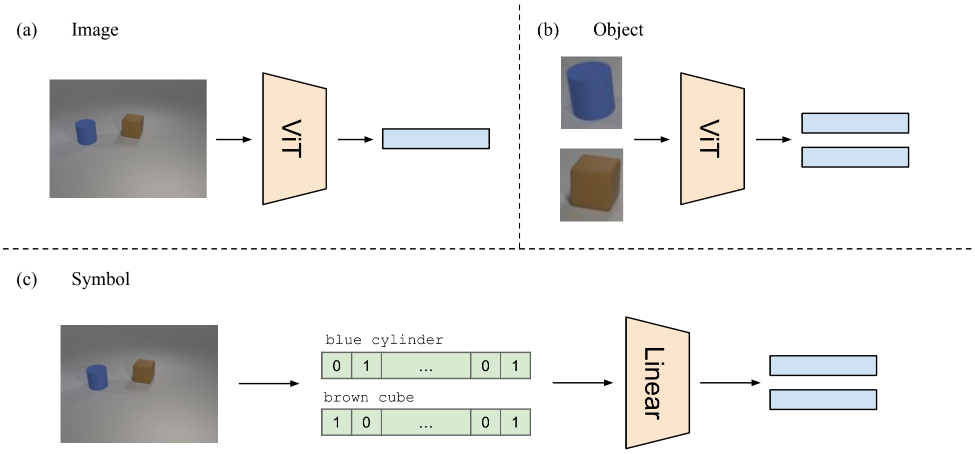


Figure B.3: Examples of variants of image inputs. (a) An image is directly fed into a ViT and obtain an image representation. (b) Each object crop is fed into a ViT and obtain an object representation. (c) Each object is parsed into a multi-hot vector, and a linear layer will output a corresponding object representation.

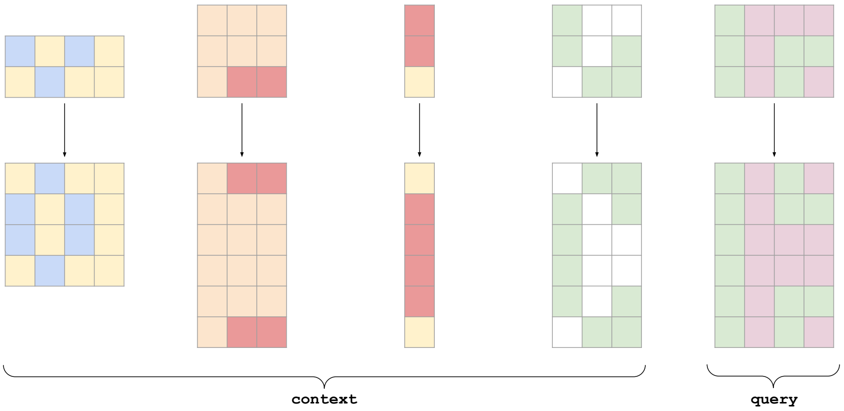
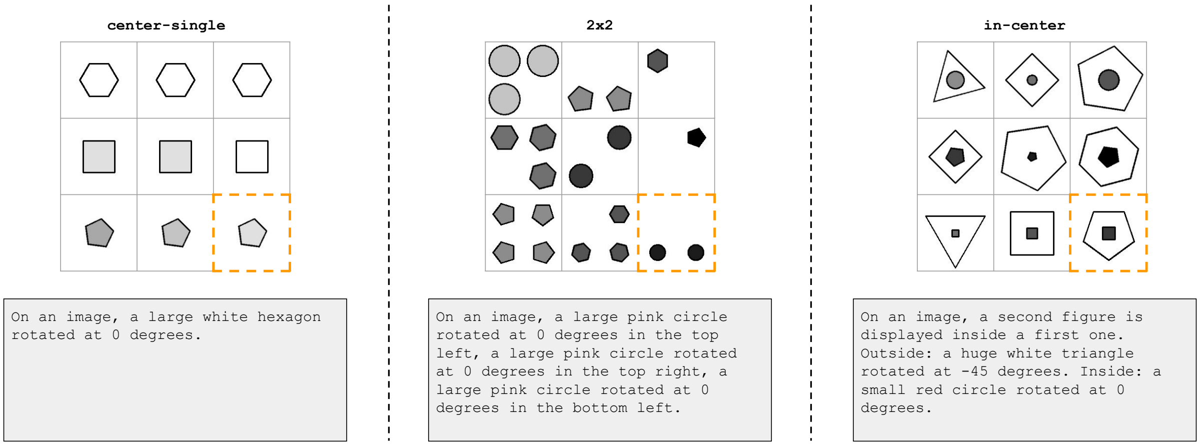


Figure B.4: Example of ARC dataset. There are 4 context examples and 1 query, where each example has an input grid (top) and an output grid (bottom). Each grid is represented as an integer array, where each integer refers to a color. In this example, the task is to generate the symmetry of the input grid and stack the symmetry on top of the original input.



On an image, a large white hexagon rotated at 0 degrees.

On an image, a large pink circle rotated at 0 degrees in the top left, a large pink circle rotated at 0 degrees in the top right, a large pink circle rotated at 0 degrees in the bottom left.

On an image, a second figure is displayed inside a first one. Outside: a huge white triangle rotated at -45 degrees. Inside: a small red circle rotated at 0 degrees.

Figure B.5: Examples of RAVEN^T tasks used in generalizability and data efficiency analysis. Top shows the data example, and bottom shows the language description of the first frame in each example. The task is to fill in the ninth pattern (highlighted in orange) given the eight context frames. We focus on three tasks: `center-single`, `2x2` and `in-center`. `center-single` is the simplest task, since there is always only one object in each frame. `2x2` and `in-center` consider more than one objects in the frames and also involve different object alignments.

Do Construction Distributions Shape Formal Language Learning In German BabyLMs?

Bastian Bunzeck and **Daniel Duran** and **Sina Zarrieß**
CRC 1646 – Linguistic Creativity in Communication
Department of Linguistics
Bielefeld University, Germany
`{firstname.lastname}@uni-bielefeld.de`

Abstract

We analyze the influence of utterance-level construction distributions in German child-directed/child-available speech on the resulting word-level, syntactic and semantic competence (and their underlying learning trajectories) in small LMs, which we train on a novel collection of developmentally plausible language data for German. We find that trajectories are surprisingly robust for markedly different distributions of constructions in the training data, which have little effect on final accuracies and almost no effect on global learning trajectories. While syntax learning benefits from more complex utterances, word-level learning culminates in better scores with more fragmentary utterances. We argue that LMs trained on developmentally plausible data can contribute to debates on how conducive different kinds of linguistic stimuli are to language learning.

Project Gutenberg: Complex sentences

Aber sie war in Angst, dass wir die Larven beschädigen würden, die zu Arbeiterinnen heranwachsen sollten. (*But she was afraid that we would damage the larvae which were supposed to grow into workers.*)

MiniKlexikon: Transitive SP sentences

Der Grafiker entwirft das Bild vorne auf dem Buch. (*The graphic designer designs the picture on the cover of the book.*)
Der Friseur schneidet die Haare. (*The hairdresser cuts the hair.*)

Child-directed speech: Fragmentary utterances

noch mehr! (*even more!*)
ja. (*yes.*)
mit dem Flugzeug. (*with the airplane.*)

Figure 1: Examples for most frequent construction types from different portions of our German BabyLM corpus

1 Introduction

One of the most contentious issues in language acquisition is the relationship between the input that learners receive and the resulting linguistic system (Pullum and Scholz, 2002; Clark and Lappin, 2011). Child-directed speech (or *CDS*) is structurally simple: Especially in the first three years of life, it abounds with questions, imperatives, and fragmentary utterances, but features fewer SV(X) and very few complex sentences, which instantiate "canonical" word order (Cameron-Faulkner et al., 2003). This distribution of utterance-level constructions is conducive to the *functional* side of language acquisition: caregivers talk in this way to elicit responses, steer behavior, or establish joint attention. But how do children acquire full-fledged, *formal* grammatical knowledge from such supposedly skewed input? While its advantages for aspects like speech segmentation or word learning are somewhat accepted (Yurovsky et al., 2012; Cristia et al., 2019), its influence on syntax remains debated: whereas some generativist approaches see any kind of input as too impoverished to learn a full-fledged syntactic system (cf. Chomsky, 1965; Crain and Pietroski, 2001; Guasti, 2002; Thomas, 2002; Berwick et al., 2011), constructionist and usage-based scholars argue that this supposedly skewed input actually aids syntax learning (MacWhinney, 2004; Tomasello, 2005; Bunzeck and Diessel, 2024).

The connectionist "renaissance", fueled by deep learning and Transformer language models, has opened up new avenues of investigating the relationship between an artificial learner's acquired linguistic system and the nature of its training data, more recently also from a constructionist/non-generativist viewpoint (Weissweiler et al., 2023; Piantadosi, 2024). LLMs, pretrained on raw language data only, and instruction-finetuned chatbots based on them, generate text without grammatical errors, and perform well in controlled syntactic test suites. Unfortunately, though, their massive parameter size does not preclude the possibility

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 169–186
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

that their linguistic capabilities result from memorization rather than generalization (Millière, 2024). Furthermore, the sheer amount of their pretraining data exceeds human learner's input by many orders of magnitude, putting their relevance for linguistic modeling into question. Work within the BabyLM community (Warstadt et al., 2023; Hu et al., 2024; Charpentier et al., 2025) has demonstrated that Transformer LMs, trained on cognitively plausible amounts of data, can often acquire fairly complex syntactic structures, even without instruction-finetuning. They can also learn accurate word-level representations when trained with character-level tokenization (Bunzeck and Zarrieß, 2025; Goriely and Buttery, 2025a). This makes them ideal testbeds for the aforementioned issue: does the construction distribution found in CDS, which features a high proportion of questions and syntactic fragments, affect the acquisition of formal linguistic capabilities? In other words, does robust linguistic knowledge at the word and syntax level emerge when the training data is closer to the fragmented, "messy" input of human learners?

The goals of this paper, then, are twofold: (1) we compile a novel German BabyLM training set, for which we conduct the first utterance-level construction analysis for German. We find that distributions align with findings for English and other languages. We then (2) create three 5M-token subsets with distinct constructional profiles, varying, e.g., the proportion of fragmentary and complex utterances, and train small, character-based and subword Llama models on them. We evaluate them with lexical, syntactic, and semantic minimal pairs (Bunzeck et al., 2025; Mueller et al., 2020; He et al., 2025) to gauge the influence of different construction distributions on these levels of linguistic knowledge, and find that differences between grammatically complex training data and a developmentally plausible constructional distribution are fairly small. While certain syntactic phenomena are learned somewhat better from more complex sentences, lexical learning improves with more fragments and questions in the input. Most interestingly, input complexity only modulates the steepness of the resulting learning trajectories, but has no principal effect on the amount of input needed to kickstart learning.

2 Constructions in children's input

Child-directed speech can be seen as a separate linguistic register and is the primary input that children encounter in their first years. On the phonetic level, it features slower speech and exaggerated intonation patterns, which infants prefer listening to (Zangl and Mills, 2007), while its vocabulary is mostly restricted to everyday topics and children's immediate surroundings (Snow and Ferguson, 1977). Structurally, child-directed utterances are usually shorter and simpler than adult-directed ones (Genovese et al., 2020) and feature high amounts of structural and lexical repetition (Tal et al., 2024). Statistical properties of the input directly influence the children's order of acquisition for syntactic patterns (Huttenlocher et al., 2002; Ambridge et al., 2015), e.g., for relative clauses (Diessel and Tomasello, 2000; Brandt et al., 2008; Chen and Shirai, 2015).

Early studies were mostly concerned with mapping out how much CDS is ungrammatical or otherwise "wrong" (in the sense of hesitations, false starts, etc., cf. Pine, 1994), but the quantitative turn in linguistics (Janda, 2013) has enabled more holistic analyses. In a seminal study, Cameron-Faulkner et al. (2003) analyze utterance-level constructions in child-directed English via corpora of toyplay sessions featuring children and caregivers. They show that CDS features only few "canonical" SV(X)-utterances but abounds with questions, lexical fragments, or copula constructions. The reported construction distributions also hold for typologically different languages, e.g., Irish (Cameron-Faulkner and Hickey, 2011). These constructions and their real-world functions help children to quickly understand the *functional* side of language. However, the most common and repetitive utterances that English-speaking children hear represent a rather skewed sample of the presumed, underlying *formal* language system. Generativist approaches would argue that certain formal processes, like question formation from relative clauses, are not attainable from this kind of language, as the input never contains specific examples (Chomsky, 1980) (although Pullum and Scholz, 2002 find that the input frequently contains exactly such specific examples). They also partly emphasize the importance of statistical learning, e.g. for providing hypotheses about competing possible mental grammars constrained by innate, language-specific mechanisms (cf. Yang, 2004, also Ambridge and Lieven, 2011, 121f.). Constructivist approaches do not view language learning as such a re-construction of the target language's abstract grammar, but rather as the re-construction of the target language's inventory of form-meaning pairings (Behrens, 2021). They argue that this kind of input is actually conducive to formal aspects of acquisition, by providing anchor points for first words and their semantic links to real-world reference, which then serve as building blocks for a gradual development into larger schemas (like questions with relative clauses).

Although CDS features such a skewed construction distribution, written language aimed at children, e.g., in children's books, is characterized by a much higher rate of canonical SV(X)-constructions than CDS (Cameron-Faulkner and Noble, 2013). Questions rarely occur in books. CDS produced in shared book reading presents a middle-ground — it contains more complex and SV(X)-constructions than regular CDS, but less than book text alone (Noble et al., 2018). They argue that shared reading therefore, plays an important role in moving children from early, isolated traces of linguistic knowledge to a rich mental language system. This also aligns with the findings by Bunzeck and Diessel (2024), who show that the distribution of constructions in CDS varies with situation type (toy-play features most questions, meal sessions beget more imperatives, shared book reading features more complex constructions) and child age (questions and imperatives become less frequent with age). They suggest that CDS is therefore adapted to support children's cognitive and linguistic development. Yet, as corpus studies are necessarily descriptive and cannot establish causal/mechanistic connections on their own (e.g. what would happen if a child never hears CDS), it remains questionable if this is actually true. Here, the potential of LMs trained on little data becomes apparent for constructionist approaches: they allow controlled experiments with different kinds of input data, which can serve as additional evidence for effects hypothesized from corpus data.

3 Input in developmentally plausible LMs

Authentic data Early approaches to modeling language acquisition with neural networks used hand-picked, manually ordered data points (Rumelhart and McClelland, 1986) or synthetic data generated with hand-crafted grammars (Elman, 1993; Christiansen and Chater, 1999; Chang et al., 2006). Both lack developmental plausibility. Since then, data availability has improved with the establishment of developmental corpora. Frequently, CDS from CHILDES (MacWhinney, 2000) is used to train developmentally plausible LMs (cf. Pannitto and Herbelot, 2020; Huebner et al., 2021). While CHILDES-based models have the advantage of learning from authentic data *only*, they have the disadvantage of not accessing the *full breadth* of the linguistic input children receive. Children are exposed to many more different registers of language throughout their linguistic development, like shared (or solitary) book reading, or television shows (Montag, 2019; Gowenlock et al., 2024). In response to this, the BabyLM corpora propose a data mix of varied spoken and written sources, from CDS over adult-adult conversations to OpenSubtitles (Lison and Tiedemann, 2016), but also children's (Hill et al., 2015) and adults' books (Gerlach and Font-Clos, 2020). All data included in them could be plausibly encountered by children, which provides opportunities to ablate the influence of architecture/training on the learned linguistic knowledge.

For languages other than English, data availability is the greatest problem for the construction of developmentally plausible datasets. Salhan et al. (2024) and Padovani et al. (2025) use only data available from CHILDES for models in different languages, whereas Prévot et al. (2024) compare models trained on spoken data (child-directed + adult-adult conversations) with models trained on the French Wikipedia. As such, these first forays into more polyglot BabyLMs are still constrained to the child-directed input found in CHILDES and do not extend to the aforementioned variety of inputs (Soderstrom, 2007; Gowenlock et al., 2024). Notably, Suozzi et al. (2025) introduce an Italian BabyLM but do not elaborate on their data sources beyond CHILDES.

Linguistic properties The linguistic make-up of pre-training data and its influence on linguistic performance have only recently begun to receive increased scrutiny. Focusing on the lexical level, Yam and Paek (2024) measure sentence-level textual complexity with readability metrics based on text-wide word/syllable–sentence ratios for different corpora (CHILDES, BabyLM corpus, synthetic data, Project Gutenberg). They find that models trained on more complex text perform better at syntactic benchmarks, but simpler data (CHILDES) is learned better in terms of perplexity and loss convergence. Muckatira et al. (2024) filter English pre-training corpora for text spans that only contain vocabulary also found in English CHILDES

Dataset	Description	# Words
CHILDES (MacWhinney, 2000)	Child-directed speech	3,626,301
	Child speech	1,511,144
OpenSubtitles (Lison and Tiedemann, 2016)	Movie subtitles	1,543,094
CallHome (Karins et al., 1997)	Phone conversations	176,313
Klexikon	Children's online encyclopedia	1,384,891
MiniKlexikon	Simplified online encyclopedia	272,886
Wikibooks Wikijunior	Educational books	226,773
Fluter	German youth magazine	2,862,278
Project Gutenberg	Literature (children's and young adult)	2,476,133
Dreambank (Domhoff and Schneider, 2008)	Dream reports	939,197
Leipzig corpus news texts (Goldhahn et al., 2012)	Short news texts	1,541,803
Total		16,560,813

Table 1: Lexical token counts for all subcorpora of our corpus

data and find that simplified models generate more coherent text than models trained on more complex data, and also succeed in syntactic tests if the test data is filtered accordingly. In contrast, Edman et al. (2024) change the semantic content of the pre-training data and use datasets that approximate the linguistic input second-language learners get, e.g., dictionary entries, grammar books, and paraphrases. While grammar books moderately improve syntactic evaluation, there is no positive effect for the addition of the other text types.

Filtered corpora While actual research on the syntactic properties of the input is rather rare, training on filtered corpora has been used in pilot studies. Patil et al. (2024) and Misra and Mahowald (2024) filter out specific grammatical constructions from the BabyLM corpora and then probe the resulting models for knowledge of these grammatical constructions (which might also be analogically learned from related constructions or constructed from their parts). Patil et al. (2024) show that their models succeed on the BLiMP benchmark (Warstadt et al., 2020), even if sentences containing structures targeted in BLiMP's minimal pair sets are removed. Similarly, Misra and Mahowald (2024) show that acceptability scores for the English AANN construction can be reliably estimated from models that have never seen it. In sum, then, models appear to be able to generalize from indirect evidence and learn language in a somewhat constructivist, bottom-up fashion.

The structural composition of child-directed data has (so far) not been scrutinized. Most studies focus on lexical or semantic properties, emphasizing content over structure; child-directed data is usually equated with a somewhat fitting vocabulary or with just being authentic data. However, findings from usage-based linguistics suggest that structural properties, like utterance-level construction distributions, play a crucial role in language acquisition. Whereas Patil et al. (2024) and Misra and Mahowald (2024) remove specific constructions from their data, we aim to explore whether different global distributions of constructions influence the resulting linguistic knowledge and learning trajectories.

4 A German BabyLM dataset

To construct a German dataset, we use a variety of developmentally plausible sources, similar to the English BabyLM data (Warstadt et al., 2023; Choshen et al., 2024). We use (1) all data from German CHILDES corpora (MacWhinney, 2000), including frog stories from TalkBank (Berman and Slobin, 1994) and math lessons from ClassBank (Stigler et al., 2000), (2) subtitles from OpenSubtitles (Lison and Tiedemann, 2016), (3) adult conversations from the CallHome corpus (Karins et al., 1997), and (4) written data from Project Gutenberg, from which we downloaded a manually curated sample of children's books, young adult literature and literature commonly read in German schools. We supply this data with two corpora, the DreamBank database of self-reported dreams (Domhoff and Schneider, 2008) and short news texts from the Leipzig corpus (Goldhahn et al., 2012); although they are not child-directed per se, these sources are child-available in everyday language.

To approximate child-available input even better, we tap into freely available child/learner-directed sources and compile four additional subcorpora for our dataset. The Wikibooks Wikijunior shelve features educational resources aimed at children, focusing on a diverse array of topics such as technology or nature. The Klexikon is a children's wiki in German, featuring more than 3,000 articles aimed

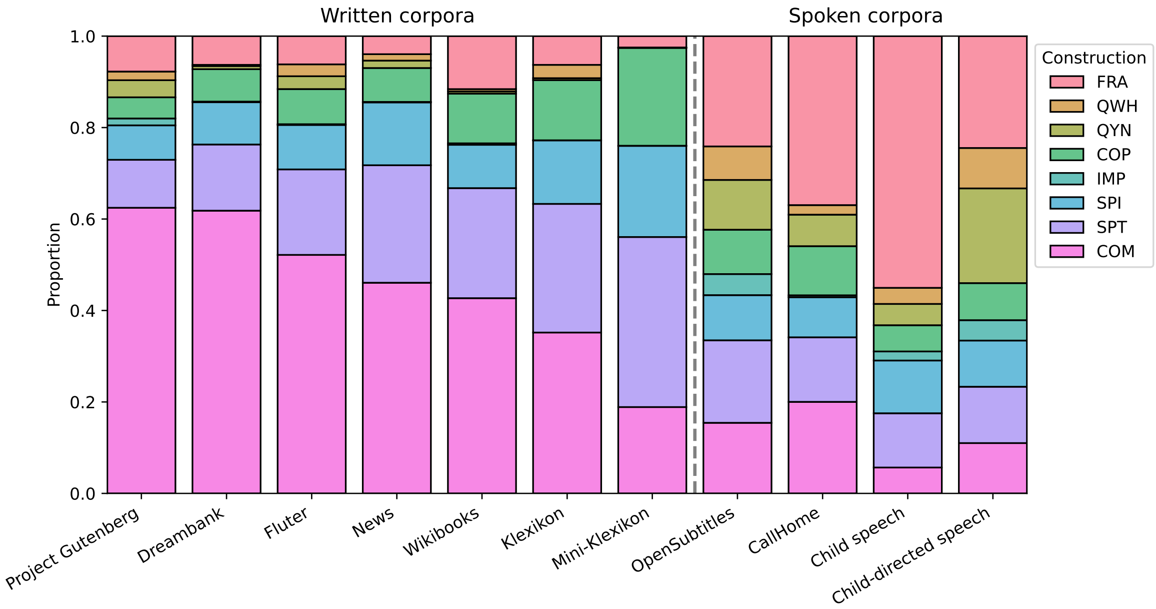


Figure 2: Proportions of utterance-level constructions for all subcorpora in our corpus

at children between 5–15. A simplified version of it is the MiniKlexikon, which features over 1,500 articles aimed at beginning readers. Finally, we also scrape the complete archives of *Fluter*, a magazine aimed at young adults published by the Federal Agency for Civic Education, which contains a large body of non-fiction. All resources are CC-licensed. Table 1 shows the raw token numbers for all corpora (16.5M overall). We extensively clean and normalize our data (details in Appendix B) and make our dataset available on Hugging Face.[1]

5 Construction distribution analysis

As there are no findings on the distribution of utterance-level construction in German, we conduct our own analysis using `spacy` (Honnibal et al., 2020). We first split larger paragraphs into individual sentences with the included senter and then annotate these with POS and dependency information. This information serves as the base of our construction annotation procedure. We devise standard construction categories in line with comparable efforts for English (Cameron-Faulkner et al., 2003; Cameron-Faulkner and Noble, 2013; Bunzeck and Diessel, 2024), and assign one of the following categories to each utterance:

- **FRA** – utterances that do not contain a verb
- **QWH** – wh-question (introduced by interrogative pronouns)
- **QYN** – yes/no-question (introduced by verbs/auxiliaries)
- **COP** – subject-predicate utterance where the predicate is a copula verb (a form of *sein* or *werden*)
- **IMP** – utterances introduced by verbs in imperative mood
- **SPI** – standard subject-predicate utterance (intransitive verb with no direct/accusative object)
- **SPT** – standard subject-predicate utterance (transitive verb with direct/accusative object)
- **COM** – utterances with two or more lexical verbs

This holistic taxonomy is applicable to every utterance in our corpus. For a balanced, manually annotated sample of 1,000 sentences our classifier reaches an accuracy of approx. 95%.

Figure 2 visualizes the results of this annotation process, exact proportions are reproduced in Appendix C. Generally, our results confirm earlier findings (Cameron-Faulkner et al., 2003; Cameron-Faulkner and Hickey, 2011; Cameron-Faulkner and Noble, 2013; Bunzeck and Diessel, 2024): Just like English CDS, German CDS features more questions than any other corpus, abounds with fragments, and contains comparatively few complex utterances. The Project Gutenberg data, on the other hand, is characterized by over 60% complex sentences. Interestingly, the construction distribution forms a continuum across our subcorpora. The MiniKlexikon, for example, contains considerably

---

[1] `https://huggingface.co/datasets/bbunzeck/babylm-german`

less complex sentences than the other written genres, but over half of its utterances are (in)transitive, canonical SV-sentences. This shows that even these particular sub-genres of child-directed linguistic input feature highly varied and specific constructional profiles that differ from each other.

6 Training data composition

We compose three different corpora of 5M words: (1) one corpus maximally resembling the construction composition of child-directed speech (`cds`), (2) one corpus containing a drastically higher amount of complex sentences, mirroring the distribution in the Project Gutenberg data (`pjg`), and (3) a corpus that is averaged between these two (`mix`). The relative distributions of construction types can be found in Table 2.

Construction	`cds`	`mix`	`pjg`
FRA	25%	16.5%	8%
QWH	9%	5.5%	2%
QYN	21%	12.5%	4%
COP	8%	6.5%	5%
IMP	5%	3.5%	2%
SPI	10%	9%	8%
SPT	12%	11%	10%
COM	10%	35.5%	61%

Table 2: Construction proportions of our training sets

Crucially, we sample the individual utterances for our training sets from all subcorpora in our German BabyLM dataset. By doing so, we approximate a similar (if not completely equal) mixture of sources and, therefore also a similar mixture of registers, semantic content, etc. This enables us to isolate the effect of construction distributions in our model's training data, without any interference from the possible differences between the subcorpora.

7 Model training and evaluation

We train small Llama models (Touvron et al., 2023) with `transformers` (Wolf et al., 2020). To account for the effect of subword tokenization, we compare character-level (3.7M parameters) and subword models (7.7M parameters) for the three datasets. We train all models for one epoch (loss curves and hyperparameters are in Appendix D) and share them on Hugging Face.[2] To test the effect of different random initializations and our sampling strategy, we reproduce pre-training for the `cds` models (see Appendix E).

In line with current best practices to linguistic probing, we use minimal pair datasets to evaluate our LMs' linguistic knowledge in German. The datasets always consist of a correct/grammatical and a matched incorrect/ungrammatical string. We use `minicons` (Misra, 2022) to score the sentences and evaluate 19 model checkpoints per model (10 for the first 10% of training, 9 for the remaining 90%). As an additional ablation, we also evaluate the multilingual Llama 3.2 1B[3] on all probing paradigms. Currently, no monolingual German Llama models exist. Therefore, the medium-sized 1B-parameter version of Llama 3.2, which is trained on a considerable amount of German language data, is a useful baseline for expected benchmark scores enabled through a higher model capacity and more training data.

Word-level probing Language acquisition first involves learning what words are, i.e. which (sound) sequences map to word-level items in the mental lexicon, before learning how they combine. To gauge this most basic learning step, we adapt the experimental setup from Bunzeck et al. (2025): We use `wuggy` (Keuleers and Brysbaert, 2010) to generate 1,000 nonce words (e.g. *promsen*) from existing words (e.g. *bremsen*) and then evaluate how surprised the models are by (1) the words with the context of a prepended white space (lexical decision, Le Godais et al., 2017), (2) the words in a plausible context sequence (surprisal, Hale, 2001), and (3) the words randomly inserted into implausible contexts (antisurprisal, Shafiabadi and Wisniewski, 2025). If the model is less surprised by the existing word, we count this as a correct choice in our paradigm. We calculate accuracies over the whole dataset.

Syntactic probing For syntactic probing, we use the CLAMS dataset (Mueller et al., 2020), which contains syntactic minimal pairs (grammatical/ungrammatical) for German (e.g. *Die Autoren lachen/*lacht.*). The included seven phenomena all revolve around subject-verb agreement in different contexts (across PPs, relative clauses, with coordination, etc.), resulting in different degrees of difficulty. We score the sentences for their likelihood. We calculate accuracies for correctly rated

---

[2] `https://huggingface.co/collections/bbunzeck/german-babylm-67b868e08ff8782a9814ceaf`

[3] `https://huggingface.co/meta-llama/Llama-3.2-1B`

		Character			Subword			Llama 3.2 1B
		`cds`	`mix`	`pjg`	`cds`	`mix`	`pjg`	–
Word-level	Lexical decision	97.4%	**97.6%**	97.4%	84.6%	81.9%	80.8%	69.6%
	Surprisal	99.8%	99.8%	**99.9%**	91.5%	90.3%	90.1%	98%
	AntiSurprisal	99.3%	98.9%	**99.7%**	76.5%	75.4%	75.4%	87.4%
Syntax	Simple Agreement	90%	90%	**95.7%**	80%	84.3%	92.1%	95.71%
	Across a Prepositional Phrase	61.5%	65.5%	61.8%	74.8%	73.5%	**75.5%**	83%
	Across a Subject Relative Clause	67.1%	66%	62.4%	78.4%	73.7%	**97.9%**	99.7%
	Short Verb Phrase Coordination	69.8%	68.8%	67.9%	82.6%	93.5%	**99.5%**	99.9%
	Long Verb Phrase Coordination	53.6%	60.6%	63%	60.6%	**78.8%**	78%	90.5%
	Across Object Relative Clause	58.6%	54.2%	53%	64%	66.7%	**81.6%**	86.1%
	Within Object Relative Clause	59.8%	56.4%	**72.5%**	55.8%	55.7%	49.9%	61.4%
Semantics	XCOMPS	51.5%	49.1%	49.1%	51.4%	52%	**52.3%**	58.9%

Table 3: Final evaluation results (accuracies) for all benchmarks

pairs (grammatical sentence more likely) over the whole dataset.

Semantic probing To evaluate our models' semantic knowledge, we use the XCOMPS dataset (He et al., 2025). It contains conceptual minimal pairs (e.g. *Garnele hat einen Kopf./*Ein Bikini hat einen Kopf.*)[4] that test whether LMs have acquired knowledge about conceptual properties of real-world entities. Again, we score the sentences for likelihood and calculate accuracy over the whole dataset.

8 Results

8.1 MP probing

Table 3 shows model-wise accuracies for all minimal pair sets after training for one epoch. For the word-level evaluations, accuracy scores are generally high. Across all tasks, the character models perform with almost perfect accuracy. No effect of the constructional composition of the training data is identifiable here. For the subword models, this is not true. Here, the model trained on more questions/fragments and less complex utterances (`cds`) outperforms the model that approximates written language on the construction level (`pjg`). The improvements range from 1% for anti-surprisal to 2-3% on lexical decision. Interestingly, the very large ablation model (Llama 3.2 1B) performs the worst on isolated lexical decision, but reaches high scores in the surprisal setting.

For the syntactic tests, the picture is more nuanced. Generally speaking, all our models learn to distinguish most types of grammatical and ungrammatical sentences involving agreement phenomena. The best scores are achieved on more simplistic phenomena like simple agreement or coordination with short verb phrases. Agreement phenomena that involve longer dependencies and distracting nouns, e.g. within and across relative clauses, are the hardest to learn. For the character models, the `cds` model outperforms the others on three out of seven tests, including both "across subj./obj. relative clause" conditions. For three other tests, the `pjg` model wins out, whereas the `mix` model achieves the highest scores on only one test (agreement across prepositional phrases). It should be noted, that for most phenomena, the character models do perform well above chance (by a margin of 10–20%), but still frequently make errors. The subword models show a somewhat different picture, with scores being generally higher and approximating perfect performance on 3/7 phenomena. Regarding construction distributions, the `pjg` model wins in five categories, whereas `cds` and `mix` only achieve best scores in one each. Here, the 1B-parameter Llama model outperforms our BabyLMs on 5/7 phenomena.

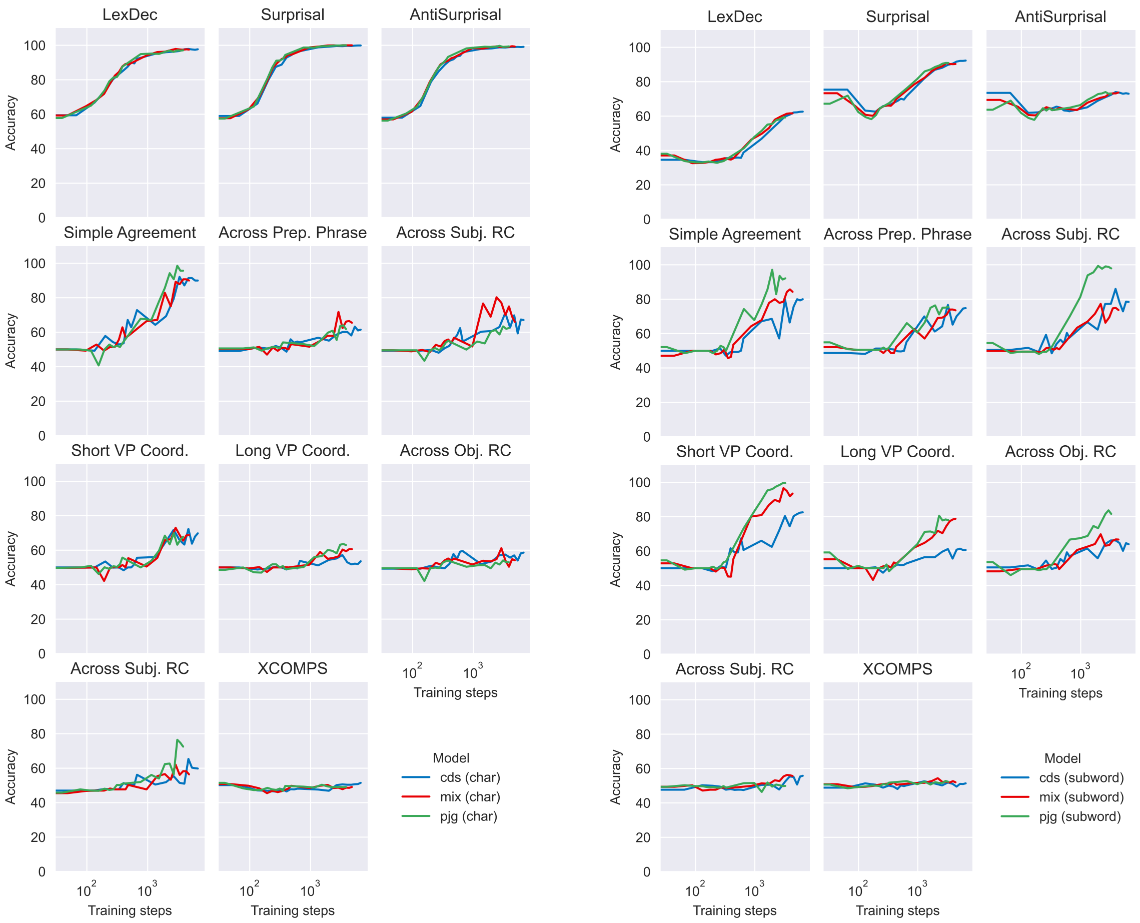
The scores on XCOMPS reveal that our small models do not reliably learn the conceptual knowledge underlying the included minimal pairs. Scores revolve around the chance baseline, with subword models performing slightly better than character models for 2/3 data mixtures. Nonetheless, these scores are also not considerably worse than the performance of our ablation model (58.9%).

8.2 Learning trajectories

Figure 3 shows the learning trajectories of our models across one training epoch. As there are no intermediate checkpoints available for the 1B-parameter ablation model, we only report trajectories for our

---

[4] We sample 1,000 MPs with randomized replacement, as the other conditions contain implausible/wrong minimal pairs. Furthermore, the quality of translation is not optimal, as exemplified by the missing determiner in front of *Garnele*.



(a) Trajectories for character models

(b) Trajectories for subword models

Figure 3: Learning trajectories for all minimal pair benchmarks

self-trained models. In line with best practices in ML (Viering and Loog, 2023), we log-scale the x-axis in our plots. This allows us to also trace early learning in more detail.

For our character models, word-level learning happens rapidly in an S-shaped curve. No differences are visible between the datasets, performance improvements align almost perfectly. For the subword models, the learning processes are not as nicely monotonically improving. Rather, the learning trajectories show a dip early in training, which then later on recovers to fairly good accuracy scores. Interestingly, despite differences in final scores, the improvements across models trained on quite different datasets still align with regard to turning and takeoff points.

This pattern is also confirmed by the learning trajectories for the syntactic phenomena. While the `pjg` models trained on more complex utterances frequently reach the highest final scores, it is remarkable to see how the improvements for all models seem to happen in parallel. The global shape of the trajectory is the same for all syntactic tests, regardless of the construction distribution. For example, the learning curve for simple agreement is steeper for the `pjg` models once learning has started, but take-off points are neatly aligned. These take-off points are pushed back by the individual paradigms' complexities — simple agreement and short VP coordination begin to improve earlier than MPs containing RCs. Finally, it is interesting to note that for the character models, word-level learning consistently stabilizes before syntactic learning, whereas both processes seem to happen concurrently in subword models (mirroring findings for English, cf. Bunzeck and Zarrieß, 2025). As our models do not learn to distinguish the semantic minimal pairs, the corresponding learning curves remain flat and performance differences are likely due to chance.

9 Discussion

This paper set out to investigate whether the constructional profile of CDS, which is shaped in a way to support the acquisition of *functional* language competence, actually influences LMs' *formal* language learning, and whether its relative lack of complex sentences and canonical SV(X) utterances makes it less useful training data, or too "impoverished" for meaningful formal learning to happen. The results of our utterance-level corpus analysis for German align with earlier findings on CDS and book language for English (Cameron-Faulkner et al., 2003; Cameron-Faulkner and Noble, 2013; Bunzeck and Diessel, 2024) and Irish (Cameron-Faulkner and Hickey, 2011), adding to the growing evidence that this linguistic distribution is fairly universal, at least in WEIRD societies (Henrich, 2024).

From a language modeling perspective, the constructional profile of training data is *not* overly important for the resulting performance on linguistic benchmarks. Rather, starting/turning points of the resulting learning trajectories are mostly determined by the respective amount of training steps. Despite models trained with more complex input resulting in slightly better performance, they do not begin to learn earlier. Global learning trajectories are extremely similar, only the local magnitude differs between different constructional setups. This provides further evidence that LMs based on the Transformer architecture (Vaswani et al., 2017) not only memorize language from their training data, but generalize to the underlying patterns. The same holds true or word-level learning processes such as lexical decision or (anti)surprisal tests, where data with more fragments and questions even seems to be rather beneficial. Furthermore, the comparison of our results to the Llama 3.2 1B model shows that rather high scores are already attainable with small models and little data (only on long VP-coordination do our models underperform).

What does this now mean for theories of language acquisition? This study was inspired by findings of construction-based corpus analyses (Cameron-Faulkner et al., 2003; Cameron-Faulkner and Hickey, 2011; Bunzeck and Diessel, 2024), which argue that the specific constructional profile of CDS is beneficial to acquisition. Of course, LMs and minimal pair evaluations do not directly correspond to the learning processes in humans and we cannot make causal claims about them. Yet, our methodology can provide evidence as to what kinds of input data is beneficial to a purely statistical learner (that does not even tap into the functional side of language, cf. Mahowald et al., 2024), an abstraction that is highly relevant to usage-based theories (Ambridge et al., 2015). On a formal level, there seem to be comparatively little disadvantages for models trained on less "complex" or somewhat impoverished data. Despite more complex data leading to slightly better benchmark scores, the learning trajectories remain largely unaffected (although somewhat erratic, cf. Bunzeck and Zarrieß, 2024). What really shapes the learning process in our LMs is the amount of input, not its formal complexity (similar to findings for children by Huttenlocher et al., 1991; Rowe, 2012). An increase in appropriate construction types for child-rearing (like questions, imperatives, or fragments) does not hinder formal learning (if only reduce its magnitude slightly). As CLAMS only focuses on subject-verb agreement in canonical SV(X)-sentences, it is rather surprising that the much higher amount of questions in the cds dataset does not negatively affect performance, although the subjects' and predicates' positions are switched in German yes/no-questions. Conversely, the cds dataset even enables word-level learning to converge to a better end state. This also aligns with a broader trend found in language acquisition studies — the complexity and quality of input can indeed predict later language skills (Noble et al., 2020; Alroqi et al., 2023), but the ground level is always extremely high already: being a competent user of the language itself. Furthermore, quality varies with many more extralinguistic factors like the number of siblings (Laing and Bergelson, 2024) or cultural factors (Bergelson et al., 2023; Bunce et al., 2024).

10 Conclusion

Our findings add to the growing body of research on BabyLMs (Warstadt et al., 2023; Hu et al., 2024). Similarly to English models, our German BabyLMs only need little data — the cds dataset contains approx. 820,000 sentences, and given the estimation by Cameron-Faulkner et al. (2003) that children hear around 7,000 utterances per day, our data approximates the number of utterances heard over only 120 days — to learn a fair amount of syntax and almost impeccable lexical knowledge, with trajectories mirroring those of English models (Bunzeck and Zarrieß, 2025). We hope that our dataset enables other scholars to carry out experiments with developmentally plausible LMs beyond the dominating English LMs, and that our data provides inspiration to those compiling BabyLM corpora for other languages.

Limitations

Our study is limited by data availability. Creating a full-fledged 100M-token BabyLM dataset with *only* child-directed speech or other explicitly child-directed materials is currently out of question, as neither CHILDES nor other sources contain even remotely enough data for languages other than English. To reach higher token counts, padding with larger data sets, e.g. more tokens from the OpenSubtitles dataset, would be necessary. Principally, synthetic corpora like the TinyStories dataset (Eldan and Li, 2023), which contains children's stories generated by GPT-3 or TinyDialogues by Feng et al. (2024) would provide an unlimited source of training data. However, our inspection of their generated dialogues yielded that they drastically underestimate the high numbers of grammatical fragments, questions and short SV(X)-utterances in real-world data. Similarly, there are little to no evaluation sets specifically aimed at German, beyond those that we included/creates ourselves, especially on the syntactic level. Only very recently, evaluation datasets like the massively multilingual MultiBLiMP have begun to fill this gap (Jumelet et al., 2025). Also, such minimal pair datasets are principally at odds with the usage-based, constructionist view on language development, because they are grounded in the Generativist notion of defining rules that can determine whether an utterance belongs to a language or not, whereas usage-based linguistics has adopted a network-based, associative model of linguistic knowledge (Diessel, 2019, 2023). As of late, these developments have begun to make their way into the broader LM evaluation landscape (Weissweiler et al., 2025), and novel evaluation methods like measuring affinities between lexical items and testing if different constructions manifest from them (Rozner et al., 2025a,b) provide promising future research avenues.

Moreover, actual developmental plausibility also hinges on the inclusion of other modalities. For audio data, there are few CHILDES subcorpora and other corpora that contain phonetic information (Lavechin et al., 2023), but larger models need to be trained on more data, e.g. audiobooks (Lavechin et al., 2025). A middle ground is training on textual phonetic transcriptions generated from raw text, e.g. for the BabyLM data (Goriely et al., 2024; Bunzeck et al., 2025; Goriely and Buttery, 2025b). More recently, also video recordings from infant-mounted cameras have been used to train on combined visual and auditory input modalities (Wang et al., 2023; Vong et al., 2024; Long et al., 2024). The inclusion of such data could help to disentangle learning processes further.

Ethical considerations

Given the nature of this work, there are no specific ethical concerns to address. However, we would like to stress that, of course, BabyLMs are not supposed to simulate real babies, but rather to instantiate abstractions, or *models* in the original scientific sense, of the distributional, frequency-driven aspects of their learning capacity. All claims regarding their implications for language development in the real world should be understood in this context, which we also attempted to explicate by distinguishing functional and formal aspects of learning.

Acknowledgments

This research has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) — CRC-1646, project number 512393437, project A02.

References

Haifa Alroqi, Ludovica Serratrice, and Thea Cameron-Faulkner. 2023. The association between screen media quantity, content, and context and language development. *Journal of Child Language*, 50(5):1155–1183.

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.

Ben Ambridge and Elena Lieven. 2011. *Child Language Acquisition: Contrasting Theoretical Approaches*. Cambridge University Press, Cambridge ; New York.

Heike Behrens. 2021. Constructivist Approaches to First Language Acquisition. *Journal of Child Language*, 48(5):959–983.

Elika Bergelson, Melanie Soderstrom, Iris-Corinna Schwarz, Caroline F. Rowland, Nairán Ramírez-Esparza, Lisa R. Hamrick, Ellen Marklund, Marina Kalashnikova, Ava Guez, Marisa Casillas, Lucia Benetti, Petra Van Alphen, and Alejandrina Cristia. 2023. Everyday language input and production in 1,001 children from six continents. *Proceedings of the National Academy of Sciences*, 120(52):e2300671120.

Ruth A Berman and Dan I. Slobin. 1994. *Different Ways of Relating Events in Narrative: A Crosslinguistic Study*. Erlbaum Associates, Hillsdale, NJ.

Robert C. Berwick, Paul Pietroski, Beracah Yankama, and Noam Chomsky. 2011. Poverty of the Stimulus Revisited. *Cognitive Science*, 35(7):1207–1242.

Silke Brandt, Holger Diessel, and Michael Tomasello. 2008. The acquisition of German relative clauses: A case study*. *Journal of Child Language*, 35(2):325–348.

John Bunce, Melanie Soderstrom, Elika Bergelson, Celia Rosemberg, Alejandra Stein, Florencia Alam, Maia Julieta Migdalek, and Marisa Casillas. 2024. A cross-linguistic examination of young children's everyday language experiences. *Journal of Child Language*, pages 1–29.

Bastian Bunzeck and Holger Diessel. 2024. The richness of the stimulus: Constructional variation and development in child-directed speech. *First Language*, 45(2):152–176.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Bastian Bunzeck and Sina Zarrieß. 2024. Fifty shapes of BLiMP: Syntactic learning curves in language models are not uniform, but sometimes unruly. In *Proceedings of the 2024 CLASP Conference on Multimodality and Interaction in Language Learning*, pages 39–55, Gothenburg, Sweden. Association for Computational Linguistics.

Bastian Bunzeck and Sina Zarrieß. 2025. Subword models struggle with word learning, but surprisal hides it. *arXiv preprint*.

Thea Cameron-Faulkner and Tina Hickey. 2011. Form and function in Irish child directed speech. *Cognitive Linguistics*, 22(3):569–594.

Thea Cameron-Faulkner, Elena Lieven, and Michael Tomasello. 2003. A construction based analysis of child directed speech. *Cognitive Science*, 27(6):843–873.

Thea Cameron-Faulkner and Claire Noble. 2013. A comparison of book text and Child Directed Speech. *First Language*, 33(3):268–279.

Franklin Chang, Gary S. Dell, and Kathryn Bock. 2006. Becoming syntactic. *Psychological Review*, 113(2):234–272.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Hu, Jaap Jumelet, Tal Linzen, Jing Liu, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Wilcox, and Adina Williams. 2025. BabyLM Turns 3: Call for papers for the 2025 BabyLM workshop. *arXiv preprint*.

Jidong Chen and Yasuhiro Shirai. 2015. The acquisition of relative clauses in spontaneous child speech in Mandarin Chinese. *Journal of Child Language*, 42(2):394–422.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. Number 11 in Massachusetts Institute of Technology. Research Laboratory of Electronics. Special Technical Report. MIT Press, Cambridge, Massachusetts.

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

Leshem Choshen, Ryan Cotterell, Michael Y. Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. [Call for Papers] The 2nd BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *Preprint*, arXiv:2404.06214.

Morten H Christiansen and Nick Chater. 1999. Toward a Connectionist Model of Recursion in Human Linguistic Performance. *Cognitive Science*, 23(2):157–205.

Alexander Clark and Shalom Lappin. 2011. *Linguistic Nativism and the Poverty of the Stimulus*, 1 edition. Wiley.

Stephen Crain and Paul Pietroski. 2001. Nature, Nurture And Universal Grammar. *Linguistics and Philosophy*, 24(2):139–186.

Alejandrina Cristia, Emmanuel Dupoux, Nan Bernstein Ratner, and Melanie Soderstrom. 2019. Segmentability Differences Between Child-Directed and Adult-Directed Speech: A Systematic Test With an Ecologically Valid Corpus. *Open Mind*, 3:13–22.

Holger Diessel. 2019. *The Grammar Network*. Cambridge University Press, Cambridge.

Holger Diessel. 2023. *The Constructicon: Taxonomies and Networks*, 1 edition. Cambridge University Press.

Holger Diessel and Michael Tomasello. 2000. The development of relative clauses in spontaneous child speech. *Cognitive Linguistics*, 11(1-2):131–151.

G. William Domhoff and Adam Schneider. 2008. Studying dream content using the archive and search engine on DreamBank.net. *Consciousness and Cognition*, 17(4):1238–1247.

Lukas Edman, Lisa Bylinina, Faeze Ghorbanpour, and Alexander Fraser. 2024. Are BabyLMs second language learners? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 166–173, Miami, FL, USA. Association for Computational Linguistics.

Ronen Eldan and Yuanzhi Li. 2023. TinyStories: How Small Can Language Models Be and Still Speak Coherent English? *Preprint*, arXiv:2305.07759.

Jeffrey L. Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Steven Y. Feng, Noah Goodman, and Michael Frank. 2024. Is child-directed speech effective training data for language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22055–22071, Miami, Florida, USA. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Giuliana Genovese, Maria Spinelli, Leonor J. Romero Lauro, Tiziana Aureli, Giulia Castelletti, and Mirco Fasolo. 2020. Infant-directed speech as a simplified but not simple register: A longitudinal study of lexical and syntactic features. *Journal of Child Language*, 47(1):22–44.

Martin Gerlach and Francesc Font-Clos. 2020. A Standardized Project Gutenberg Corpus for Statistical Analysis of Natural Language and Quantitative Linguistics. *Entropy*, 22(1):126.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Zébulon Goriely and Paula Buttery. 2025a. BabyLM's First Words: Word Segmentation as a Phonological Probing Task. *arXiv preprint*.

Zébulon Goriely and Paula Buttery. 2025b. IPA-CHILDES &amp; G2P+: Feature-Rich Resources for Cross-Lingual Phonology and Phonemic Language Modeling. *arXiv preprint*.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Anna Elizabeth Gowenlock, Courtenay Norbury, and Jennifer M. Rodd. 2024. Exposure to Language in Video and its Impact on Linguistic Development in Children Aged 3–11: A Scoping Review. *Journal of Cognition*, 7(1):57.

Maria Teresa Guasti. 2002. *Language Acquisition: The Growth of Grammar*. MIT Press, Cambridge, Mass.

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*.

Linyang He, Ercong Nie, Sukru Samet Dindar, Arsalan Firoozi, Adrian Florea, Van Nguyen, Corentin Puffay, Riki Shimizu, Haotian Ye, Jonathan Brennan, Helmut Schmid, Hinrich Schütze, and Nima Mesgarani. 2025. XCOMPS: A Multilingual Benchmark of Conceptual Minimal Pairs. *arXiv preprint*.

Joseph Henrich. 2024. WEIRD. In *Open Encyclopedia of Cognitive Science*, 1 edition. MIT Press.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *Preprint*, arXiv:1511.02301.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning More Grammar With Small-Scale Child-Directed Language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Janellen Huttenlocher, Wendy Haight, Anthony Bryk, Michael Seltzer, and Thomas Lyons. 1991. Early vocabulary growth: Relation to language input and gender. *Developmental Psychology*, 27(2):236–248.

Janellen Huttenlocher, Marina Vasilyeva, Elina Cymerman, and Susan Levine. 2002. Language input and child syntax. *Cognitive Psychology*, 45(3):337–374.

Daniel Jach and Gunther Dietz. 2024. KORPUS EINFACHES DEUTSCH (KED). *Korpora Deutsch als Fremdsprache*, 4.

Laura A. Janda, editor. 2013. *Cognitive Linguistics: The Quantitative Turn: The Essential Reader*. Mouton Reader. De Gruyter Mouton, Berlin.

Jaap Jumelet, Leonie Weissweiler, and Arianna Bisazza. 2025. MultiBLiMP 1.0: A Massively Multilingual Benchmark of Linguistic Minimal Pairs. *arXiv preprint*.

Krisjanis Karins, Robert MacIntyre, Monika Brandmair, Susanne Lauscher, and Cynthia McLemore. 1997. CALLHOME German Transcripts.

Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior Research Methods*, 42(3):627–633.

Marc Kupietz, Cyril Belica, Holger Keibel, and Andreas Witt. 2010. The German reference corpus DeReKo: A primordial sample for linguistic research. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Catherine Laing and Elika Bergelson. 2024. Analyzing the effect of sibling number on input and output in the first 18 months. *Infancy*, 29(2):175–195.

Marvin Lavechin, Maureen De Seyssel, Hadrien Titeux, Guillaume Wisniewski, Hervé Bredin, Alejandrina Cristia, and Emmanuel Dupoux. 2025. Simulating Early Phonetic and Word Learning Without Linguistic Categories. *Developmental Science*, 28(2):e13606.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. BabySLM: Language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*, pages 4588–4592. ISCA.

Gaël Le Godais, Tal Linzen, and Emmanuel Dupoux. 2017. Comparing character-level neural language models using a lexical decision task. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 125–130, Valencia, Spain. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Bria Long, Violet Xiang, Stefan Stojanov, Robert Z. Sparks, Zi Yin, Grace E. Keene, Alvin W. M. Tan, Steven Y. Feng, Chengxu Zhuang, Virginia A. Marchman, Daniel L. K. Yamins, and Michael C. Frank. 2024. The BabyView dataset: High-resolution egocentric videos of infants' and young children's everyday experiences. *arXiv preprint*.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*, 3 edition. Lawrence Erlbaum Associates, Mahwah, NJ.

Brian MacWhinney. 2004. A multiple process solution to the logical problem of language acquisition. *Journal of Child Language*, 31(4):883–914.

Kyle Mahowald, Anna A. Ivanova, Idan A. Blank, Nancy Kanwisher, Joshua B. Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in Cognitive Sciences*, pages 517–540.

Raphaël Millière. 2024. Language Models as Models of Language. *arXiv preprint*.

Kanishka Misra. 2022. Minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models. *arXiv preprint*.

Kanishka Misra and Kyle Mahowald. 2024. Language models learn rare phenomena from less rare phenomena: The case of the missing AANNs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 913–929, Miami, Florida, USA. Association for Computational Linguistics.

Jessica L. Montag. 2019. Differences in sentence complexity in the text of children's picture books and child-directed speech. *First Language*, 39(5):527–546.

Sherin Muckatira, Vijeta Deshpande, Vladislav Lialin, and Anna Rumshisky. 2024. Emergent Abilities in Reduced-Scale Generative Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1242–1257, Mexico City, Mexico. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-Linguistic Syntactic Evaluation of Word Prediction Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Claire Noble, Thea Cameron-Faulkner, Andrew Jessop, Anna Coates, Hannah Sawyer, Rachel Taylor-Ims, and Caroline F. Rowland. 2020. The Impact of Interactive Shared Book Reading on Children's Language Skills: A Randomized Controlled Trial. *Journal of Speech, Language, and Hearing Research*, 63(6):1878–1897.

Claire H. Noble, Thea Cameron-Faulkner, and Elena Lieven. 2018. Keeping it simple: The grammatical properties of shared book reading. *Journal of Child Language*, 45(3):753–766.

Francesca Padovani, Jaap Jumelet, Yevgen Matusevych, and Arianna Bisazza. 2025. Child-Directed Language Does Not Consistently Boost Syntax Learning in Language Models. *arXiv preprint*.

Ludovica Pannitto and Aurélie Herbelot. 2020. Recurrent babbling: Evaluating the acquisition of grammar from limited input data. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 165–176, Online. Association for Computational Linguistics.

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.

Steven T. Piantadosi. 2024. Modern language models refute Chomsky's approach to language. In Edward Gibson and Moshe Poliak, editors, *From Fieldwork to Linguistic Theory: A Tribute to Dan Everett*, Empirically Oriented Theoretical Morphology and Syntax. Language Science Press, Berlin.

Julian M. Pine. 1994. The language of primary caregivers. In Clare Gallaway and Brian J. Richards, editors, *Input and Interaction in Language Acquisition*, 1 edition, pages 15–37. Cambridge University Press.

Laurent Prévot, Sheng-Fu Wang, Jou-An Chi, and Shu-Kai Hsieh. 2024. Extending the BabyLM initiative : Promoting diversity in datasets and metrics through high-quality linguistic corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 147–158, Miami, FL, USA. Association for Computational Linguistics.

Geoffrey K Pullum and Barbara C Scholz. 2002. Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, 18(1-2).

Silke Reineke, Arnulf Deppermann, and Thomas Schmidt. 2023. Das Forschungs- und Lehrkorpus für Gesprochenes Deutsch (FOLK). In Arnulf Deppermann, Christian Fandrych, Marc Kupietz, and Thomas Schmidt, editors, *Korpora in Der Germanistischen Sprachwissenschaft*, pages 71–102. De Gruyter.

Meredith L. Rowe. 2012. A Longitudinal Investigation of the Role of Quantity and Quality of Child-Directed Speech in Vocabulary Development. *Child Development*, 83(5):1762–1774.

Joshua Rozner, Leonie Weissweiler, Kyle Mahowald, and Cory Shain. 2025a. Constructions are Revealed in Word Distributions. *arXiv preprint*.

Joshua Rozner, Leonie Weissweiler, and Cory Shain. 2025b. BabyLM's First Constructions: Causal interventions provide a signal of learning. *arXiv preprint*.

David E. Rumelhart and James L. McClelland. 1986. On Learning the Past Tenses of English Verbs. In *Parallel Distributed Processing*, volume 2, pages 535–551. MIT Press, Cambridge, MA.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is More: Pre-Training Cross-Lingual Small-Scale Language Models with Cognitively-Plausible Curriculum Learning Strategies. *arXiv preprint*.

Nazanin Shafiabadi and Guillaume Wisniewski. 2025. Beyond surprisal: A dual metric framework for lexical skill acquisition in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6636–6641, Abu Dhabi, UAE. Association for Computational Linguistics.

Catherine E. Snow and Charles A. Ferguson, editors. 1977. *Talking to Children: Language Input and Acquisition*. Cambridge University Press, Cambridge, MA.

Melanie Soderstrom. 2007. Beyond babytalk: Re-evaluating the nature and content of speech input to preverbal infants. *Developmental Review*, 27(4):501–532.

James W. Stigler, Ronald Gallimore, and James Hiebert. 2000. Using Video Surveys to Compare Classrooms and Teaching Across Cultures: Examples and Lessons From the TIMSS Video Studies. *Educational Psychologist*, 35(2):87–100.

Alice Suozzi, Luca Capone, Gianluca E. Lebani, and Alessandro Lenci. 2025. BAMBI: Developing Baby Language Models for Italian. *arXiv preprint*.

Shira Tal, Eitan Grossman, and Inbal Arnon. 2024. Infant-directed speech becomes less redundant as infants grow: Implications for language learning. *Cognition*, 249:105817.

Margaret Thomas. 2002. Development of the concept of "the poverty of the stimulus". *The Linguistic Review*, 18(1-2).

Michael Tomasello. 2005. Beyond formalities: The case of language acquisition. *The Linguistic Review*, 22(2-4).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *Preprint*, arXiv:2302.13971.

Mark VanDam, Anne Warlaumont, Elika Bergelson, Alejandrina Cristia, Melanie Soderstrom, Paul De Palma, and Brian MacWhinney. 2016. HomeBank: An Online Repository of Daylong Child-Centered Audio Recordings. *Seminars in Speech and Language*, 37(02):128–142.

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

Tom Viering and Marco Loog. 2023. The Shape of Learning Curves: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7799–7819.

Wai Keen Vong, Wentao Wang, A. Emin Orhan, and Brenden M. Lake. 2024. Grounded language acquisition through the eyes and ears of a single child. *Science*, 383(6682):504–511.

Wentao Wang, Wai Keen Vong, Najoung Kim, and Brenden M. Lake. 2023. Finding Structure in One Child's Linguistic Experience. *Cognitive Science*, 47(6):e13305.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Gotlieb Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–6, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Taiqi He, Naoki Otani, David R. Mortensen, Lori Levin, and Hinrich Schütze. 2023. Construction Grammar Provides Unique Insight into Neural Language Models. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, pages 85–95.

Leonie Weissweiler, Kyle Mahowald, and Adele Goldberg. 2025. Linguistic Generalizations are not Rules: Impacts on Evaluation of LMs. *arXiv preprint*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick Von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hong Meng Yam and Nathan Paek. 2024. What should baby models read? Exploring sample-efficient data composition on model performance. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 284–291, Miami, FL, USA. Association for Computational Linguistics.

Charles D. Yang. 2004. Universal Grammar, statistics or both? *Trends in Cognitive Sciences*, 8(10):451–456.

Daniel Yurovsky, Chen Yu, and Linda B. Smith. 2012. Statistical Speech Segmentation and Word Learning in Parallel: Scaffolding from Child-Directed Speech. *Frontiers in Psychology*, 3.

Renate Zangl and Debra L. Mills. 2007. Increased Brain Activity to Infant-Directed Speech in 6- and 13-Month-Old Infants. *Infancy*, 11(1):31–62.

A Excluded corpora

Several corpora that are — in principal — available for German were excluded from our analysis. The Folk corpus (Reineke et al., 2023) and the Simple German corpus (Jach and Dietz, 2024) are not available under any open licenses, while the data in other German reference corpora (Kupietz et al., 2010) are not available in their entirety but can only be queried through web interfaces. Finally, Homebank features day-long audio recordings of children and their surroundings/inputs (VanDam et al., 2016), but without any written transcriptions.

B Data cleaning

In line with best practices in language modeling, we extensively clean and normalize our data.

All subcorpora We replaced all local variants of single/double quotation marks with either `' '` or `" "`. We further reduced multiple superfluous whitespace and newlines to singular whitespaces.

Talkbank data For the data sourced from talkbank (i.e. the CHILDES corpora and CallHome), we remove all mark-up and additional info on false starts, hesitations, implicit completions or other explanations. Furthermore, we also remove all empty utterances and those containing `xxx` or `yyy`, placeholder symbols for personally identifiable information.

Project Gutenberg For the Project Gutenberg data, we excluded all lines with more than 6 consecutive whitespaces, as these always turned out to be title pages, index pages, etc., which contain no useful language data. Additionally, we removed all textual data in square brackets, which almost always corresponded to pointers to pictures which are not found in text-only version, or additional explanations by the volunteers who digitized the respective books.

OpenSubtitles For the OpenSubtitles data, we removed all text in parentheses, which corresponds to speaker information. Also, we removed sentence-initial dashes (`-`) which were sometimes added. We also amended OCR errors (like mangled uppercase I and lowercase l) as far as possible.

Fluter For the data sourced from the Fluter magazine, we removed all lines containing additional metatextual data, like author info and image credits, before pre-training.

C Exact construction proportions

Table 4 shows the exact construction proportions for all of our subcorpora. This data underlies the visualization in Figure 2.

Construction	Proj. Gut.	Dreamb.	Fluter	News	Wikib.	Klex.	Mini-Klex.	OpenSub.	CallHome	Child speech	CDS
FRA	7.8%	6.3%	6.2%	4.0%	11.6%	6.3%	2.5%	24.1%	37.0%	55.1%	24.5%
QWH	1.9%	0.3%	2.6%	1.4%	0.5%	2.9%	<0.1%	7.3%	2.1%	3.5%	8.8%
QYN	3.7%	0.7%	2.8%	1.6%	0.5%	0.4%	<0.1%	10.9%	6.9%	4.7%	20.7%
COP	4.6%	7.1%	7.7%	7.4%	10.9%	13.2%	21.4%	9.7%	10.7%	5.7%	8.1%
IMP	1.5%	0.1%	0.2%	0.1%	0.3%	<0.1%	<0.1%	4.6%	0.4%	2.0%	4.5%
SPI	7.5%	9.2%	9.7%	13.7%	9.5%	13.9%	19.9%	9.9%	8.8%	11.5%	10.1%
SPT	10.5%	14.5%	18.7%	25.7%	24.1%	28.1%	37.2%	18.0%	14.1%	11.9%	12.3%
COM	62.5%	61.8%	52.2%	46.1%	42.7%	35.2%	18.9%	15.4%	20.0%	5.7%	11.0%

Table 4: Exact proportions of constructions for all subcorpora

D Model hyperparameters and training details

Our models share a hidden/intermediate/embedding size of 256, 8 hidden layers and attentions heads, and a context length of 128. For the character models, the vocabulary consists of all printable ASCII characters and characters used in written German (üäöß and their uppercase variants), amounting to a vocab. size of 110 and 3,730,688 parameters. For the subword models, we train a BPE tokenizer (Gage, 1994) with a vocab. size of 8,000 and add two special tokens (BOS, EOS/PAD), resulting in 8,002 vocab. tokens and 7,771,392 parameters. Model training takes approx. 2h on a MacBook Pro with an Apple M2 Pro CPU/GPU.

We reproduce the training and test loss curves for our models in Figure 4. For the test loss, we evaluated perplexity on a held-out, randomly sampled portion of each individual training corpus. We find no principal differences in loss development, although the character models and models trained on the cds data seem to converge the fastest. As the similar curves for train and test loss indicate, all models succeed in optimizing for their next-token prediction goal. It should be noted that due to longer/shorter sequences in the different data mixtures and our choice of padding to the maximum sequence length, some models are trained for more *steps*, although the number of *lexical tokens* remains the same.

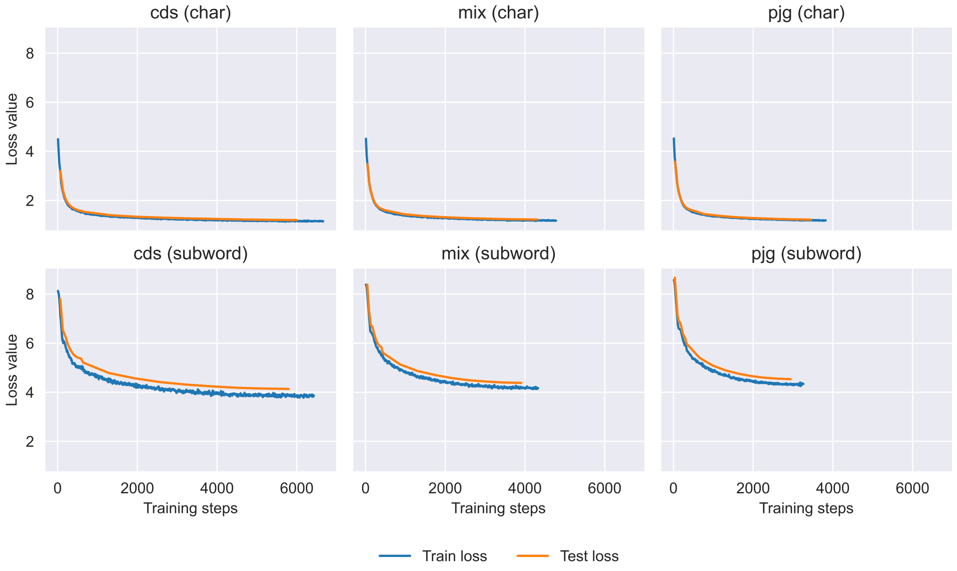
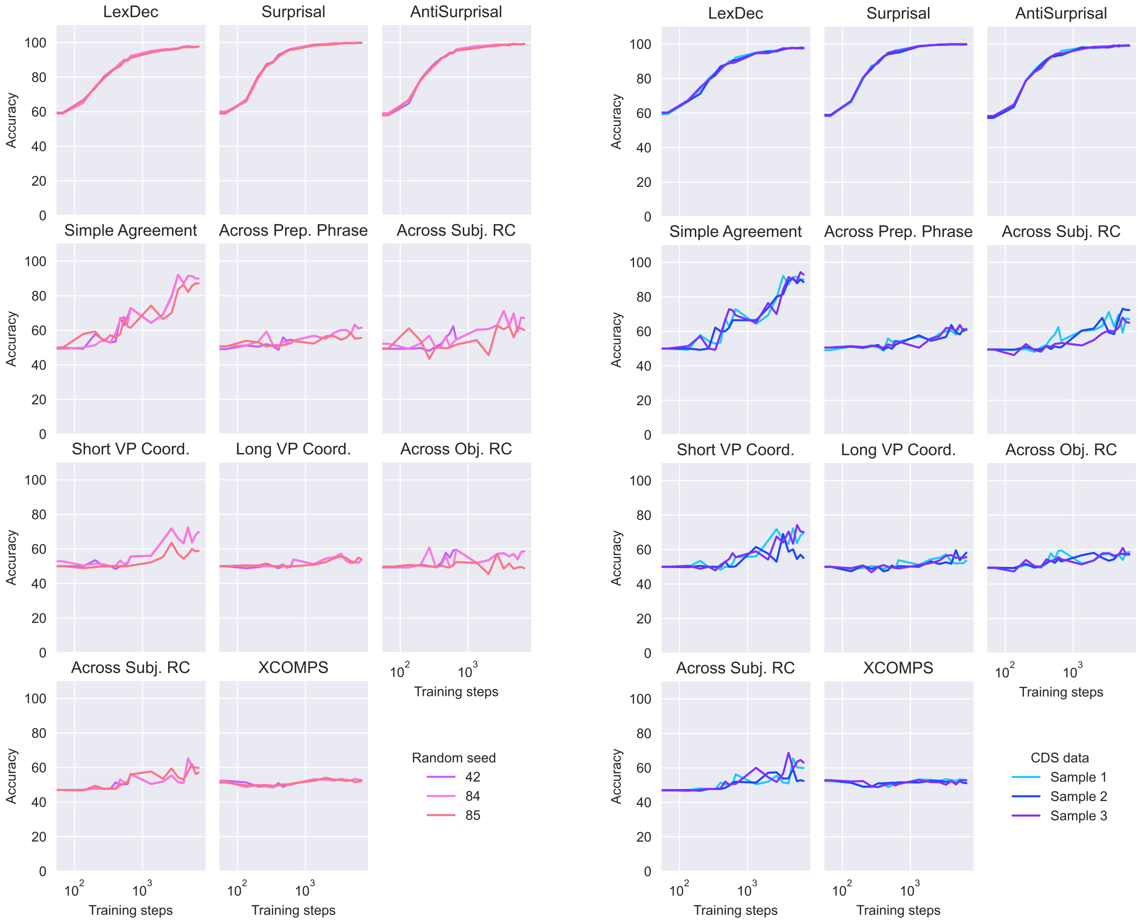


Figure 4: Loss curves for our self-trained character and subword models

E Repeated training runs

A common criticism towards the BabyLM paradigm is the purported effect of training noise on model performance, which is hard to disentangle from real training data effects. While training and evaluating multiple random seeds for all our models would be too costly, we repeated two additional training runs for the character-level cds model with different random initializations (learning trajectories in Figure 5a) and two additional training runs where we re-sampled the cds dataset from our whole corpus with the exact same construction composition, but different content (learning curves in Figure 5b). In both cases, the learning trajectories do not differ tremendously. For the word-level phenomena (LexDec, Surprisal, AntiSurprisal), the curves overlap almost perfectly. For the syntax phenomena, we can see some variation and oscillation in the curves, but the trajectories still remain extremely similar (and do not differ in their steepness, the main effect that we see in Figure 3 between the datasets with different construction compositions).



(a) Trajectories for different random initializations

(b) Trajectories for different samples of cds data

Figure 5: Learning trajectories for our comparison models

Adapting Large Language Models for Movie Domain with Narrative Understanding Tasks

Siqi Shen
University of Michigan
shensq@umich.edu

Amanmeet Garg
Amazon Inc.
amanmega@amazon.com

Abstract

Large language models (LLMs) have been deployed in a wide spectrum of domains and applications due to their strong language understanding capabilities obtained through pretraining. However, their performance on specific domain is usually suboptimal due to limited exposure to domain-specific tasks. Adapting LLMs to the movie domain poses unique challenges due to complex narratives that cannot be fully captured through subtitles or scripts alone. In this paper, we decompose movie understanding capability into a suite of narrative understanding tasks based on narrative theory. We construct a dataset for these tasks based on resources in the movie domain, and use it to examine the effect of different domain adaptation strategies. Our experiment results show the effectiveness of our approach in improving the narrative understanding of LLMs and highlight the trade-offs between domain-specific and general instruction capabilities.

1 Introduction

Large language models have revolutionized natural language processing with their ability to understand and generate text across diverse domains (Radford and Narasimhan, 2018). However, these models often struggle with specialized tasks in domains that are underrepresented in their training data, such as cinematic content. While domain adaptation has shown success in fields such as medicine (Yang et al., 2023), finance(Wu et al., 2023), and law(Cui et al., 2024), adapting LLMs to understand movie narratives remains an underexplored challenge despite cinema's widespread cultural impact.

Adapting an LLM to the movie domain, on the other hand, remains underexplored despite movies and TV shows having such a large audience. An adapted LLM could potentially enable and benefit many movie-related tasks, be it a better summarization of movie content or more accurate content moderation.

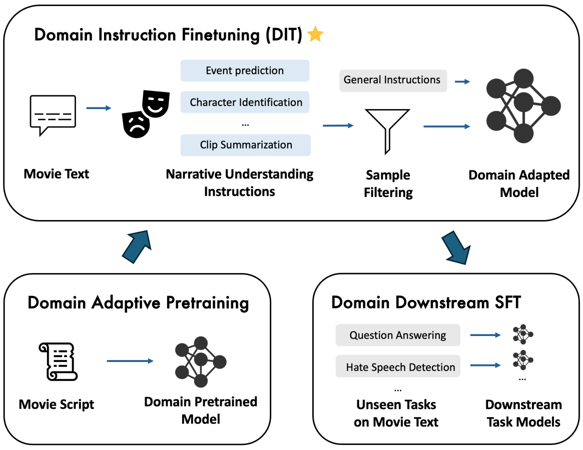


Figure 1: Different stages of adapting a generic LLM to the movie domain.

Understanding text in the movie domain, however, pose its unique set of challenges. Movie subtitles lack visual context and contain fragmented dialogues with interruptions, incomplete sentences, and repetitions. In addition, understanding a line sometimes requires knowledge of the story setups or the background of the characters. Existing work on domain adaptation of general-use LLMs generally falls into the following three categories. The first uses a domain-specific corpus $\mathcal{C}$ directly for pretraining without specifying domain tasks (Wu et al., 2023, 2024) referred to as Domain Adaptative Pretraining (DPT), which is only effective when substantial domain knowledge is unseen during the model's pretraining stage. The second category focuses on a single domain task t, for example, question answering, while generalization to unseen tasks $\mathcal{U}$ is not a main consideration(Wu et al., 2024; Singhal et al., 2025). The last category compiles a set of domain-related tasks $\mathcal{T}$ and uses all of them during training, aiming to broaden the coverage of domain use cases (Cui et al., 2024; Liu et al., 2023).

A narrative consists of two key components: the story elements (characters, events, and settings)

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 187–200
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

and how these elements are presented (discourse). In movies, for example, a story includes both what happens (a detective solving a crime) and how it is told (revealing clues gradually to build suspense). This framework, established by Chatman (Chatman, 1980), provides a systematic way to analyze movie narratives by breaking them down into these fundamental components. Understanding these elements is crucial for teaching LLMs to comprehend movie content effectively.

We build the instruction dataset with movie text that focuses on narrative understanding capabilities. Our narrative understanding tasks $\mathcal{T}$ include predicting the main action or place of a movie clip, inferring the characters along with their interactions and relations, summarizing the subtitle, and segmenting the story according to plots. We control the quality of the dataset by sourcing from various datasets and databases in the domain, and curate instruction samples with both a suite of designed rules and LLM-based judges.

Using this dataset, we conduct domain adaptation with instruction finetuning on both narrative understanding and general instructions. We examine the effect of the training schema as well as data mixture, and illustrate that LLM can be effectively adapted to the movie domain with a trade-off between general instruction following. We also showcase the interplay of domain instruction finetuning with pretraining on movie text and in-domain downstream tasks.

Overall, our work offers insight on how to adapt an LLM to the movie domain and a better understanding of the effect of the adaptation procedure, with the following main contributions: (1) We propose a suite of tasks for adapting LLM to the movie domain based on narrative theory. (2) We collected an instruction dataset consisting of narrative understanding tasks with various quality control measures. (3) We conduct systematic experiments that analyze the effect of adapting LLMs on narrative understanding tasks, showing that adaptation also helps in the movie domain.

2 Related works

Instruction finetuning Instruction fine-tuning enables large language models (LLMs) to follow user instructions across various tasks by exposing them to diverse task instructions. Early works on instruction datasets, such as T0-SF (Sanh et al., 2021) and NaturalInstructions (Wang et al., 2022b) were mostly compiled by humans. It shows the potential of instruction finetuning in zero-shot and few-shot scenarios, particularly in generalization across unseen tasks (Wei et al., 2021; Chung et al., 2024). Methods like Self-Instruct (Wang et al., 2022a) that automatically generate new tasks have become popular alternatives to human annotation. It increases the diversity of tasks based on seed tasks in a bootstrapping way that drives the success of many open-source models (Taori et al., 2023; Xu et al., 2023; Peng et al., 2023).

Domain Adaptation Deploying an off-the-shelf LLM on domain tasks usually leads to suboptimal performances, thus there are attempts on adapting them to various domains. BloombergGPT (Wu et al., 2023) mixes the proprietorial finance text corpus with the general text corpus and is trained with the regular LLM pre-training task. Finetuning on domain-specific tasks is a more common and efficient solution for most application scenarios. FinGPT (Yang et al., 2023) adapts to the finance domain with a new task to predict the change in stock price, and sees improvement in in-domain tasks such as portfolio management. ChatLaw (Cui et al., 2024) construct a legal dataset comprising 10 major categories including case classification, statute prediction, and test its performance with Unified Qualification Exam for Legal Professionals. ChipNeMo (Liu et al., 2023) adopts LLM for chip design by training it in chip design documents and code. These existing works suggest that training general LLM on a selection set of domain-specific tasks can increase the model's general capability tasks in that domain.

3 Narrative Understandings in Movie Domain

3.1 Definition of Narrative

Narratives can be defined as stories in which a series of events or experiences unfold over time. Novel, fable, opera, and film are all good examples of narratives in different formats. Narratives are built upon different constituents, such as agent, scene, with the events organized in a particular order, and that structure is called the narrative structure. Despite the fact that narratology scholars have no consensus on what a narrative structure is, they offer different ways to comprehensively understand a narrative. In our work, we adopt the version of the narrative structures of Seymour Chatman (Chatman, 1980) as shown in Figure 5, with *Story* and

Task	Input	Output
subtitle_action	Look, see that? It's on. It worked. What do you mean it worked? Everything's off. Shutdown tripped the circuit breakers. Turn them back on, reboot a few systems	smoke, watch screen, press button
subtitle_place	The defendant, please.	court (inside)
synopsis_event	"Ron tries to sell medicine to the gay men from the support group he attended before. Nobody buys, and Ron leaves."	selling
synopsis_place	*same as above*	meeting room
subtitle_character	PersonA: Here is a support group that meets daily in Draddy Auditorium. I suggest you try it out and maybe go talk about your feelings, your concerns. PersonB: I'm dying. You telling me to go get a hug from a bunch of fag*ots?	PersonA: Eve; PersonB: Ron Woodroof
subtitle_interaction	*same as above*	suggests, rebukes
subtitle_relation	How you doing, Miss Clark? All right, Henry. Thank you. How you doing? If God is willing, Miss Clark. That's good.	acquaintance
short_subtitle_synopsis	*subtitle of a clip*	*synopsis of the clip*
long_subtitle_synopsis	*subtitle of a story part*	*synopsis of the story part*
synopsis_turning_point	(37) Later, Zira gets close to Taylor's cage and he grabs her note book and pencil.(38) Julius, (Buck Kartalian), the gorilla guard, enters the cage to beat up Taylor and retrieves the stolen items...	(41)

Table 1: Examples of narrative understanding tasks

Discourse as its two main constituents.

Story The *Story* of a narrative is the content occurring in the narrative, including events and existents. *Events* is also traditionally referred as plot, capturing key information about what is going on in a movie clip or a paragraph of text. That includes actions originating from characters such as "fell on to the ground" or things happening with other objects such as "It is raining here." *Existents*, on the other hand, instantiate events with concrete characters and settings. For example, the character who fell can be "Indiana Jones" and the setting can be "in a rainforest", and that immediately brings up the image of muddy ground and the thrill of treasure hunting to our mind.

Discourse Segments of content scattered around on their own do not give us anything interesting or thought-provoking, as they need to be delivered with some arrangements to make sense. That is done by the *Discourse* of a narrative, which is about the way to express the content. The discourse is in charge of both temporal and spatial arrangements. The temporal arrangement is more straightforward, for example, how does that events chronologically revolve and if there is non-linear storytelling etc. The spatial arrangement is more about the focus of spatial attention, maneuvering what the audience sees through the camera eye, and figuratively in verbal narratives.

3.2 Movie Narrative Understanding Tasks

Movie as a form of narrative consists of information in different modalities, including text, audio, and visual information. However, a lot of information can be inferred from the text alone. Taking a simple one-word line "Order!" One can guess that the setting of the story is probably in court or parliament. Also, if a host is announcing "Shari and Prakash are so happy that so many of you are here today joining in holy matrimony," then it is likely that Shari and Prakash are new spouses. It is expected that LLMs' capabilities on different narrative elements can help them adapt to the movie domain with better language understanding. We restrain our scope to the text modality, nonetheless, our general framework can be extended to a multimodal setting with Vision-Language models and visual narrative understanding tasks.

To better facilitate LLMs' narrative understanding on the movie domain, we propose a comprehensive array of tasks shown in Table 1, which cover each key element of a narrative at both the story level or discourse level.

For the *Story* element, we include several prediction tasks on the events including actions and happenings, as well as the existents of characters and settings. The tasks of event and place prediction from the synopsis take the summarized text description of the movie clip and answer the question based on that. It improves the model on extracting information about essential narrative components from the summarization. These prediction tasks are also conducted with subtitles as the input. Since subtitle is not as concise and well-formatted as the synopsis, this set of tasks is considered more difficult and often involves making inferences between the conversation lines. As character is another centerpiece making the content, we include a character disambiguation task that predicts the corresponding speaker given the subtitle and the story background. Story background is needed in this case to match the speaker to names, as there is usually no direct mention of it unless some speaker calls others' names. The character interaction prediction and character relation prediction go a step further and require the model to understand the dynamics between multiple characters.

At the *Discourse* level, we include a turning point prediction task as well as two summarization tasks. The turning points are crucial narrative moments that segment a movie into thematic story parts (Papalampidi et al., 2019). The story part is a larger unit than the scene, where a movie usually consists of several parts for setting up, complications of the plot, etc., and predicting the turning points requires an overall understanding of long and complex narratives. Summarization based on subtitles of a movie clip or a whole story part is also introduced, which requires correctly capturing the plot progression.

4 Movie Domain Adaptation

We discuss the sample collection procedures for each narrative understanding task in § 4.1 and the quality control measures in § 4.2. We discuss the method to perform domain adaptation in § 4.3.

4.1 Data Collection

We construct our samples around the subtitle and synopsis since they are the most available textual sources for movies. Among all our proposed narrative understanding tasks, most of them do not have a straightforward way to get labels directly from the subtitle only. Therefore, we look at existing human-annotated datasets on the movie domain for our need, as they may offer better fidelity than relying on synthesized data alone.

More specifically, we collect the place and action tags for movie scenes from MovieNet (Huang et al., 2020) for action prediction and place prediction based on subtitles. Each movie scene is further divided into movie shots in MovieNet, we aggregate the subtitles of movie shots that belong to the same scene, and match the place or action tag as its label. We collect the same information from (Vicol et al., 2018) with the difference that the event and place now match the synopsis of the clip instead of the subtitle, which is a short descriptive sentence in natural language. For tasks centered on characters, we collect characters occurring within a movie clip from MovieGraphs, and obtain their relations and interactions as well. We also keep a record of the corresponding subtitles and timestamps for all samples. The turning point prediction task is based on the segmentation of the story parts in MovieNet by tracking the sentence index where the synopsis turns into a new story part. The clip summarization and story summarization tasks are constructed by matching the subtitle with its synopsis at a clip and story part level correspondingly.

Instruct prompt construction We manually construct a prompt template for each narrative understanding task and use the collected labels to instantiate training samples. We specify the requirement for each individual task in the user prompt and use a format aligned with the Alpaca (Taori et al., 2023). The specific prompts that we used can be found in Table 8.

4.2 Sample Refinement

The multimodal nature of the MovieNet and MovieGraphs datasets presents challenges for text-only analysis. For instance, when aggregating subtitles from consecutive shots, redundant dialogues may appear, particularly in scenes with minimal conversation. For example, a subtitle can sometimes occur in multiple consecutive movie shots in MovieNet, which can introduce repeated utterances when aggregated into a movie clip. That usually happens when the main characters are not having a lot of conversation and the subtitle lingers for more than intended.

There are also many noises in the annotated samples. For example, a lot of subtitles contain very few verbal exchanges in MovieGraphs, since the annotation is at the clip level, which is usually less than one minute in length. Also, the subtitles are split by timestamps with no information of speakers available, which makes it more difficult to make sense of whether it is the same speaker talking.

Besides, a common issue with most tasks is that textual data alone does not contain sufficient information to infer the desired answer, especially for clips with multiple labels. For example, inferring multiple locations like *[desert, doorway, living room, yard (outside)]* for a single scene would be very hard, and it is a problem rooted in scene parsing of the datasets we use. It is the same case for character labels, there are labels such as *bridesmaid #4*, which sometimes just appear in the clip without saying anything. All these issues require more careful preprocessing and sample selection as follows to curate a useful dataset.

Rule-based filtering To avoid samples with too much ambiguity, we enforce some restrictions on the labels. For character-related tasks, we keep samples with exactly two named characters and remove samples with more than one interaction and relationship. The order of the character is tracked as there are relationships such as "parent", which are not mutual. We also keep only samples with one place and event label, as that corresponds to correctly segmented scenes. For actions, we keep the actions that are salient in the clip with a duration of more than 4 seconds.

Introducing script information Since subtitles are noisy and lack information of the speakers at an utterance level, we decide to use the dialogue in the script to replace the subtitle. Using a clip from the movie Indiana Jones and the Last Crusade as an example (Table 10), it is apparent that the script provides richer information, including the description of each scene and the speaker information for all the dialogues. We use all scripts on The Internet Movie Script Database (IMSDb)[1] , and use the TMDB API [2] to get the meta-information for each title. The labels from MovieNet and MovieGraphs are based on the IMDB ID, so we get the mapping from IMDB to TMDB ID, and then map the samples to the corresponding script files.

Note that subtitle does not strictly follow the dialogue provided in the script, and is more like an improvisation from the actors. Therefore, finding the corresponding segment of script for a subtitle itself is a long-text retrieval task. We use fuzzy string match for each utterance in the subtitle to an utterance in the script based on editing distance. An utterance in the script may be split into multiple utterances in the subtitle, so we use a partial ratio match, which matches the shorter utterance with substrings of the same length in the script utterance. We add dummy tokens to the script utterance to make sure it is the longer one, and cut off with a match score of 90. We then collect the matched utterance from the script to replace the original subtitles for our tasks. Our method expects high precision with lower recall compared with the embedding-based retrieval method for ignoring sentences with the same semantics, which meets our requirement for data filtering.

Modified character prediction task It usually does not make much sense to predict a named character from the subtitle alone unless there is a name called out in the clip, while providing a synopsis makes it a named entity recognition task without making an inference. Therefore, we modify the character prediction task and let the model match the character to the speakers. This task requires an understanding of the synopsis and dialogue, and can have samples constructed without any human labeling.

4.2.1 LLM-based sample selection

We take the common assumption that verifying an answer is easier than generating one, and we do another round of filtering using LLM for all tasks other than summarization tasks and turning point prediction. We provide the LLM with data samples and ask whether the expected output can be inferred from the subtitle as well as the synopsis for some tasks. We also ask the model to give an explanation for the decision. The prompts that we use for sample filtering can be found in Table 7. We only keep samples that are deemed to be inferable from the input and use them in both training and testing. We also provide the explanation at the training time so that the model generates the label corresponding to the task and gives an explanation.

4.3 Domain Adaptation with Instruction Finetuning

We adapt LLM to the movie domain following the instruction finetuning paradigm (Shi et al., 2024; Zhang et al., 2023a). Instruction finetuning is an approach that finetune pretrained LLMs on a va-

---

[1] https://imsdb.com/

[2] https://www.themoviedb.org/

riety of prompts and tasks in the form of natural language, such that the model learns the desired capability while being able to follow different instructions for practical use. To maintain models' instruction-following capability, we mix the movie domain-specific tasks with samples from general instruction dataset. We train the model in a regular supervised finetuning setup using causal language model loss, with the difference that we mask out the loss on the instruction. That encourages the model to learn the narrative understanding tasks itself given the input rather than completing the input, which sometimes consists of movie text much longer than the expected output.

5 Experiments & Results

5.1 Experimental Setup

Dataset We split the samples for each task by movie titles using the train-test split from the MovieNet and MovieGraphs datasets. This prevents any information leakage from the model seeing the same movie content from different samples during the training run. We train models with a mixture of narrative understanding tasks and general instructions from LIMA (Zhou et al., 2024) and GPT-4-LLM (Peng et al., 2023). General instructions are sampled from these two datasets to the specified amount, with the order of the samples shuffled randomly. The number of samples for each individual task group can be found in Table 6.

Models We conducted the experiments on the instruction-finetuned version of the LLama3 and LLama3.1 family (Dubey et al., 2024), as it generally leads to better performance for continued instruction fine-tuning (Zhang et al., 2023b). We use a standard setting for fine-tuning and train each model for 3 epochs with a learning rate of 1e-5 and a weight decay of 1e-2 following AlpaGasus (Chen et al., 2023). We used a total batch size of 64 on 8 A100 GPUs with DeepSpeed Stage3 [3]. We keep multiple checkpoints for each setting for evaluation, as loss in the evaluation set is a poor indicator of the quality of text generation (Zhou et al., 2024).

Evaluation Metrics Story elements tasks that predict events (interaction, action) and existent (character, place) expect output that is usually a word or a phrase for the corresponding element. We use exact match to check if the ground-truth label is correctly generated by the model while ignoring the generated explanation. For interaction prediction, we extract the lemma of the root verb in the output and ground truth before matching.

For discourse tasks, we report traditional generation metrics including BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) for clip and story summarization. We report the average distance from the predicted index to the ground truth for the turning point prediction.

5.2 Domain Instruction Finetuning (DIT)

Training Method for DIT We first investigate what LLMs' zero-shot capability is on different narrative understanding tasks $\mathcal{T}$. We also examine how different training methods affect the domain adaptation procedure, including the parameter-efficient training (PEFT) method LoRA (Hu et al., 2021) and instruction embedding noise NEFTune (Jain et al., 2023). We show the average metrics for the story tasks and the discourse tasks in Table 2. The metrics for the individual tasks are available in the Appendix.

Model	Story	Discourse	
	Acc.	BLEU-2	RougeL
LLAMA3.1-8B	0.151	0.092	0.169
LLAMA3.1-70B	0.214	0.124	0.194
Finetune	**0.321**	**0.150**	**0.194**
LoRA-8-16	0.273	0.146	0.193
w/o NEFTune	0.304	0.147	0.191
LLAMA3-8B-Finetune	0.310	0.147	0.191

Table 2: Narrative tasks metrics for different ablations

Our results show that fine-tuning on narrative understanding tasks greatly improves models' performance on all training tasks. For LoRA finetuning, we test different LoRA rank and alpha with a dropout equal to 0.01 and a learning rate of 3e-4 following Alpaca-LoRA[4]. The LoRA finetuned model shows lower average accuracy on story tasks and generation metrics for the summarization tasks. Increasing the rank and alpha does not increase the results correspondingly either. Removing the embedding noise NEFTune from the instruction hurts the performance, so we keep it in all our following experiments.

We also compare the Llama3.1-8B model with the 70B version and its earlier variant Llama3. Although the 70B variant has a stronger zero-shot performance than the 8B model, it has a much

---

[3] https://github.com/microsoft/DeepSpeed

[4] https://github.com/tloen/alpaca-lora

Model	Story	Discourse		
	Acc.	BLEU-2	Rouge-L	Distance
Zero-shot	0.151	0.092	0.169	3.08
+Discourse	0.036	0.148	0.193	3.02
+Story	0.312	0.053	0.128	4.68
+Domain	0.317	0.149	0.193	2.79
+Domain +General	**0.322**	**0.150**	**0.194**	**2.52**

Table 3: Effect of data composition for Domain Instruction Finetuning

lower performance than the finetuned model. The Llama3 model appears to be a little worse than the more recent Llama3.1 after the same finetuning.

Effect of Data Composition for DIT We then look into what is the best data composition for the instruction finetuning. We finetune the model with all narrative understanding tasks from our dataset combined with general instructions and examine whether a different mix ratio affects the outcomes. To examine whether learning some narrative understanding tasks helps the others, we also trained the model with either the tasks for the story element or discourse.

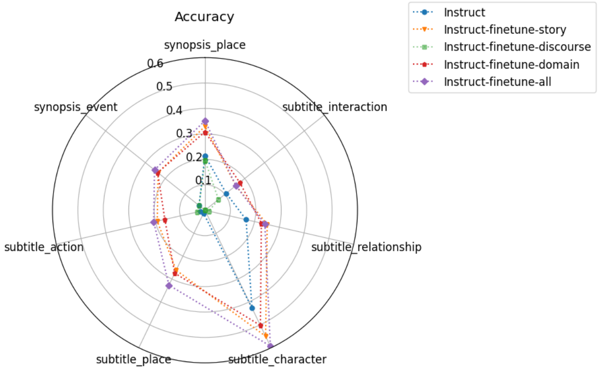


Figure 2: Story narrative understanding tasks performance trained with different data mixtures

Table 3 shows the model's average metrics on story element tasks and discourse tasks respectively, and the accuracies on individual prediction tasks are shown in Figure 2. While the general LLM fails to predict actions or places from the subtitle out-of-the-box, training only on discourse worsens the accuracy to close to zero. This suggests that training on story tasks or discourse tasks does not generalize to the other category of tasks since the model overfits to the specific input format and tasks. Finetuning on all movie domain tasks increases the performance across all tasks in both categories of tasks, while adding generation instructions benefits the model's performance as well. However, as shown in Figure 4, oversampling general instructions does not improve the model's training task performance any further.

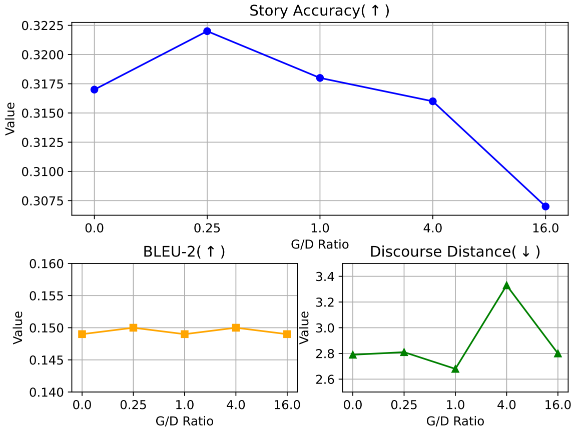


Figure 3: Effect of General/Domain instructions Ratio on narrative understanding tasks performance

Generalization on unseen tasks with DIT We check the performance on three out-of-distribution benchmarks to see how the finetuned model generalizes to unseen tasks $\mathcal{U}$ or maintains its original capability.

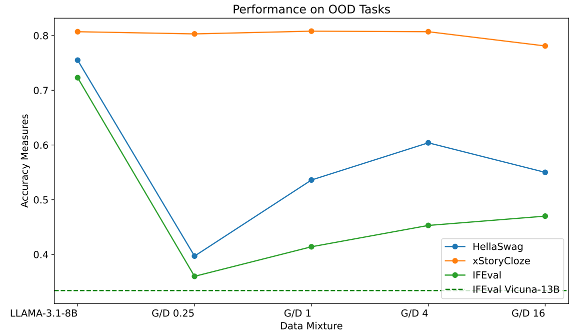


Figure 4: Models performance on out-of-domain tasks with different General/Domain instruction Ratio. The most left value are the base model without domain instruction finetuning.

Hellaswag (Zellers et al., 2019) and the English part of xStoryCloze (Lin et al., 2022) both test the model's commonsense natural language inference capability by letting it choose a sentence to finish a passage or story. Meanwhile, IFEval (Zhou et al., 2023) focuses mainly on the ability to follow various instructions. We obtain benchmarking results with the *lm-evaluation-harness* library from EleutherAI [5]. Figure 4 shows that the performance of xStoryCloze remains intact as a text completion task without complex formatting, while the performance on Hellaswag and IFEval drops dras-

---

[5] https://github.com/EleutherAI/lm-evaluation-harness

Model	Story Acc.	BLEU-2	Rouge-L	HellaSwag	xStoryCloze	IFEval
Baseline	0.151	0.092	0.169	**0.755**	**0.807**	**0.723**
DIT	0.322	**0.150**	**0.194**	0.397	0.803	0.360
DPT+DIT	**0.331**	0.144	0.187	0.277	0.772	0.349

Table 4: Effects of Domain Pretraining on Domain Instruction Finetuning. The metrics for narrative understanding tasks are the averaged results.

Model	NarrativeQA	Hate Speech	Political	LGBTQ	Religious
Zero-shot	0.190	0.771	0.450	0.692	0.839
SFT	0.373	0.931	0.922	**0.923**	0.934
DIT-SFT	**0.375**	0.928	0.915	0.922	**0.934**
DPT-DIT-SFT	0.369	**0.934**	**0.925**	0.920	0.933

Table 5: Results on downstream task supervised finetuning. BLEU-2 for narrativeQA. Accuracy for harmful content detection datasets.

tically after the instruction finetuning. The model regains some of the instruction-following capability with an increased amount of general instructions, but is not able to recover to the original level. This suggests the necessity of a high-quality and large enough general instruction dataset, for domain adaptation while maintaining the instruction-following capability. The reinforcement learning from human feedback stage could also potentially help.

5.3 Interplay of Domain Adaptation Stages

Impact of DPT on domain and unseen tasks We also examine whether DPT influences the effectiveness of subsequent domain instruction finetuning as in Wu et al. (2023). We train the LLM on all the movie scripts as a text completion task for one epoch, before conducting the same instruction finetuning as in previous setups.

The results in Table 4 show that although pretraining increases the accuracy of story element prediction, it decreases the performance on the discourse tasks, and it is detrimental to model's general instruction-following capability. This aligns with the results of studies in other domains (Wu et al., 2024).

Impact of domain adaptation on downstream tasks We further investigate whether the domain adaptation uniformly improves the model's performance on downstream tasks in the movie domain.

We compare the supervised finetuning results on several tasks in the movie domain with the same training setups, based on the baseline models (SFT) or the domain-adapted models respectively. We train the model using an internal dataset to classify harmful content from subtitles, including hate speech, references to political, LGBTQ, or religious content. We also train the model on NarrativeQA (Kočiský et al., 2018), which is a long-form generative QA based on the full text of a book or movie script. Example prompts for downstream tasks can be found in Table 9.

As suggested in Table 5, the performances on all tasks are effectively improved by supervised finetuning compared to the zero-shot setting, that includes the accuracy for *story* tasks and BLEU or Rouge for *discourse* tasks. However, the additional stages of domain instruction finetuning and pretraining do not bring a consistent improvement across all tasks. And it is still an open research question on how to effectively measure the correlation between different domain tasks and provide a wider coverage through the domain adaptation.

6 Conclusion

In this paper, we construct a movie-domain instruction dataset consisting of a suite of narrative understanding tasks inspired by Narrative Theory, and use it to analyze the effect of different domain adaptation stages.

We demonstrate that instruction finetuning on the movie domain effectively increases the model's performance on all narrative understanding tasks, but comes with trade-offs between general instruction finetuning capability. Additionally, we examine the interaction of instruction finetuning with domain pretraining and domain downstream tasks, revealing the benefits and limitations of adaptation approaches. Our findings provide insights into how LLMs can be effectively adapted to domains with complex storytelling structures, paving the way for future advancements in cinematic AI applications.

7 Limitations

We rely on existing annotated datasets as our source data. Despite quality control measures, there can be noise in the sample that we constructed, including mislabeling, etc. We run our experiments mainly on the Llama family of models, and the results can be further validated on other models. The training data may contain offensive content and is not examined by the authors.

References

Seymour Benjamin Chatman. 1980. *Story and discourse: Narrative structure in fiction and film*. Cornell university press.

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, et al. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Jiaxi Cui, Munan Ning, Zongjian Li, Bohua Chen, Yang Yan, Hao Li, Bin Ling, Yonghong Tian, and Li Yuan. 2024. Chatlaw: A multi-agent collaborative legal assistant with knowledge graph enhanced mixture-of-experts large language model.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Qingqiu Huang, Yu Xiong, Anyi Rao, Jiaze Wang, and Dahua Lin. 2020. Movienet: A holistic dataset for movie understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 709–727. Springer.

Neel Jain, Ping-yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, et al. 2023. Neftune: Noisy embeddings improve instruction finetuning. *arXiv preprint arXiv:2310.05914*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

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

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mingjie Liu, Teodor-Dumitru Ene, Robert Kirby, Chris Cheng, Nathaniel Pinckney, Rongjian Liang, Jonah Alben, Himyanshu Anand, Sanmitra Banerjee, Ismet Bayraktaroglu, et al. 2023. Chipnemo: Domain-adapted llms for chip design. *arXiv preprint arXiv:2311.00176*.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie plot analysis via turning point identification. 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 1707–1717, Hong Kong, China. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna

Ebrahimi, and Hao Wang. 2024. Continual learning of large language models: A comprehensive survey. *arXiv preprint arXiv:2404.16789*.

Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, pages 1–8.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. 2018. Moviegraphs: Towards understanding human-centric situations from videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022a. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. 2022b. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. 2024. Pmc-llama: toward building open-source language models for medicine. *Journal of the American Medical Informatics Association*, page ocae045.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2023. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*.

Hongyang Yang, Xiao-Yang Liu, and Chris Wang. 2023. Fingpt: Open-source financial large language models. *ArXiv*, abs/2306.06031.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023a. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Zihan Zhang, Meng Fang, Ling Chen, and Mohammad-Reza Namazi-Rad. 2023b. CITB: A benchmark for continual instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9443–9455, Singapore. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

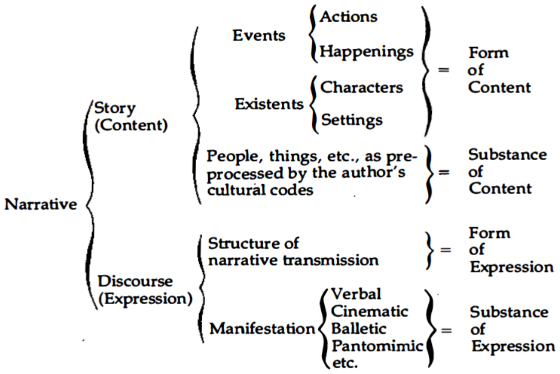


Figure 5: Elements of Narrative Theory (Chatman, 1980)

Task	samples
LIMA	1000
GPT-4-LLM	52002
subtitle place	328
subtitle action	214
subtitle character	367
synopsis place	795
synopsis event	928
subtitle relationship	233
subtitle interaction	165
Total Story	3030
clip subtitle synopsis	1017
story subtitle synopsis	1218
synopsis turning point	2217
Total Discourse	4452

Table 6: Statistics of instruction datasets used. The general instructions are sampled from LIMA and GPT-4-LLM

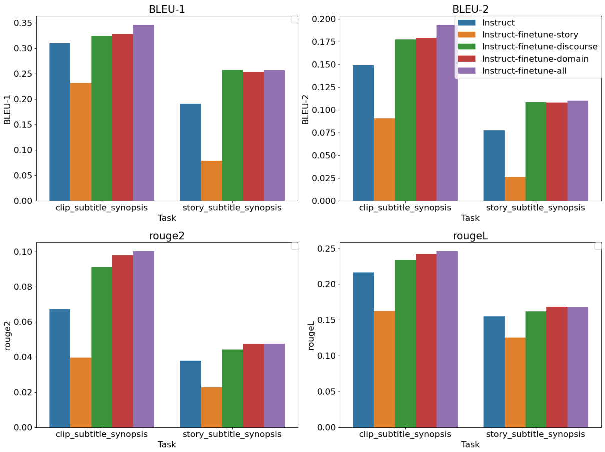


Figure 6: Discourse narrative understanding tasks performance trained with different data mixtures

Prompt

You are an expert in film theory and film criticism, with a deep understanding of cinema from various genres, eras, and cultures. You are good at analyzing films by examining narrative structure, cinematography, sound design, and thematic content. You can draw inferences from subtitles, should demonstrate a deep understanding of film as an art form. Focus on character development, plot progression, subtext, and cultural context. Provide objective and impartial analysis, avoiding personal biases.
You will be provided with {*task-specific label name*} and {*task-specific input format*}. Please determine if {*task-specific label name*} is implied or can be inferred from the {*task-specific input format*} and explain why.
Label: {*label*}
Input: {*input*}

Table 7: Prompt for LLM sample filtering. The *task-specific input format* includes the subscript, synopsis,and subtitle. The *task-specific label name* can be place, event, etc.

Task	Prompt
story_subtitle_synopsis	Your input will consist of subtitles from a scene in a movie. Please provide a concise synopsis that summarizes the content of the whole clip in a few sentences. ### Input: Subtitle: {subtitle} ### Response: {synopsis}
subtitle_action	You will be provided with subtitles of a shot from a movie and a background description of the story as the input. Try your best to imagine the plot of the movie shot, and predict the actions involved by the characters. Respond with the actions as a list of verb phrases. ### Input: Subtitle: {script} ### Response: {actions}
subtitle_character	You will be provided with subtitles and a list of characters of a movie clip, and a synopsis describing the background. Try your best to identify how the speakers correspond to the characters provided. Response with a json format with character as the key and corresponding speaker as the value. ### Input: Synopsis: {synopsis} Characters: {character} Subtitle: {masked_script} ### Response: {label}
synopsis_turning_point	You will be provided with a synopsis of a part of a movie with each sentence indexed. Predict the turning point between two story parts provided based on the plot. Answer with the sentence index only. ### Input: Synopsis: {synopsis} ### Response: {turning_point}

Table 8: Prompts used for different narrative understanding tasks. The content in brackets is replaced with the content from each sample. The rest of tasks use prompt similar to the provided ones with some adjusts on the wording.

Task	Prompt

narrativeQA	You will be given the synopsis of a movie and a question whose answer can be found in the movie. Answer the question concisely with a phrase or a short sentence.
	### Input: Question: {question}
	Synopsis: {synopsis}
	### Response: {answer}

hate speech	You will be given the caption of a movie clip. Determine if there it contains hate speech, answer either positive or negative.
	### Input: Caption: {caption}
	### Response: {final_label}

Table 9: Example prompts used for downstream classification. The content in the brackets are replaced with the content from each sample.

Subtitle	Script
00:00:13.502 –>00:00:14.662 Well, yes, sir. 00:00:14.736 –>00:00:16.260 It's right here. 00:00:16.338 –>00:00:17.999 I'm glad to see that... 00:00:18.073 –>00:00:21.008 because the rightful owner of this cross 00:00:21.076 –>00:00:24.068 won't press charges if you give it back. 00:00:24.146 –>00:00:27.638 He's got witnesses, five or six of them.	INDY Well, yes, sir. It's right here! SCENE *INDY shows the CROSS, more or less handing it to the SHERIFF to make his point. The Sheriff takes it casually.* SHERIFF I'm glad to see that... because the rightful owner of this Cross won't press charges, if you give it back. *FEDORA enters the house, followed by ROSCOE, ROUGH RIDER and HALFBREED. He politely removes his hat and holds it in his hand. He nods at INDY in a friendly manner.* SHERIFF He's got witnesses, five or six of them.

Table 10: Comparison of Subtitle and Script for the same movie clip

Task	Dataset	LLM kept ratio	Samples
subtitle_action	MovieNet	46.2%	1604
subtitle_place	MovieNet	46.0%	2031
subtitle_character	MovieGraphs	81.1%	3804
subtitle_interaction	MovieGraphs	87.8%	836
subtitle_relation	MovieGraphs	73.4%	712
synopsis_event	MovieGraphs	85.0%	4276
synopsis_place	MovieGraphs	78.3%	3937
short_subtitle_synopsis	MovieGraphs	-	4705
long_subtitle_synopsis	MovieNet	-	2366
synopsis_turning_point	MovieNet	-	2226

Table 11: Statistics on train tasks

From Stories to Statistics: Methodological Biases in LLM-Based Narrative Flow Quantification

Amal Sunny
IIIT-Hyderabad
`amal.sunny@research.iiit.ac.in`

Advay Gupta
IIIT-Hyderabad
`advay.gupta@research.iiit.ac.in`

Yashashree Chandak
Stony Brook University
`ychandak@cs.stonybrook.edu`

Vishnu Sreekumar
IIIT-Hyderabad
`vishnu.sreekumar@iiit.ac.in`

Abstract

Large Language Models (LLMs) have made significant contributions to cognitive science research. One area of application is narrative understanding. Sap et al. (2022) introduced *sequentiality*, an LLM-derived measure that assesses the coherence of a story based on word probability distributions. They reported that recalled stories flowed less sequentially than imagined stories. However, the robustness and generalizability of this narrative flow measure remain unverified. To assess generalizability, we apply *sequentiality* derived from three different LLMs to a new dataset of matched autobiographical and biographical paragraphs. Contrary to previous results, we fail to find a significant difference in narrative flow between autobiographies and biographies. Further investigation reveals biases in the original data collection process, where topic selection systematically influences sequentiality scores. Adjusting for these biases substantially reduces the originally reported effect size. A validation exercise using LLM-generated stories with "good" and "poor" flow further highlights the flaws in the original formulation of sequentiality. Our findings suggest that LLM-based narrative flow quantification is susceptible to methodological artifacts. Finally, we provide some suggestions for modifying the *sequentiality* formula to accurately capture narrative flow.

1 Introduction

The advent of Large Language Models (LLMs) over the last few years has fundamentally changed the landscape of language and cognitive research as we know it (Zhao et al., 2023). These models have gotten sophisticated to the point where the debate now is whether they display emergent properties (Wei et al., 2022). LLMs can solve real-world problems ranging from code synthesis (Nijkamp et al., 2022) to lie detection (Loconte et al., 2023). Such advances have further motivated research using these models to gain insight into human behavior and cognition (Demszky et al., 2023; Mihalcea et al., 2024). While LLMs are impressive, rigorous use of statistics and better metrics have previously demonstrated that initial claims about their capabilities were overblown (Schaeffer et al., 2023). Narrative understanding is one important area of human cognition that finds a natural application for LLMs. In this study, we rigorously assess a recently proposed LLM-derived measure of narrative flow (Sap et al., 2022) and find that methodological biases drive a large part of the original results. We discuss conceptual issues with the original formulation and propose some ways to address these biases. We expect that our analytical approach comprising both conceptual and direct replications, with appropriate randomization procedures in the evaluation pipeline, will serve as a template for future assessments of LLM-derived measures of human cognition.

Narrative understanding has historically been hindered due to difficulty in quantifying narrative elements in texts at scale (Toubia et al., 2021). Sap et al. (2022) introduced *sequentiality*, a measure of how well an LLM predicts sentences in a narrative based on the preceding sentences and the topic (i.e., *context*), compared to predictions based on the *topic* alone. This relative measure, averaged over all sentences in a story, characterizes the narrative flow, that is, the organization of the sequence of events and how well they progress from one to the next. This formulation was a breakthrough as the measure considered the entire story in its whole context, whereas previous methods relied on either detecting certain words from sentences (Sims et al., 2019; Li et al., 2013) or tracking particular elements over time in stories (e.g., sentiment, emotion, categories of words, or sentence embeddings (Reagan et al., 2016; Boyd et al., 2020; Toubia et al., 2021)). Specifically, Sap et al. (2022) captured signatures of episodic memory retrieval in narra-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 201–215
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

tives by contrasting the sequentiality of a “recalled” narrative (a story that happened to person A) versus an “imagined” narrative (a story that person B made up given the topic of person A’s recalled narrative). Recalled stories were less sequential compared to imagined ones, suggesting that spontaneously recalled stories deviate more from the schema of the event due to recalled events possibly triggering memories of other events. Imagined stories, on the other hand, stick to themes that are predictable from the event schema. The work of Sap et al. (2022) was seminal, inspiring multiple studies that directly incorporated their sequentiality metric (Cohen et al., 2025; Cornell et al., 2023) and numerous others (Demszky et al., 2023; Mihalcea et al., 2024; Piper et al., 2023) that built upon its findings as a foundational result.

However, LLM-based research is prone to biases (Gallegos et al., 2024; Zhou et al., 2024). A large body of recent NLP research is dedicated to evaluating biases already inherent in LLMs in different domains (Yeh et al., 2023; Huang et al., 2024) and to evaluating biases in methods and metrics that utilize LLMs (Lin et al., 2025; Hu et al., 2024; Zheng et al., 2023; Ni et al., 2024). Broadly, we can classify the biases in the field as i) methodological ones - where the bias exists in how the methods were framed, overlooking some aspects of the problem or favoring certain assumptions, and ii) data-driven biases, which arise from the dataset used and manifest in ways specific to the task at hand, influencing model predictions based on patterns, imbalances, or artifacts present in the underlying datasets (Yeh et al., 2023). We propose to evaluate Sap et al. (2022) for methodological biases, by evaluating their *sequentiality* metric as is on an entirely different curated dataset of matched autobiographies and biographies (analogous to “recalled” and “imagined” stories from Sap et al. (2022)) to see if we can capture the same difference across the two groups demonstrating properties of episodic memory retrieval. A successful replication of the results on a dataset from a conceptually similar domain would ameliorate concerns of bias by demonstrating generalizability.

To further assess methodological biases in Sap et al. (2022), we examine the generalizability of their findings across different LLMs. Since these prior experiments were conducted, the LLM landscape has evolved rapidly, with the development of significantly more advanced models (Zhao et al., 2023). These newer models have been trained on larger and more diverse datasets spanning multiple domains and incorporate key advancements such as alignment techniques (Rafailov et al., 2023; Ouyang et al., 2022). As a result, they may offer a more human-representative estimation of *sequentiality*. We experiment with several open-source and cutting-edge models such as LLaMa-3.1 (Grattafiori et al., 2024), Qwen-2.5 (Yang et al., 2024) and Falcon3 (Team, 2024) by first replicating the analysis on the original Sap et al. (2022) data to ensure parity.

Then, we consider the possibility that data-driven biases drive the original *sequentiality* results. Models applied to biased data will produce biased outcomes. We need to ensure that there are no confounds in the data that can explain the results before we make claims about complex measures such as narrative flow being a useful metric for various downstream applications. We discover a possible data-driven bias in how topics are collected and find that this bias directly influences the results. On correcting this data-driven bias using randomization techniques, we find a significantly decreased effect size from the original finding.

Apart from data-driven biases, there are concerns about the formulation itself that contains both a topic-driven term and a contextual term (that incorporates the topic in it as well). Sap et al. (2022) did not attempt to validate the formula on stories that are known to have good/poor narrative flow. Therefore, to further assess the *sequentiality* formulation, we conduct a small-scale experiment of generating stories that exhibit visibly good and poor narrative flow and apply the sequentiality measure to these stories. We do not find the expected sequentiality difference between the two group of stories. This further reinforces our concerns with the *sequentiality* formulation.

Our contributions in this study are fourfold - 1) we curate a dataset of matched autobiographical and biographical accounts and attempt to *conceptually* replicate Sap et al.’s (2022) sequentiality measure; 2) we *directly* replicate their results on the original dataset using more modern LLMs to ensure generalizability; 3) we demonstrate a methodological bias that when corrected leads to a much smaller effect size than originally claimed; 4) we demonstrate that even after removing the methodological bias emanating from the topic, the formulation does not work for stories generated with explicit good and poor narrative flow. We conclude that *sequentiality*, as originally formulated, is not an appropriate measure to compute narrative flow of a story.

2 Methods

2.1 Datasets

2.1.1 Hippocorpus

Hippocorpus[1], used in Sap et al.'s (2022) original study, consists of 6854 stories collected by crowd sourcing from Amazon Mechanical Turk (MTurk) human workers. The dataset consists of three different categories of stories (recalled, retold, and imagined), but we only use two - the recalled and imagined stories. The recalled group consists of stories written by the workers ranging from 15-25 sentences about a memorable or salient event that they experienced in the past 6 months. The same workers also provided 2-3 sentence summaries that served as the topics of these stories. The imagined group consists of stories written by another set of workers who are given the summaries from the recalled group and told to write an imagined story about the same topics.

Hippocorpus contained more than one imagined story for some topics. We restrict our analysis to topics that had exactly one recalled and one imagined story. We obtained 2395 such matched recalled-imagined story pairs. The recalled stories had an average of 18.5 sentences and 277.5 words and the imagined stories had an average of 17.7 sentences and 240.1 words.

2.1.2 Autobiography-Biography Dataset

We collected autobiography-biography book pairs on the lives of 63 distinct individuals (126 books in total). Paragraphs in the books were embedded using *gte-base-en-v1.5* (Zhang et al., 2024). We used these embeddings to match paragraphs for thematic content across the autobiography and biography of any given personality to obtain autobiographical and biographical narratives of the same events. We retained auto-bio paragraph pairs with cosine similarity > 0.7 ensuring alignment of thematic content, while avoiding verbatim overlap using a ROUGE-L threshold < 0.4 (Lin, 2004). An auto-bio book pair was retained only if it contained > 25 pairs of paragraphs meeting the above criteria for semantic similarity. 4175 story pairs from 38 pairs (76 books) matched all the criteria and were retained for further analysis. The autobiographical paragraphs contained an average of 7.0 sentences and 116.8 words and the biographical paragraphs contained an average of 8.2 sentences and 136.0 words. Examples of matched auto-bio paragraph pairs and information about the books retained in the dataset can be found Appendix A.

2.1.3 Synthetic "Good" and "Poor" Flow Stories for Validating Sequentiality

Given that the sequentiality formulation was proposed but not validated using "ground-truth" stories by Sap et al. (2022), we attempt to validate the measure by prompting an LLM to generate stories with "good flow" and "poor flow"(see Appendix E for examples of generated stories and Appendix B.3 for the prompt used). We used *Mistral-7B-Instruct-v0.2-AWQ* (Jiang et al., 2023) (henceforth *Mistral*) to generate these stories, given randomly sampled topics from the Hippocorpus dataset. We manually verified and filtered the generated stories to ensure agreement with their respective flow labels ("good" or "poor") before computing their sequentiality scores.

As illustrated in Fig. 1, our analytical approach begins by prompting a high-performing LLM to generate a topic for each paragraph before computing the *sequentiality* of each paragraph. The details of topic generation and the sequentiality formulation are provided below.

2.2 Topic Generation

Topics were generated using *Mistral* for a given paragraph/story. The model was given a structured prompt containing a definition of topic as the "main idea of a paragraph," an example paragraph and topic (see Appendix B.1 for the example and prompt), and an instruction to return the topic in one to two sentences. We refer to this strategy as "one-shot prompting" and use this as the default topic generation strategy throughout this study unless specified otherwise. A small minority of the responses ($< 5\%$) did not strictly adhere to the instructions to return just the topic. The additional paragraphs generated, providing additional context or justification for the choice of the topic, were discarded, retaining only the generated topic.

As we will see in the next subsection, the sequentiality formula has a topic-driven term and a contextual term. In Sap et al.'s (2022) dataset, the topics are written only by the recalled group and they report that sentences in the recalled stories are better predicted by the topic compared to the sentences in

---

[1] `https://www.microsoft.com/en-us/download/details.aspx?id=105291`

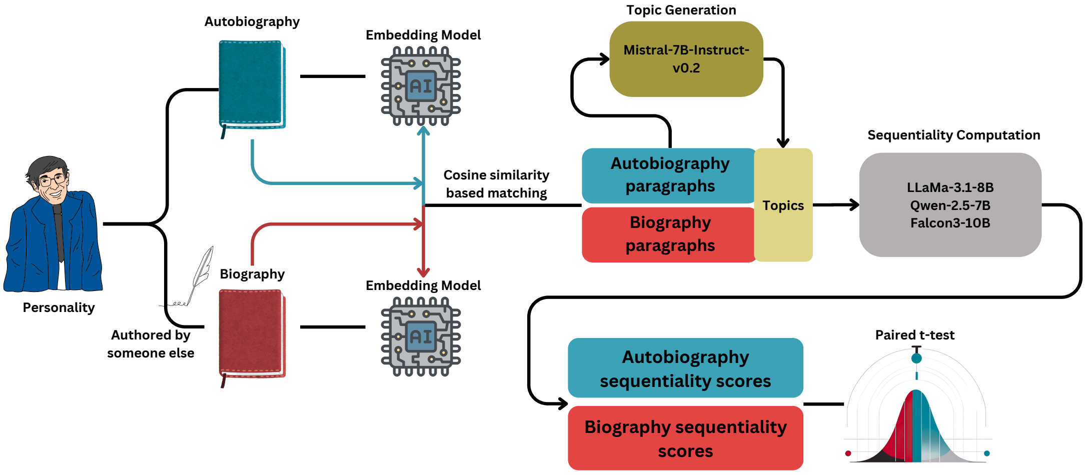


Figure 1: Overview of the methodology and analysis pipeline.

the imagined stories. Critically, they report no significant differences in the contextual term between recalled and imagined stories. To evaluate potential topic-driven bias in these results, we select topics in three different ways in our direct replication: i) exclusively generated from the autobiographical/recalled stories, ii) exclusively generated from the biographical/imagined story, and iii) generated from a story that is randomly selected from the autobiography/recalled or biographical/imagined group. The third approach guards against potential bias due to selecting topics from only one group.

To ensure that our results are not specific to the topic generation strategy described above, we repeat the analysis using an alternative method. In this approach, we generate topics using a zero-shot strategy, where the model receives only a prompt without examples, based on the instructions in Sap et al. (2022) (see Appendix B.2 for the prompt). We evaluate the effect of this strategy on sequentiality difference across groups in the most unbiased condition (i.e., randomly selecting topics from either group) and compare it against the results we obtained with the default one-shot strategy.

2.3 Sequentiality

Following Sap et al. (2022), we use *sequentiality* to quantify the extent to which the ideas/sentences in a story proceed in a well-organized sequence. Sequentiality of a sentence s_i compares the likelihood of the sentence under two probabilistic models: i) a **topic-driven model**, which assumes that the sentence is generated conditioned only on the overarching theme (topic) of the story, denoted by $\mathcal{T}$, and ii) a **context-driven model**, which assumes that the sentence depends on both the theme $\mathcal{T}$ and the preceding sentences $s_{0:i-1}$.

Sequentiality of s_i is computed as the difference in negative log-likelihoods between the topic ($\text{NLL}_{\mathcal{T}}$) and contextual ($\text{NLL}_{\mathcal{C}}$) models:

$$\Delta\ell(s_i) = -\frac{1}{|s_i|}\Big[\underbrace{\log p_{LM}(s_i \mid T)}_{\text{topic-driven}} - \underbrace{\log p_{LM}(s_i \mid T, s_{0:i-1})}_{\text{contextual}}\Big], \tag{1}$$

where the log-probability of a sentence s given some context C (e.g., topic T and preceding sentences $s_{0:i-1}$) is the sum of log-probabilities of its tokens w_t given the same context: $\log p_{LM}(s|C) = \sum_t \log p_{LM}(w_t|C, w_{0:t-1})$; and the likelihoods are normalized by sentence length $|s_i|$. Finally, the sequentiality of a paragraph is computed by averaging the sequentiality of all the sentences in the paragraph. Higher values of sequentiality are taken to indicate that sentences are highly predictable from the topic and context of the unfolding story whereas lower values indicate greater deviation from the ideas predicted by the preceding sentences. However, we note here that true sequentiality differences between stories should be driven primarily by $\text{NLL}_{\mathcal{C}}$ with $\text{NLL}_{\mathcal{T}}$ providing a "baseline" topic-based likelihood. If results are primarily driven by significant differences in the topic-based likelihood with no differences in the contextual likelihood, as in Sap et al. (2022), the measure would be incongruent with the intuitive concept of "sequentiality" as a measure of flow from the preceding context. Sap et al. (2022)'s main argument seems to rest on the difference in effect sizes between the overall sequentiality measure and the topic-term. Therefore, we compute the effect sizes associated with overall sequentiality, NLL$_\mathcal{T}$, and NLL$_\mathcal{C}$ to make our arguments.

We estimated the likelihoods using three different models, (i) *Meta-Llama-3.1-8B-Instruct-AWQ-INT4* (henceforth Llama-3.1), (ii) *Qwen2.5-7B-Instruct-AWQ* (henceforth Qwen-2.5), and (iii) *Falcon3-10B-Instruct-AWQ* (henceforth Falcon3), all of which are trained on extensive high-quality text corpora featuring web, code, STEM, and curated high-quality and multilingual data. We used these different models to test whether results are model-dependent. We conduct our main analysis with the model producing the closest direct replication of Sap et al.'s (2022) inferences on their own dataset.

2.4 Statistical Analysis

For comparing sequentiality and its constituent terms across the two groups, we use a paired t-test and report the t-statistic, p-value, degrees of freedom (df) and Cohen's d (effect size). We emphasize Cohen's d over p-values because given a dataset of this size, statistical significance could be trivial if not weighted properly with the corresponding effect size (Sullivan and Feinn, 2012).

3 Conceptual Replication

We computed the sequentiality of matched pairs of autobiographical and biographical paragraphs in our curated dataset and compared them using a paired t-test. We randomly pick one of the two paragraphs from a given pair of paragraphs to pass to an LLM to generate a topic for the sequentiality computation. We find no significant sequentiality differences between topic-matched biographical and autobiographical paragraphs ($t = 0.11, p = 0.90, d = 0.001, df = 4174$) using LLaMa-3.1 (Fig. 2). Even after aggregating the scores for each personality, we do not find a significant difference across biographies and autobiographies ($t = -1.22, p = 0.23, d = 0.19, df = 37$).

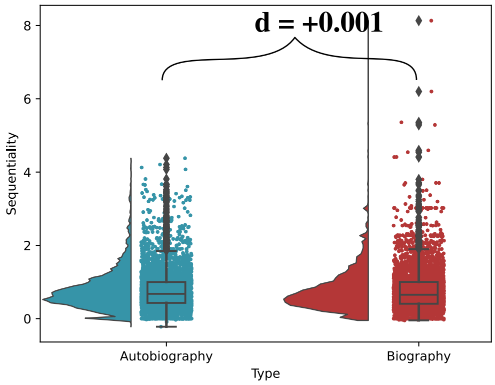


Figure 2: Distribution of sequentiality scores for topic-matched autobiographical and biographical paragraphs. d represents Cohen's d (effect size)

Model	t-stat	p-val	Cohen's d
LLaMa-3.1	0.11	0.90	0.001
Falcon3	3.05	*0.001*	0.047
Qwen-2.5	0.76	0.44	0.014

Table 1: Results of a paired t-test comparing sequentiality of biographical and autobiographical stories across three different LLMs.

To evaluate whether this result was specific to the LLM we used, we experimented with two other high-performing LLMs, *Qwen-2.5* and *Falcon3* and report our results in Tab. 1. All the models display either no statistically significant differences between the groups or a statistically significant difference but with a negligible effect size. Clearly, none of the models replicate the large differences reported in Sap et al. (2022). Our conceptual replication could have failed due to one or more of three reasons - 1) the autobiographies in our dataset have potentially undergone heavy editing (or ghost writing) and multiple passes by the author resulting in a more narrativized version that doesn't contain any trace of autobiographical memory retrieval, 2) our implementation of sequentiality calculation is flawed, 3) the original analysis was biased in some way. While it is difficult to evaluate the effect of editing in the dataset, our implementation can be verified by directly applying it to the Hippocorpus dataset to replicate Sap et al.'s (2022) results. Furthermore, as part of this direct replication, we can also examine the impact of slightly different but conceptually valid methodological choices on the original results.

4 Direct Replication

4.1 Original Topics

To verify that the failure to replicate the original result is not due to an implementation difference/error or due to specificity of the LLM used for the task, we compute sequentiality of the stories in the original Hippocorpus dataset. Using the same story pairs and topics they collected, we replicate their finding that imagined stories flow more sequentially than recalled stories

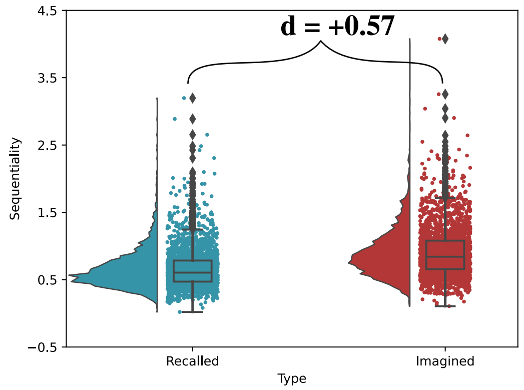


Figure 3: Distribution of sequentiality scores for recalled and imagined stories, given human-generated topics from the Hippocorpus dataset. d represents Cohen's d (effect size)

($t = 28.29, p < 0.001, d = 0.57, df = 2394$) using LLaMa-3.1 (Fig. 3). We also examine the difference between the topic-driven ($t = -19.48, p < 0.001, d = 0.39, df = 2394$) and contextual terms ($t = -2.24, p = 0.02, d = 0.04, df = 2394$) from Eq. (1) and, similarly to the original study, find a sizeable difference in the topic-driven term but not in the context-driven term. We repeat the analysis using different LLMs (Qwen-2.5 and Falcon3) and report similar results in Tab. 2.

These results indicate that the sequentiality measure generalizes to different LLMs and verifies that our implementation of sequentiality works. However, this "replication" does not fully alleviate concerns about the formulation or the potential for bias driving the original results. Sap et al. (2022) reported that the difference between the two groups was predominantly driven by the topic-driven term in Eq (1) and not the contextual term. While we replicated the same patterns, we also notice that the contextual term shows a slight difference in the opposite direction, i.e., recalled sentences flow better from the context of the unfolding story compared to the sentences in the imagined stories. This observation, combined with the fact the topic for these stories are provided only by the recalled group and not the imagined group, points towards a potential source of bias. To further assess this potential bias, we replaced the human-generated topics/summaries with LLM-generated topics but using i) only the recalled stories, ii) only the imagined stories, and iii) stories randomly sampled from the recalled and imagined conditions as input for topic generation. We expect these three conceptually equivalent ways of picking topics to yield similar results if there is no bias from the topic term.

4.2 Different Topic Generation Strategies

We find drastically different sequentiality patterns depending on how the topics are generated. We report results for LLaMa-3.1 since it displayed the largest difference (i.e. replicated Sap et al.'s (2022) results the best) between the groups in the previous section. We report results from the other models in Appendix C.

On generating topics using the imagined stories, which is a valid way of choosing topics since both recalled and imagined stories are about the same themes, we find that recalled stories are more sequential than imagined ones ($t = -4.91, p < 0.001, d = 0.10, df = 2394$; Fig. 4a), completely flipping the direction of the original results. The topic-driven differences ($t = 5.63, p < 0.001, d = 0.11, df = 2394$) and context-driven differences ($t = 3.11, p = 0.001, d = 0.06, df = 2394$) have also changed directions, compared to the results in Tab. 2.

On the other hand, when generating topics using the recalled stories as in the original work, we find that the imagined stories flow more sequentially than the recalled ones ($t = 18.27, p < 0.001, d = 0.37, df = 2394$, Fig. 4b), albeit the effect size is smaller than when using the original human-generated topics ($d = 0.37$ vs $d = 0.57$). We also observe that the directions of the topic-driven ($t = -15.44, p < 0.001, d = 0.31, df = 2339$) and context-driven ($t = -3.59, p < 0.001, d = 0.07, df = 2360$) differences replicate what were originally reported. Critically, we note here that the effect size differences between the overall sequentiality measure ($d = 0.37$) and topic-driven NLL ($d = 0.31$) are not as stark as the original findings, likely due to the topics being generated by an LLM rather than the same humans who recalled the stories. This result already calls into question the validity of the overall sequentiality measure if the observed differences are driven almost entirely by the topic term.

A conceptually equivalent way of generating topics completely flipping the results reported originally strongly indicates bias from the topic-driven term in the sequentiality measure. When the same people who recalled events from their lives also generate summaries that are used as the topics in the analysis, it should be expected that the sentences in the recalled stories would be significantly

Model	Metric	t-stat	p-val	Cohen's d	NLL_I	NLL_R
LLaMa-3.1-7b	Topic-driven $(-)$	-19.48	10^{-78}	0.39	-3.39	-3.12
	Context-driven $(+)$	-2.24	0.025	0.04	-2.48	-2.46
	Sequentiality $(+)$	**28.29**	$\mathbf{10^{-152}}$	**0.57**	0.90	0.66
Falcon3-10b	Topic-driven $(-)$	-17.41	10^{-64}	0.35	-3.81	-3.51
	Context-driven $(+)$	-2.62	0.008	0.05	-2.69	-2.66
	Sequentiality $(+)$	23.89	10^{-113}	0.48	1.12	0.85
Qwen-2.5-7b	Topic-driven $(-)$	-18.73	10^{-73}	0.38	-3.52	-3.24
	Context-driven $(+)$	-2.62	0.008	0.05	-2.59	-2.56
	Sequentiality $(+)$	27.04	10^{-141}	0.55	0.93	0.67

Table 2: Sequentiality comparison of imagined and recalled stories using different LLMs and human-generated topics taken from the Hippocorpus dataset. NLL_I & NLL_R are the mean values for the negative log likelihood of imagined and recalled stories, respectively. $(+)$ and $(-)$ indicate the expected direction of difference for the metric to replicate Sap et al.'s (2022) result. For example, Sequentiality $(+)$ indicates that a positive value in that row replicates the finding that imagined stories flow more sequentially than recalled ones. LLaMa is chosen for further analyses based on the strongest replication effects identified in **bold**. Additionally, our requirement that the contextual term drives the effect is also indicated by a $(+)$. Extremely small p-values are approximated to the closest power of 10.

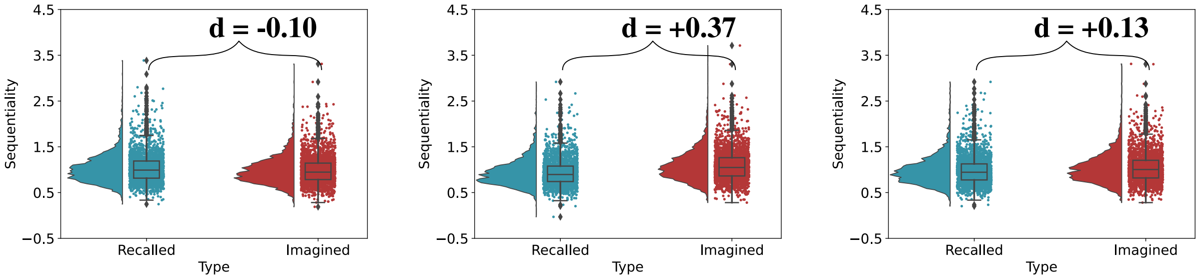


Figure 4: Distribution of sequentiality scores for recalled and imagined stories, given topics generated from the imagined (a), recalled (b), and random stories (c), respectively. A positive and large effect size (**d**) replicates Sap et al. (2022).

better predicted from the topics than those in the matched imagined stories. Since the topic also goes into the contextual term in Eq (1), the bias not only affects the topic-driven term but also the contextual term, explaining why the components also change directions when generating topics differently. However, the topic-driven term (which contributes negatively to the sequentiality formula) exerts a stronger influence and drives a decrease in sequentiality for the group from which topics were generated and trivially explains the original result of Sap et al. (2022).

We address the topic-driven bias by randomly sampling a topic either from the recalled or the imagined story. Now, we find that while imagined stories are still more sequential than recalled ones ($t = 6.62, p < 0.001, \boldsymbol{d = 0.13}, df = 2394$; Fig. 4c), the difference has a much lower effect size compared to the difference obtained by using the original human-generated topics ($d = 0.13$ vs $d = 0.57$, respectively). Furthermore, there is no context-driven difference ($t = -0.23, p = 0.81, d = 0.004, df = 2394$) and sequentiality difference is almost entirely topic-driven ($t = -4.56, p < 0.001, d = 0.09, df = 2394$). Repeating this analysis with a different topic generation strategy yields similar results (see Appendix D for details). Yet again, when not using human-generated biased topics, the effect sizes of the topic-driven term and the overall sequentiality measure are comparable, unlike those reported in Tab. 2, challenging Sap et al.'s (2022) claim that the overall measure explains narrative flow differences to a much greater extent than the topic-only term.

5 Sequentiality in LLM-generated Narratives

All experiments conducted thus far indicate a bias in the formulation primarily from the topic term. Here, we validate the sequentiality measure by applying it on a synthetic dataset of stories with "good" and "poor" flow (see Sec. 2.1.3).

Surprisingly, sequentiality of the stories generated to have good flow is lower than those with poor flow ($t = -12.59, p < 0.001, d = 0.51, df = 597$). Critically, we find that the contextual term behaves as we would expect intuitively: stories with good flow have a higher contextual likelihood than those with poor flow ($t = 10.39, p < 0.001, d = 0.42, df = 597$). However, the topic-driven likelihood ($t = 14.93, p < 0.001, d = 0.61, df = 597$) cancels out the contextual effect and drives the overall measure in the opposite direction. Therefore, in this validation exercise as well, the topic-driven term is the driving force masking an actual observable and meaningful effect from the contextual term. These results suggest that the topic-driven sentence likelihoods are meaningful since even LLMs generate stories that are more predictable from the topics under a "good flow" instruction. However, a standalone topic term in the formula may be masking effects that are adequately captured by the contextual term, which already incorporates the topic as part of its context. Therefore, a modified sequentiality formula with only the contextual term may be a more appropriate measure of narrative flow, especially when the comparison is between stories about the same topics.

6 Conclusion

To summarize, in the current study, we curated a novel dataset of matched autobiographies and biographies and applied Sap et al.'s (2022) sequentiality measure but using more modern open-source LLMs and failed to find a significant difference in narrative flow between the two. We then directly replicated Sap et al.'s (2022) results using the same LLMs applied to the original dataset. To understand why the narrative flow differences were so stark in Sap et al.'s (2022) data but not in our carefully curated dataset of matched autobiographical and biographical paragraphs, we analyzed both the data and the formula in (Sap et al., 2022) and report a bias in how the topics were collected which directly influenced the results through the narrative flow formula. We corrected for this bias and found a significantly reduced effect size for Sap et al.'s (2022) original finding. We further curated an LLM-generated dataset of stories with good and poor narrative flow and demonstrated the same topic-driven bias that pushed sequentiality in the opposite direction from the expected pattern. However, the contextual term captured the expected difference in narrative flow, suggesting that the sequentiality formula modified to have only the contextual term may be adequate. Importantly, this modified formulation would indicate that there are no real narrative flow differences between recalled and imagined stories, contrary to Sap et al.'s (2022)'s claims. We confirmed that our findings were not influenced by the specific LLM we used and conclude that LLM-based sequentiality, as originally formulated, is not a suitable metric for analyzing narrative flow.

These results have direct implications for studies that use sequentiality as a measure. Cohen et al. (2025) evaluated the relationship between the readability of medical texts and sequentiality. Rather than considering sequentiality as a whole, they analyzed the topic term and contextual term separately. Their results showed that the contextual term (referred to as the chain model) performed best and was primarily used in their analysis. This approach avoids topic bias and provides further evidence in favor of a modified formulation that includes only the contextual term. Cornell et al. (2023) conducted a similar study to Sap et al. (2022), comparing sequentiality across groups, with the addition of a new group consisting of generated stories based on topics from Sap et al.'s (2022) dataset. Their goal was to evaluate LLM-generated storytelling in comparison to human storytelling to better understand underlying memory processes. They reported multiple significant differences across groups (imagined, recalled, generated zero-shot, generated few-shot), but only in terms of overall sequentiality, not its components (topic and contextual terms). These differences may stem from how closely the stories align with the topic rather than from the overall cohesion of the story and its ideas, indicating the need for further investigation.

More generally, we recommend that future research on LLM-derived measures adopt proper randomization of aspects (such as topic-generation) that enter the formula and can potentially bias results. It is also important to provide an independent validation of such measures before they are adopted widely, to prevent accumulating biased methods of assessing/measuring human cognition from LLMs that have been argued to be cognitively implausible (Connell and Lynott, 2024).

Limitations

Due to computational limitations, we were unable to evaluate the generalizability of the sequentiality metric on larger models. However, given that the disparity in computational capabilities across model sizes has been narrowing and that all the models we utilized surpass GPT-3—the largest model examined by (Sap et al., 2022), this limitation may be less consequential. Furthermore, we were able to reproduce the results across all the models we considered, ameliorating concerns about generalizability of our findings. We were also unable to extensively evaluate different LLMs for topic generation. However, we found that our topic generation methods replicated the original results, when applied to the recalled stories, and also helped demonstrate the issue of topic-driven bias. While we speculated that the contextual term in the formula by itself should work as a measure of narrative flow, a finer-grained investigation of how the sequentiality formula can be modified, possibly by incorporating weights for the topic and contextual terms, may be warranted but is beyond the scope of the current work. A more carefully curated synthetic or actual dataset with ground-truth flow is necessary to accurately assess measures of narrative flow. We attempted to validate the sequentiality measure using stories generated by an LLM to have good and poor flow and showed that the contextual term can indeed capture narrative flow differences. However, further validation of the idea that the contextual term is sufficient by itself through more elaborate experiments is required.

Code and Data

The code and data extraction methods are available at `https://github.com/mandalab/narrative-flow-autobio`. To recreate the autobiography-biography dataset, we provide a pipeline to extract and match paragraphs from the books, provided a text file version of all these books. Other datasets can be accessed from their referenced locations.

Acknowledgments

We would like to thank Aruneek Biswas, Pooja R, Pritha Ghosh, and Gargi Shukla for providing useful feedback on earlier versions of the manuscript. We also thank IIIT Hyderabad for the Faculty Seed Fund (VS, IIIT/R&D Office/Seed-Grant/2021-22/018).

References

Ryan L Boyd, Kate G Blackburn, and James W Pennebaker. 2020. The narrative arc: Revealing core narrative structures through text analysis. *Science advances*, 6(32):eaba2196.

Trevor Cohen, Weizhe Xu, Yue Guo, Serguei Pakhomov, and Gondy Leroy. 2025. Coherence and comprehensibility: Large language models predict lay understanding of health-related content. *Journal of Biomedical Informatics*, 161:104758.

Louise Connell and Dermot Lynott. 2024. What can language models tell us about human cognition? *Current Directions in Psychological Science*, 33(3):181–189.

Charlotte Cornell, Shuning Jin, and Qiong Zhang. 2023. The role of episodic memory in storytelling: Comparing large language models with humans. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, et al. 2023. Using large language models in psychology. *Nature Reviews Psychology*, 2(11):688–701.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xinyu Hu, Mingqi Gao, Sen Hu, Yang Zhang, Yicheng Chen, Teng Xu, and Xiaojun Wan. 2024. Are llm-based evaluators confusing nlg quality criteria? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9530–9570.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Yang Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. Position: TrustLLM: Trustworthiness in large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20166–20270. PMLR.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Boyang Li, Stephen Lee-Urban, George Johnston, and Mark Riedl. 2013. Story generation with crowdsourced plot graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 598–604.

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

Luyang Lin, Lingzhi Wang, Jinsong Guo, and Kam-Fai Wong. 2025. Investigating bias in llm-based bias detection: Disparities between llms and human perception. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10634–10649.

Riccardo Loconte, Roberto Russo, Pasquale Capuozzo, Pietro Pietrini, and Giuseppe Sartori. 2023. Verbal lie detection using large language models. *Scientific reports*, 13(1):22849.

Rada Mihalcea, Laura Biester, Ryan L Boyd, Zhijing Jin, Veronica Perez-Rosas, Steven Wilson, and James W Pennebaker. 2024. How developments in natural language processing help us in understanding human behaviour. *Nature Human Behaviour*, 8(10):1877–1889.

Jinjie Ni, Fuzhao Xue, Xiang Yue, Yuntian Deng, Mahir Shah, Kabir Jain, Graham Neubig, and Yang You. 2024. Mixeval: Deriving wisdom of the crowd from llm benchmark mixtures. *arXiv preprint arXiv:2406.06565*.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Haiquan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2022. Codegen: An open large language model for code with multi-turn program synthesis. In *International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Andrew Piper, Hao Xu, and Eric D Kolaczyk. 2023. Modeling narrative revelation. In *CHR*, pages 500–511.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Andrew J Reagan, Lewis Mitchell, Dilan Kiley, Christopher M Danforth, and Peter Sheridan Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ data science*, 5(1):1–12.

Maarten Sap, Anna Jafarpour, Yejin Choi, Noah A. Smith, James W. Pennebaker, and Eric Horvitz. 2022. Quantifying the narrative flow of imagined versus autobiographical stories. *Proceedings of the National Academy of Sciences*, 119(45):e2211715119.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Advances in Neural Information Processing Systems*, volume 36, pages 55565–55581. Curran Associates, Inc.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 3623–3634.

Gail M Sullivan and Richard Feinn. 2012. Using effect size—or why the p value is not enough. *Journal of graduate medical education*, 4(3):279–282.

Falcon-LLM Team. 2024. The falcon 3 family of open models.

Olivier Toubia, Jonah Berger, and Jehoshua Eliashberg. 2021. How quantifying the shape of stories predicts their success. *Proceedings of the National Academy of Sciences*, 118(26):e2011695118.

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

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Kai-Ching Yeh, Jou-An Chi, Da-Chen Lian, and Shu-Kai Hsieh. 2023. Evaluating interfaced llm bias. In *Proceedings of the 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)*, pages 292–299.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, Meishan Zhang, Wenjie Li, and Min Zhang. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *Preprint*, arXiv:2407.19669.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Jianlong Zhou, Heimo Müller, Andreas Holzinger, and Fang Chen. 2024. Ethical chatgpt: Concerns, challenges, and commandments. *Electronics*, 13(17).

A Dataset: Autobiography and Biography

A.1 Selected Books

Tab. 3 lists all the books with their respective ISBN codes, retained in the autobiography-biography dataset after matching and filtering as described in Sec. 2.1.2.

A.2 Examples of Auto-Bio Paragraph Pairs Describing the Same Events

Listing 1 and Listing 2 are examples of matching autobiographical and biographical paragraphs narrating the same events from Nelson Mandela's and Luis Suarez's lives, respectively.

```
Personality: Nelson Mandela
AUTOBIOGRAPHY:
Mr. de Klerk decided to gamble. He announced that as a result
of the by-election in Potchefstroom he would call a
nationwide referendum for March 17 so that the white people
of South Africa could vote on his reform policy and on
negotiations with the ANC. He stated that if the referendum
was defeated, he would resign from office. The referendum
asked a plain and direct question of all white voters over
the age of eighteen: "Do you support the continuation of the
reform process which the state president began on 2 February
1990 which is aimed at a new constitution through
negotiation?"

BIOGRAPHY:
Rather than delay the matter any further, within days of the
Potchefstroom defeat de Klerk announced the holding of a
white referendum on the reform process on 17 March. A clear
majority in favour, he said, would obviate the need for the
government to return to the white electorate. The risks he
took were considerable. A defeat for the government would
almost certainly have led to civil war. All the resources of
the white establishment were thrown into securing a 'Yes'
vote. De Klerk explicitly promised the electorate that he
would prevent majority rule. The National Party's advertising
campaign was based on the slogan, 'Vote Yes, if you're scared
of majority rule'. The Conservative Party campaigned for a
'No' vote, claiming that the choice for Afrikaners was
between 'the survival of the Afrikaner volk and two cars in
the garage'. The ANC did its best to support the 'Yes'
campaign, reassuring whites about its good intentions in the
negotiating process and promising job security for civil
servants.
```

Listing 1: Sample paragraphs from Nelson Mandela's autobiography and biography pertaining to the same theme.

```
Personality: Luis Suarez

AUTOBIOGRAPHY:
What came next was Ghana in the quarter-final and a penalty
shoot-out to send Uruguay through to the semi-final for the
first time in forty years. After the sending off for having
'saved' a goal on the line, I watched Asamoah Gyan miss his
penalty from the entrance to the tunnel. I was in tears,
heartbroken, but when the ball flew over the bar, I ran off
celebrating. Maybe it had been worth it.

BIOGRAPHY:
It was an instinctive move and it did the job. Ghana hadn't
scored; Uruguay were still in the tournament but now down to
ten men, as Luis was sent off. Of course, Ghana still had the
chance to seal that semi-final spot from the consequent
penalty but Asamoah Gyan could only watch in despair as his
spot-kick hit the bar and bounced away to safety.
```

Listing 2: Sample paragraphs from Luis Suarez's autobiography and biography pertaining to the same theme.

B Prompts

B.1 Prompt for Topic Generation - One Shot

Listing 3 shows the prompt we used to generate topics given a story/paragraph. The prompt consists of an example paragraph and its corresponding topic (topic manually annotated, refer Listing 4 for the example paragraph and topic) to condition the LLM (*Mistral*) on the expected form of topic generation - which is commonly known as one-shot prompting. Subsequently, the prompt accepts the given story/paragraph as input and generates a topic based on it.

```
<s>[INST] A topic is the main idea discussed in a
text.
Identify the topic for the paragraph. Return only
the topic in 1-2 sentences.
Example:
Paragraph: The man who explained the difference ...
[/INST]
Topic: Bruno Silva's vital support in Groningen.
</s>
[INST] Identify the topic for the given paragraph.
Return a single topic which is most relevent for
the paragraph. Return only the topic in 1-2
sentences with no additional text or information.
```

```
Paragraph: {paragraph_text} [/INST]
```

Listing 3: Structured prompt for topic generation for stories/paragraphs using the Mistral-7B model

```
Example
Topic: Bruno Silva's vital support in Groningen.

Paragraph: The man who explained the difference between net
and gross to me, and much more besides, was Bruno Silva.
Bruno would be our salvation at Groningen in those first few
months. I remembered him as a Uruguayan international and as
a player for Danubio - one of the third teams in Uruguay
along with Defensor, behind Nacional and Penarol. We used to
get together to watch games from the Uruguayan league or we
would meet up for family barbecues. We couldn't find any
Uruguayan steak so we managed to get hold of some Brazilian
meat instead from a Brazilian who had played for many years
in Groningen called Hugo Alves Velame. He was coaching in the
academy at that point and he was someone else who was great
with Sofi and me, becoming our translator whenever we had to
deal with the club.
```

Listing 4: Example of the topic and paragraph pair used in the prompt for topic generation

B.2 Prompt for Topic Generation - Zero Shot

Listing 5 shows the prompt we used to generate topics given a story/paragraph, in a zero shot setting. Unlike the previous prompt, there is no example given and the instructions correspond to those provided to the participants of Sap et al.'s (2022) study to collect the summary. The prompt accepts the given story/paragraph as input and generates a topic based on it.

```
<s>[INST] You are given a story or event that has
happened in the given paragraph. Come up with a
short summary of the event (2-3 sentences), written
with enough details that you will remember what you
wrote about in the future. Return only the summary
in the format of "Topic: <summary>" with no
additional text.
"Paragraph: {paragraph_text}" </s>[/INST]
```

Listing 5: Zero shot structured prompt for topic generation for stories/paragraphs using the Mistral-7B model

B.3 Prompt for Story Generation

Listing 6 details the prompt template utilized for generating stories with "Good" narrative flow. The prompt for "Poor" flow replaces the "Good" for "Poor" in the instructions on generating the story, but is identical otherwise.

```
<s>[INST] Narrative flow refers to the logical
progression of a story with smooth transitions
between events. A story with good narrative flow
implies it is well structured, predictable and
transitions smoothly around the events detailed,
while a story with poor narrative flow implies it
would be poorly structured, unpredictable and with
abrupt transitions across events making it harder
to understand. Generate a single distinct short
story based on the given topic - one with good
narrative flow
Create one story, written in around 200-300 words
and 15-20 sentences on the given topic and
structure it as follows:
    Topic: {p1}
    Story (Good Flow): <insert story here>

    Do not write anything but the given template
    above before or after the story. [/INST]
```

Listing 6: Structured prompt given to the Mistral-7B for generating a story with "good" narrative flow

C Sequentiality Computation of Recalled and Imagined Stories, Given Randomly Sampled Topics: Comparison Across LLMs

We report the results of sequentiality computation from Sec. 4.2 applied to all the models in Tab. 4. The other models produce similar results as LLaMa-3.1, with much smaller effect sizes for sequentiality differences between recalled and imagined stories when using randomly sampled topics compared to human-generated topics from the recalled stories (Qwen-2.5: $d = 0.12$ vs $d = 0.55$; Falcon3: $d = 0.05$ vs $d = 0.48$; compared against the direct replication reported in Tab. 2). We again note that this difference is primarily driven by the topic term, with the contextual term having a negligible effect. Additionally, we observe that Falcon reveals negligible sequentiality differences between recalled and imagined stories when using unbiased topics even though it yields a moderate effect size for the original replication in Tab. 2.

D Sequentiality Computation of Recalled and Imagined Stories, Given Randomly Sampled Topics: Comparison Across Topic Generation Strategies

To ensure the robustness of our results, we reevaluate the sequentiality computation and comparison across the groups for randomly sampled topics using a different strategy for topic generation. We utilize the zero-shot prompting strategy detailed in Sec. 2.2. The results follow the same trends as using one-shot prompting in Sec. 4.2, with imagined stories still being more sequential than recalled stories ($t = 7.78, p < 0.001, \boldsymbol{d = 0.15}, df = 2394$) but with a much lower effect size compared to the difference obtained by using the original human-generated topics ($d = 0.15$ vs. $d = 0.57$) and comparable to one-shot prompting ($d = 0.15$ vs. $d = 0.13$). The sequentiality difference is almost entirely topic-driven ($t = -6.65, p < 0.001, d = 0.13, df = 2394$), but there is now a small difference between the context terms ($t = -2.27, p = 0.02, d = 0.04, df = 2394$).

E Generated Stories

Listing 7 shows examples of a few generated stories, demonstrating clear narrative flow differences between “good” and “poor” flow stories.

TOPIC: I had to have a thyroidectomy for thyroid cancer. I ended up fracturing my wrist on the way to surgery and then having a complication of a seroma on my neck after surgery.

Good Flow: I had to have a thyroidectomy, a routine surgery to remove my thyroid due to cancer. The day of the procedure arrived, and I felt a strange sense of calm. I had prepared myself mentally for this moment. The sun was just starting to rise as I stepped out of my house, ready to embark on this new journey. But little did I know, my day was about to take an unexpected turn. As I walked to the car, I tripped over a loose stone and fell hard, fracturing my right wrist. The pain was sharp and intense, but I pushed it aside, knowing I couldn't let my fear of surgery be outweighed by a simple injury. At the hospital, I was reassured by the medical staff. They assured me that my wrist would heal, and the surgery would go on as planned. I was reassured, but deep down, I couldn't shake off the feeling of unease. The surgery went smoothly, and I was relieved to wake up in recovery, feeling groggy but otherwise fine. But then, a few days later, I noticed a strange lump forming on my neck. It was a seroma, a pocket of fluid that had collected under the incision site. I was prescribed antibiotics and given instructions to keep the area clean. But as the days passed, the seroma grew larger, and I began to feel a sense of panic. I called my doctor, who scheduled an emergency appointment. At the hospital, they drained the seroma, and I was given more antibiotics. This complication added to my recovery time, but I remained determined. I knew that, with time and patience, I would get through this. Despite the setbacks, I remained hopeful. I knew that the road to recovery would be long, but I was ready to face it head-on. With each day that passed, I grew stronger, both physically and mentally. And though the memories of that fateful day still lingered, I knew that I would emerge from this experience stronger and more resilient than ever before.

Poor Flow: I had to have a thyroidectomy for thyroid cancer. The day of the surgery, I woke up early and prepared myself. I packed my bag, ate a light breakfast, and headed out the door. As I was walking to the hospital, I tripped over a crack in the pavement and fractured my wrist. I was shocked and in pain, but I knew I couldn't let this stop me from getting the surgery I needed. I made it to the hospital and checked in. The surgery went well, but a few days later, I developed a serious complication. A seroma formed on my neck, a pocket of fluid that caused discomfort and made it difficult to move. I went back to the hospital, and they drained the seroma several times. But it kept coming back. The doctors were baffled, and I was frustrated. I couldn't understand why this was happening. The fractured wrist was a setback, but I had expected to recover smoothly from the surgery. But now, with this new complication, I was unsure of what to expect. The days turned into weeks, and the seroma refused to go away. I went through numerous treatments and procedures, but nothing seemed to work. I was losing hope, feeling like I was stuck in a never-ending cycle of pain and discomfort. Just when I thought things couldn't get any worse, I received some unexpected news. The pathology report from the thyroidectomy showed that the cancer had spread. My world came crashing down around me. I was devastated, feeling like I had been dealt a terrible hand. The fractured wrist and seroma seemed insignificant compared to this new revelation. I was left with so many questions and so few answers. The journey ahead was uncertain, and I was unsure if I had the strength to face it.

TOPIC: I went to the fair with some of my friends and my partner and we had a picture-perfect day. Everything from the food, to the weather, to the crazy fair art was amazing.

Good Flow: I went to the fair with my partners hand in mine and my friends by my side, the sun casting a golden glow over the vibrant scene before us. The air was filled with the sweet scent of funnel cakes and popcorn, and the sounds of laughter and excitement echoed through the park. We began our day with a leisurely stroll along the midway, taking in the sights and sounds. My friends and I marveled at the carnival games, our partners teasing us as we tried our hand at ring toss and balloon darts. The sun continued to shine, the weather perfect, as we made our way to the food stands. We shared a picnic blanket beneath the shade of a large tree, enjoying the flavors of corn dogs, cotton candy, and funnel cakes. My friends and I reminisced about our childhood memories of fair visits, while my partner shared stories of their own. The day went on in this blissful manner, with each event seamlessly leading to the next. We explored the fair art exhibit, where our jaws dropped in awe of the creative masterpieces. The sun began to set, and as we watched the sky turn pink and orange, we knew it was time to leave. As we gathered our things and said our goodbyes, we couldn't help but feel grateful for the picture-perfect day we had shared together. The memories would stay with us long after the fair, a testament to the beauty of friendship and love.

Poor Flow: I went to the fair with some of my friends and my partner. We had an amazing roller coaster ride to start the day. Suddenly, we were at the cotton candy stand. My partner wanted to buy some, but we had lost track of my friends. I went to look for them, but they were nowhere to be found. Suddenly, it started raining. We took shelter under a tree, but my partner was insistent that we should go back home. I disagreed, and we decided to continue our fair day. But as we were walking, we came across a petting zoo. My partner wanted to see the animals, but we ran out of tickets. Suddenly, we heard a loud announcement that the Ferris wheel was malfunctioning. We decided to check it out and found a long line, but my partner was impatient and insisted on cutting in line. The crowd protested, but we managed to get on. As we were enjoying the view, suddenly, the rain stopped, and the sun came out. We left the Ferris wheel and went to the art stand. My partner wanted to buy a painting, but we couldn't decide which one to choose. Suddenly, my friends appeared out of nowhere and helped us make a decision. We all had fun at the carnival games, but then, my partner got sick and we had to leave. The day ended abruptly, and we didn't even get to try the funnel cakes. The fair day was full of unexpected twists and turns, and the day ended as suddenly as it had begun.

Listing 7: Two example story pairs, consisting of "good" and "bad" flow stories generated by Mistral-7B.

Name	**Autobiography**	**Biography**
Isaac Asimov	9780553569971	9780810831292
Diane von Furstenberg	9781451651546	9780062041234
Lucille Ball	9781101667088	9781504018920
Muhammad Ali	9781631680496	9780791081563
Anne Frank	9780553577129	9781408842119
Ansel Adams	9780316437011	9781620405550
Bruce Springsteen	9781501141515	9781101606247
Andrew Carnegie	9789354203503	9781101201794
Fidel Castro	9781416562504	9780745630069
Julia Child	9780307277695	9780307762856
Winston Churchill	9781587315367	9780805023961
Jacques Cousteau	9780792267966	9780307378279
Alex Ferguson	9780340919408	9780224083072
Richard Feynman	9780393355628	9781453210468
Benjamin Franklin	9781508475095	9780684807614
John Kenneth Galbraith	9780345303233	9781466893757
Mahatma Gandhi	9780486245935	9780307269584
Billy Graham	9780061171062	9780849917028
Che Guevara	9781644210963	9780802197252
Buster Keaton	9780306801785	9781497602311
Henry Kissinger	9781451636468	9780698195691
Langston Hughes	9781466883499	9780195146431
Niki Lauda	9781473577954	9781471192036
Malcolm X	9780345350688	9781101445273
Nelson Mandela	9780316548182	9781586489519
Michelle Obama	9781524763138	9780307958822
Paul Robeson	9780807096932	9781497635364
Theodore Roosevelt	9781438295343	9780307777829
Elizabeth Cady Stanton	9781505923551	9780195037296
Luis Suarez	9781472224255	9781784181949
Sachin Tendulkar	9781473605190	9788174363602
Nikola Tesla	9781684222063	9781585093083
Margaret Thatcher	9780062049452	9780713992823
Mark Twain	9780520267190	9780307874597
Mike Tyson	9780007502516	9781476618029
Edith Wharton	9780684847559	9780307555854
Virginia Woolf	9781448181889	9781407066240
Paramhansa Yogananda	9781565892125	9780190668051

Table 3: List of personalities and the ISBN codes of their autobiographies and biographies used in the analysis.

Model	Metric	t-statistic	p-val	Cohen's d	NLL_I	NLL_R
LLaMa-3.1-7b	Topic-driven$(-)$	-4.56	10^{-6}	0.09	-3.58	-3.52
	Context-driven$(+)$	-0.23	0.81	0.004	-2.54	-2.54
	Sequentiality$(+)$	6.62	10^{-11}	0.12	1.03	0.98
Falcon3-10b	Topic-driven$(-)$	-2.30	0.021	0.04	-3.98	-3.94
	Context-driven$(+)$	-0.008	0.24	0.01	-2.75	-2.74
	Sequentiality$(+)$	2.65	0.008	0.05	1.23	1.20
Qwen-2.5-7b	Topic-driven$(-)$	-4.51	10^{-6}	0.09	-3.82	-3.75
	Context-driven$(+)$	-0.35	0.72	0.007	-2.62	-2.61
	Sequentiality$(+)$	6.29	10^{-10}	0.12	1.20	1.14

Table 4: Sequentiality comparison of imagined and recalled stories using different LLMs, given LLM-generated topics sampled randomly from imagined and recalled stories. NLL_I & NLL_R are the mean values for negative log likelihood of imagined and recalled stories, respectively. $(+)$ and $(-)$ indicate the expected direction of difference for the metric to replicate Sap et al.'s (2022) result. For example, Sequentiality $(+)$ indicates that a positive value in that row replicates the finding that imagined stories flow more sequentially than recalled ones. Additionally, our requirement that the contextual term drives the effect is also indicated by a $(+)$. Extremely small p-values are approximated to the closest power of 10.

Components of Creativity: Language Model-based Predictors for Clustering and Switching in Verbal Fluency

Sina Zarrieß, Simeon Junker, Judith Sieker, Özge Alaçam
Computational Linguistics, Department of Linguistics
Bielefeld University, Germany
`{sina.zarriess;j.sieker;simeon.junker;oezge.alacam}@uni-bielefeld.de`

Abstract

Verbal fluency is an experimental paradigm used to examine human knowledge retrieval, cognitive performance and creative abilities. This work investigates the psychometric capacities of LMs in this task. We focus on *switching* and *clustering* patterns and seek evidence to substantiate them as two distinct and separable components of lexical retrieval processes in LMs. We prompt different transformer-based LMs with verbal fluency items and ask whether metrics derived from the language models' prediction probabilities or internal attention distributions offer reliable predictors of switching/clustering behaviors in verbal fluency. We find that token probabilities, but especially attention-based metrics have strong statistical power when separating between cases of switching and clustering, in line with prior research on human cognition.

1 Introduction

The processes underlying human creative abilities have been an important topic of research in several fields. Research in cognitive science suggests that semantic association and search are core aspects of creative thinking (Mednick, 1962; Gilhooly et al., 2007; Beaty and Silvia, 2012). Therefore, creative abilities in humans are commonly tested and measured using semantic search tasks such as verbal fluency, in which participants are asked to list lexical items for a given category in a short period of time (e.g., name as many animals as possible in 60 seconds) (Beaty et al., 2014a).

Human responses to such tasks exhibit a well-known search pattern, which has been termed "clustering and switching" or "exploitation and exploration" (Troyer et al., 1997). During clustering, humans generate sequences of words that belong to the same subcategory, exploiting the neighbourhood of previous items in the semantic space. As this subcategory becomes increasingly exhausted, they switch to other subcategories, shifting their attention to a different patch in their conceptual space (see Figure 1). Recent work suggests that clustering and switching are two fundamental components of semantic search related to creative abilities and has aimed to identify neurocognitive correlates of these processes (Ovando-Tellez et al., 2022).

In this paper, we investigate whether transformer language models (LMs) provide further evidence for the hypothesis that creative semantic search in verbal fluency involves two distinct, separable processes related to clustering and switching. The design of our experiments follows Ovando-Tellez et al. (2022), who tested correlations between the occurrence of clusters and switches in participants' responses to fluency tasks and metrics for participants' creativity, semantic network structure, and brain connectivity. In our study, we replace these metrics of human neuro-cognitive processes with a set of probability and attention-based measures computed with language models over human verbal fluency sequences. We test whether these measures provide predictors of clusters and switches in the human sequences, e.g., whether attention is distributed differently in the LM when retrieving a word within a cluster as compared to a switch.

Our motivation for studying clustering and switching in verbal fluency using LMs is twofold: First, we note that cognitive science has a longstanding interest in computational models that capture human behavior in verbal fluency and other creative search tasks. Existing models in this area typically implement graph-based semantic networks and explicit search algorithms on top of these networks (Hills et al., 2012; Zemla and Austerweil, 2017). We believe that LMs are an obvious alternative modeling approach worth exploring here since their implicit semantic representations and word prediction processes have been shown to excel in a variety of generative tasks. LM-based correlates of clustering and switching would demonstrate the

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 216–232
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

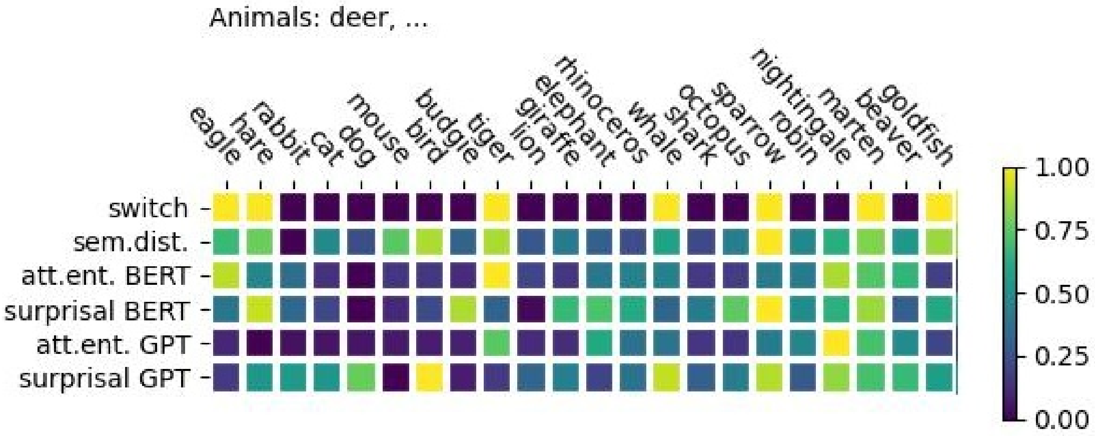


Figure 1: Translated verbal fluency response from BIEFU (Alacam et al., 2022) with annotations of clusters and switches (first row); semantic distances (cosine distances of ConceptNet embeddings) between consecutive items; LM predictors: attention entropy and surprisal predictors from BERT and GPT respectively (all scores are min-maxed normalized for visualization). "Animals: deer" is the LM prompt used to (re)-generate the sequence.

potential of LMs to complement the landscape of computational approaches in this field and, in particular, to provide an account of general language and word sequence processing mechanisms in verbal fluency that are hard to come by in small-scale net-work-based models (cf. Heineman et al., 2024). At the same time, research on LMs is increasingly interested in testing their elementary language processing abilities. Recent studies have tested the extent to which surprisal or attention-based scores computed with LM predict human reading times, providing a cognitively plausible account of processing difficulties in reading and language comprehension (see Oh and Schuler, 2022; Shain et al., 2024). The verbal fluency paradigm complements the landscape of existing probing tasks and analysis methods toward production-oriented tasks involving semantic search and creative abilities. In this study, we ask whether LM-based metrics separate between clustering and switching, as two central components of creative semantic search. Our results suggest that LMs provide novel and strong predictors for modeling human behavior in the verbal fluency task and that attention distribution in LMs has predictive power in accounting for clustering and switching.

2 Background

2.1 Verbal fluency

The verbal fluency task is a neuropsychological test of verbal functioning that is commonly used to measure cognitive performance in e.g. lexical knowledge and retrieval or executive control (Shao et al., 2014). We focus on categorical fluency, which involves repeated retrieval of lexical items for the same category. This gets more challenging when easily accessible words are exhausted and participants are required to transition from fast, associative processes to a more controlled semantic search (Demetriou and Holtzer, 2017). Verbal fluency data is often analyzed in terms of *clusters* and *switches* structuring the word sequence, i.e., word spans that fall into the same semantic subcategories or transitions between subcategories (Troyer et al., 1997; Kim et al., 2019). In Figure 1, for example, the sequence *rabbit, cat, dog, ...* corresponds to a cluster followed by switch from *budgie* to *tiger*. Words within a cluster are typically produced in a fast, associative manner. Switches, in turn, show longer retrieval times as they involve effortful search, executive control (i.e. inhibition of common or previous items), and efficient navigation of long-term semantic memory (Michalko et al., 2023).

The interaction of clustering and switching that typically appears in human verbal fluency responses plays an important role in creativity research (Silvia et al. 2013; Beaty et al. 2014b; Beaty and Kenett 2023, among others). Ovando-Tellez et al. (2022) show that clustering is related to *divergent thinking*, i.e., generating new and effective ideas, while switching is connected with *convergent thinking* or combining available information in creative ways, and both are characterized by distinct brain connectivity patterns. They argue that clustering involves associative abilities, while switching requires controlled memory retrieval processes, executive functions and memory.

2.2 Computational Models of Verbal Fluency

The computational modeling of verbal fluency data has received considerable attention in cognitive science research. Existing models typically implement the generation of verbal fluency responses as a search over a semantic network or graph (Hills et al., 2012; Abbott et al., 2015; Zemla and Austerweil, 2017; Avery and Jones, 2018), where clustering and switching emerges from the search strategy as in the foraging model by (Hills et al., 2012) or from the underlying structure of the network as in the model by (Abbott et al., 2015). To a similar end, other approaches make use of biologically inspired neural networks (Kajić et al., 2017) or, more recently, pre-trained transformer models (Nighojkar et al., 2022) and LLMs (Heineman et al., 2024; Wang et al., 2025). In general, these models are tested for their ability to predict or simulate human fluency sequences on a word level.

Other computational work on verbal fluency focused explicitly on automatically annotating clustering-switching patterns in sequences produced by humans. Some studies have explored the use of distributional semantic representations and word embeddings for scoring semantic fluency data (Linz et al., 2017; Paula et al., 2018; Kim et al., 2019; Alacam et al., 2022) or the ability of pre-trained LMs in predicting category switches (Heineman et al., 2024).

In contrast to these models, our study does not aim to explicitly reproduce or simulate the semantic search strategies observed in human verbal fluency responses with LMs. Instead, we focus on investigating their underlying word retrieval and prediction processes. Inspired by Ovando-Tellez et al. (2022), we ask whether we can identify distinct components of verbal fluency, i.e. clustering and switching, from processing-related behavioural measures computed with an LM.

2.3 Linguistic and Cognitive Probing of LMs

The analysis of linguistic and cognitive capabilities captured in LMs has become an important area of research (Belinkov and Glass, 2019; Baroni, 2022; Chang and Bergen, 2023; Binz and Schulz, 2023; Strachan et al., 2024). A common paradigm in LM probing is behavioral analysis, which treats the pretrained LM as a black box and uses carefully controlled test suites or experimental datasets from (psycho-)linguistics to compare model outputs against human productions or judgments. This paradigm is useful for testing whether LMs learn particular linguistic rules and generalizations, in particular in the domain of syntax (Warstadt et al., 2020), but provide very limited insights into how underlying processing mechanisms in LMs align to human language processing and cognition (cf. Baroni, 2022; Chang and Bergen, 2023).

Other work on probing LMs focuses on their ability to account for effects of processing difficulty, and mostly goes back to the idea of "surprisal" (Hale, 2001; Levy, 2008; Demberg and Keller, 2008; Smith and Levy, 2013). Surprisal is defined as the negative log probability of a word in context and has been demonstrated to provide a very robust predictor for human processing times (e.g., to reading times) when computed with language models of different sizes or perplexities (Goodkind and Bicknell, 2018; Shain et al., 2024). These findings lend support to expectation-based accounts of sentence processing in psycholinguistics, aligning word prediction processes in LMs with humans' anticipation of upcoming material in sentence reading. A few recent studies explored further predictors complementing surprisal. Thus, the attention mechanism of transformer LMs has been considered to approximate aspects of memory and attention in human cognition (Ryu and Lewis, 2021; De Varda and Marelli, 2024). Most importantly for our study, Oh and Schuler (2022) showed that attention distribution and distance metrics from internal layers of the LM yield very powerful predictors for self-paced reading times and gaze durations in naturalistic reading, drawing connections to memory-based accounts of sentence processing. As memory is an important aspect of semantic search in the verbal fluency task (Ovando-Tellez et al., 2022), our study will examine both surprisal (or, more generally, probability-based) predictors computed at the LM's output layer as well as attention-based predictors from the internal layers.

However, although LMs are now frequently used as computational testbeds for theories of language processing and cognition, the field is still debating which of the many existing LMs can provide the most robust and cognitively plausible predictors of human processing. Oh et al. (2022) tested surprisal estimates from GPT-2 models of different sizes and showed that the surprisal computed with smaller model sizes achieved a better fit with human reading times than larger model sizes. Similar observations have been made in (Kuribayashi et al., 2022; Oh and Schuler, 2023). Wilcox et al. (2023), on the other hand, trains LMs of small and medium size on a range of languages and finds that LM quality generally correlates with its psychometric predictive power. Therefore, in the following, we will rely on some less recent but widely used LMs such as BERT or GPT-2, but also include variants of more recent models available in different sizes.

3 Motivation and research questions

The main question of this work is whether current transformer LMs can account for effects of processing difficulty in a creative word retrieval task – verbal fluency – where clear differences in retrieval difficulty have been widely observed in terms of clustering-switching patterns (Troyer et al., 1997; Hills et al., 2012). In the following, we will detail the assumptions underlying this question.

Why could prediction and attention mechanisms implemented in LMs explain effects of processing difficulty in the verbal fluency task? At a basic level, verbal fluency involves repeated retrieval of lexical items, which aligns well with the autoregressive, left-to-right word prediction objective implemented in modern LMs. Research on verbal fluency in psychology and linguistics typically emphasizes that the verbal fluency task involves a whole range of different cognitive and verbal abilities, such as access to the mental lexicon, semantic knowledge, search strategies, language processing, executive control functions, long-term memory, and attention (Kim et al., 2019; Michalko et al., 2023; Ovando-Tellez et al., 2022). Importantly, many previous studies have found strong evidence for lexical access and language production processes being critical components in verbal fluency (Weckerly et al., 2001; Whiteside et al., 2016; Marko et al., 2023). Therefore, we believe that LMs with their complex underlying architecture for representing and modeling word sequences may offer additional benefits over traditional, relatively small-scale network models (Hills et al., 2012; Abbott et al., 2015) building on Markovian assumptions and being detached from general language processing accounts (Heineman et al., 2024).

Which LM-based predictors can be expected to account for processing effects of clustering and switching in verbal fluency? While previous modeling approaches typically rely on some form of semantic distance to account for clustering-switching patterns, this work proposes to use word prediction and attention-based measures computed from LMs as proxies of retrieval difficulty in verbal fluency. We expect these predictors to inherently account for sequence processing effects and to capture retrieval difficulties beyond semantic-taxonomic distances. As a motivating example, consider the first cluster of the sequence in Figure 1 corresponding to common "pets" (*rabbit, dog, cat, ...*). Here, attention entropy and surprisal scores computed with BERT predict that these words are easy to retrieve, matching the annotation as a cluster. However, the semantic distance predicts greater difficulty, potentially due to taxonomic distances between, e.g. *mouse* and *bird*. In simple terms, we assume that words corresponding to switches and higher retrieval difficulty in humans are modeled as less predictable and requiring higher attention-entropy in LMs. For instance, the word *dog* following *cat* should have a relatively low surprisal compared to the word *tiger* following *budgie*, as illustrated in Figure 1. Higher attention entropies, in turn, indicate that the model distributed attention weights more evenly across the preceding sequence which in Figure 1 is often the case for words corresponding to switches (*tiger, whale, sparrow, ...*).

4 Experimental Method

4.1 Data

We base our experiments on BIEFU (Alacam et al., 2022), a dataset of German verbal fluency responses, which covers a fairly high number of categories. The BIEFU data was collected from 100 participants and contains verbal fluency responses that enumerate words for 10 different semantic categories (e.g., animals, hobbies, body parts). An overview of the data is shown in Table 4 (App. A).

Soft and Hard Switches The BIEFU dataset includes manual annotations of lexical items with semantic subcategories. Based on these, we determine soft (fluid) and hard (static) switches, following Zemla and Austerweil (2019). A soft cluster switch occurs when the next word in a list does not share a sub-category label with the previous word, while a hard switch occurs whenever the next word does not share a sub-category label with any of the previous words since the start of the last cluster. Soft switches are the most commonly examined types of switches in the literature and we will focus on these in the following.

4.2 Prompting

To obtain prompts from human verbal fluency sequences, Nighojkar et al. (2022) replaced the last item in a partial verbal fluency sequence with a mask token, cf. (1).

(1) *[C]*s I know are w_{n-1-ct}, …, w_{n-1}, and the [MASK] .

Here, w_{n-1-ct} (ct being the context size) is the initial and w_{n-1} the penultimate item in a sequence produced for category *C*. [MASK] always represents the last item. We adopt this scheme and iteratively mask out subsequent items in each human-produced sequence, i.e., shift the masked token from left to right by truncating them at the position of the masking token, cf. the prompts in Table 1. Baseline prompt (pr-0), which consists of a simple enumeration preceded by the category name, is added for comparison. Since LMs can be very sensitive to the specification of their prompts, we conducted further experiments with prompt design that addresses both auto-regressive and bidirectional prompt strategies with different wording variations, see Table 5 (App. B.2) for additional results on these.

Seq:	dog, cat, mouse, ...
pr-0_1	Animals: dog, [MASK]
pr-0_2	Animals: dog, cat, [MASK]
pr-1_1	Animals I know are dog, [MASK]
pr-1_2	Animals I know are dog, cat, [MASK]

Table 1: A (translated) sample of a human response and derived LM prompts for two subsequent steps in a verbal fluency sequence for autoregressive prompting.

4.3 Language Models

Since our investigation is one of the first to test the predictive power of LMs in distinguishing clustering and switching, we select basic transformer LMs that have also been widely used in the literature on cognitive probing – GPT-2 (Radford et al., 2019), BERT (Devlin et al., 2019) and T5 (Raffel et al., 2020). Next to these, we also include open-source German or multilingual models that come in different size – Bloom (350m, 1b5, 1b7) (Scao et al., 2023) and XGLM (560M, 1b7) models (Lin et al., 2021). This model selection ensures a representative comparison across transformer architectures that employ different versions of the self-attention mechanism: BERT as a bidirectional encoder model, GPT-2, Bloom and XGLM as unidirectional autoregressive decoder models, and T5 as an encoder-decoder transformer.

4.4 Predictors of Switching and Clustering

We use generalized linear mixed-effect models to test the predictive power of probability-based and attention-based metrics derived from LMs to separate clustering and switching in verbal fluency data. In the following, we describe the predictors we include in this statistical analysis.

4.4.1 Psycholinguistic Predictors

We implement a strong baseline model that predicts clustering/switching based on fixed and random effects established in recent verbal fluency literature (Michalko et al., 2023). These predictors are *temporal order*, *task demand*, *Typicality*, *Inter-response similarity*. We add the participants and semantic categories as a crossed random effect to the initial model ($m0$).

Temporal order (TEMP). The normalized temporal order (TEMP) corresponds to the current position of the word in a sequence divided by the length of that sequence (range between 0 and 1). This predictor captures the fact that words are more difficult to produce the longer the sequences become.

Task demand (TD). This predictor reflects that certain verbal fluency categories are systematically easier to enumerate than others, due to their familiarity, frequency, and lexical specificity. For instance, categories like *animals* and *vegetables* are easier to enumerate since they are more frequent, while other categories like *fabrics* or *insects* are less easily accessible. Following Michalko et al. (2023), we manually group the verbal fluency categories into three so-called "task demand categories".

Typicality (TYP). Next, we add a fixed effect that captures the typicality of an item within a verbal fluency category (TYP). TYP is calculated as the logarithm of the absolute number of occurrences of a word among all items enumerated by all participants within that particular category. See App. A for further detail.

Inter-response similarity (IRS) We compute the semantic similarity of subsequent lexical items in a verbal fluency sequence. Here, we deviate slightly from Michalko et al. (2023) and use the cosine similarity between the items' word embeddings, computed with the ConceptNET Numberbatch word embedding. This semantic space is enriched with ConceptNet taxonomic relations (Speer et al., 2017), achieves the best performance in predicting clustering and switching patterns in BIEFU data Alacam et al. (2022).

Retrieval latency (RL) Our data records time stamps of every typed character in the verbal fluency sequence. We define retrieval latency as the time span as the offset between a preceding item and the onset of the next item. We calculate it by subtracting the offset of the first item from the onset of the second item.

4.4.2 Probability-based Predictors

Our first set of LM predictors is derived from word probabilities. We regard these as measures of retrieval difficulty or predictability in sequence generation, mirroring the notion of "expectation" in sequence understanding (Shain et al., 2024). We expect that clustering corresponds to less surprising items, whereas switching should show higher surprisal and lower probabilities. The handling of words composed of subwords across different LM architectures is detailed Appendix B.1. To test this hypothesis, we consider the following predictors:

Surprisal (Surp.). We transform word probabilities into surprisal scores, quantifying the information content it conveys in the context in which it appears. The surprisal of a word w is calculated as the negative log-likelihood of its probability. We expect a positive correlation with switching.

$$\text{Surprisal}(w_i) = -\log_2 p(w_i \mid w_{<i})$$

Rankings (Rank). This predictor derives determines the rank of the word w in the word probability distribution. We expect a positive correlation with switching. The rank parameter is highly dependent on the vocabulary size of the LM architecture. In our analysis, the rank scores are normalized, but see Appendix B.4 for more information.

$$\text{Rank}(w) = \text{arg min}_i \; \{p(w \mid \text{context}) : i = 1, 2, \ldots, N\}$$

Entropy (Ent.). As another account of retrieval difficulty in context, we include the entropy of the word probability distribution, quantifying the model's uncertainty in the given context, regardless of the probability or rank of the target item. We expect a positive correlation with switching.

$$\text{Entropy}(w_i) = -\sum_{w_i} p(w_i \mid w_{<i}) \log_2 p(w_i \mid w_{<i})$$

4.4.3 Attention-based Predictors

The second set of LM predictors derives from the model's internal attention distributions as measures of cognitive effort, related to monitoring and shifting working memory and attention (Ryu and Lewis, 2021; De Varda and Marelli, 2024). We expect that switching corresponds to higher cognitive effort, e.g., wider attention distributions across layers and heads, than clustering which we expect to show more localized attention patterns.

We extract the attention-based predictors considering different layers and attention heads in the transformer architecture (144 heads in total for the smaller LMs, 256 for the larger LMs). We first transform the embeddings of tokens or hidden states of a sequence to a triple of query (q), key (k), and value (v) embeddings. The heads then compute the attention weight between the query and key vectors for all pairs of tokens in the input prompt as soft-max-normalized dot products.

$$\alpha_{ij} = \frac{exp(q_i^T k_j)}{\sum_{l=1}^{n} exp(q_i^T k_l)}$$

The diffuseness of attention obtained from these attention maps α can be calculated in different ways. We follow Clark et al. (2019) and consider attention head entropy and distance between attention distribution for subsequent items in the sequence.

Average Attention-Heads Entropy (AHE). The attention entropy is calculated in a similar way to the probability-based entropy metric. The key distinction lies in its application to attention weight distributions instead of a softmax-adjusted probability distribution. Subsequently, the attention entropy is obtained by averaging across all heads for the respective iteration of the input prompt. High entropy is associated with bag-of-words context incorporation (Clark et al., 2019).

$$Entropy(\text{head}) = -\sum_{i=1}^{N} \alpha(i) \log_2 \alpha(i)$$

Here, $\alpha(i)$ represents the probability associated with the i-th element in the attention distribution.

Average JS-Divergence in attention heads (AH-JSD). To explore whether attention heads in the same layer can be grouped based on similar behavior, we compute the distances between all pairs of attention heads. This pairwise distance between the attention distribution of each pair of heads H_i and H_j is calculated using Jenson-Shannon Divergence following (Clark et al., 2019). Lower divergence indicates that all heads process the inputs in a similar way.

$$JSD = \sum_{token \in Prompt} JS(H_i(token), H_j(token))$$

5 Experiments

We now describe our experiments, testing the predictive power of LM predictors in distinguishing between clustering and switching in a creative semantic search task. All analyses were carried out in R version 2024.12.x (R Core Team, 2021). The models are compared using ANOVA and all numerical values are (z-)normalized using the *scale* function in R.

5.1 Baseline Models

We use mixed-effect logistic regression (glmer) and fit them on annotations of switching and clustering in human verbal fluency responses. The dependent variable is coded as a binomial variable (0: cluster, 1: switch), indicating clustering or switching between consecutive words in a sequence.

For the baseline model, we applied forward stepwise inclusion starting with $m0$ which has only crossed-random effects of participant and category. The order of the inclusion of the parameters in the baseline is from more basic (temporal order) to complex (retrieval latency). For model $m1$ to $m5$, we add the baseline predictors from Section 4.4.1 as follows:

$$\begin{aligned} m0 &: switch + (\sim 1|part.) + (\sim 1|cat.) \\ m1 &: m0 + \text{TEMP} \\ m2 &: m1 + \text{TASKDEMAND} \\ m3 &: m2 + \text{TYP} \\ m4 &: m3 + \text{IRS} \\ m5 &: m4 + \text{RL} \end{aligned}$$

The temporal order parameter did not improve the model fit ($\mathcal{X}^2(1) = 1.31, p > .05$). Adding task demand (TEMP) has a significant effect ($\mathcal{X}^2(2) = 6.64, p < .05$). The main effects of the typicality (TYP) and of the inter-response similarity parameter (IRS) were also found significant (($\mathcal{X}^2(1) = 44.63, p < .0001$) and ($\mathcal{X}^2(1) = 3384, p < .001$), respectively). For the hard switch, all parameters significantly contributed to model fit (see Appendix B.3 for the details). The results indicate that $m5$ is the strongest baseline for switch modeling.

This set of baseline models, commonly used in the verbal fluency literature, enables us to quantify and compare the individual contributions of a rich array of LM predictors that we propose.

5.2 Models with LM predictors

Next, we analyze the power of LM predictors in modeling clustering and switching. The following model list shows in which order the probability and attention-based variables from Sections 4.4.2 and 4.4.3 are included:

$$\begin{aligned} lm\_m6 &: (m3,\ m4, or\ m5) + Prob_{LMtype} \\ lm\_m7 &: (m3,\ m4, or\ m5) + Rank_{LMtype} \\ lm\_m8 &: (m3,\ m4, or\ m5) + Ent_{LMtype} \\ lm\_m9 &: (m3,\ m4, or\ m5) + AHE_{LMtype} \\ lm\_m10 &: (m3,\ m4, or\ m5) + AH - JSD_{LMtype} \end{aligned}$$

Thus, adding LM predictors to $m3$ shows the contribution of probability and attention-based predictors to a model that includes the baseline predictors of temporal order, task demand, and typicality. Then, we test the predictive power of LM parameters to the $m4$ model, which includes a significant predictor for semantic similarity between consecutive words (IRS). Finally, we add them to the $m5$ model, which further includes retrieval latency (RL), a highly predictive variable for clustering and switching.

5.3 Results

Table 2 summarizes the contribution of each LM predictor for soft switch modeling when added to the defacto baseline model ($m3$). The results for $m3$ in Table 2 show clear evidence for the predictive power of LM predictors, in separating between clustering and switching processes. The attention-based metric AH-JSD, in particular, models these processes very robustly and independently from the underlying LM, i.e. it is highly significant for all LMs. This also holds for the AHE metric, which achieves slightly lower values across the board, though. The probability-based metrics are less consistent across LMs: T5, Bloom350, and XGLM yield a highly significant RANK variable while surprisal is less significant. However, SURPRISAL derived from BERT achieves substantial predictive power, comparably to AHE. Most probability-based predictors from GPT-2 are insignificant.

Analysis with Concept Similarities and Retrieval Latency. We further investigate the re-

Table 2: Soft Switch: the individual contributions of LM-predictors to the base model ($m3$) (Chi-Square). *** denotes significance (p) < 0.001. ** : p < 0.01 and * : p < 0.05

		BERT	T5	GPT-2	Bloom350	Bloom1b5	Bloom1b7	XGLM560	XGLM1b7
$m3$	**Prob**	37.44***	11.28***	2.20	0.65	0.62	2.64	4.68*	15.26***
	Rank	9.64**	51.25***	1.49	50.41***	0.74	2.41	67.79***	76.78***
	Surprisal	64.08***	12.89***	3.86*	46.99***	23.09***	2.78	30.25***	17.25***
	Entropy	2.91	0.83	3.54	33.02***	0.72	1.03	63.16***	3.21
	AHE	60.43 ***	33.66***	45.02***	32.31***	32.31***	31.97***	52.68***	52.68***
	AH-JSD	**106.26** ***	**63.64*****	**92.35** ***	**71.07*****	**68.34*****	**73.56*****	**85.11*****	**79.52*****

Table 3: Soft Switch: the individual contributions of LM-predictors on top of $m4$ and $m5$ models (Chi-Square). *** denotes p < 0.001. ** : p < 0.01 and * : p < 0.05

		BERT	T5	GPT-2	Bloom350	Bloom1b5	Bloom1b7	XGLM560	XGLM1b7
$m4$	**Prob**	1.56	22.85 ***	29.05 ***	**56.94** ***	50.15	0.005	9.77**	15.11***
	Rank	4.26 *	16.50 ***	8.35 **	35.96 ***	48.61***	8.92**	14.92***	29.39***
	Surprisal	10.76 **	0.89	1.34	53.55 ***	**74.28*****	1.22	0.02	4.19*
	Entropy	0.15	0.97	1.96	42.27***	0.79	0.01	**71.10*****	7.19**
	AHE	46.65 ***	21.03 ***	31.24 ***	20.27 ***	20.27***	15.79***	34.95***	34.95***
	AH-JSD	**71.41** ***	**29.28** ***	**58.64** ***	38.61 ***	35.16***	**34.88*****	43.88***	**39.10*****
$m5$	**Prob**	1.85	**24.05** ***	**30.05** ***	**56.13** ***	50.69***	0.001	8.81**	16.83***
	Rank	4.54 *	17.57 ***	6.80 **	33.12***	43.35***	**7.13****	15.82***	**28.39*****
	Surprisal	8.95 **	2.06	0.93	51.61 ***	**74.14*****	1.55	0.20	4.68*
	Entropy	0.49	1.59	2.05	35.39 ***	1.32	0.02	**69.39*****	6.12*
	AHE	14.99 ***	3.51 *	7.44 **	2.67	2.67	1.11	8.73**	8.73**
	AH-JSD	**24.93** ***	4.02 *	17.93 ***	7.28 **	5.71*	5.13*	8.64**	6.35*

lationship between LM parameters and semantic similarity (IRS) – one of the most frequently used NLP metrics in verbal fluency modeling – as well as retrieval latency (RL) as a strong behavioural measure of processing difficulty. Table 3 summarizes the contribution of each LM predictor for soft switch modeling when added to the $m4$, and $m5$ models, respectively. Looking at the results for $m4$, we find that a number of LM predictors remain highly significant, even on top of the strong similarity variable IRS. This holds in particular for the attention-based metrics, most notably for AH-JSD. This confirms our hypothesis that attention distributions in the internal layers of LMs capture aspects of processes in semantic search beyond static similarities in embedding space. However, we also see notable differences in how predictors from different LMs interact with IRS. Bloom350 and Bloom1b5's attention-based metrics seem to be more closely aligned with the IRS parameter (resulting in lower contributions) compared to their probability-based parameters. The probability-based predictors of BERT, however, are not significant anymore when combined with IRS.

The results for $m5$ closely align with those of $m4$, with the primary difference being a substantial decrease in the magnitude of contribution for attention-based models. As $m5$ includes the highly significant retrieval latency parameter from the human data, we take this as a promising finding suggesting that attention-based metrics derived from LMs show some alignment with humans internal retrieval processes. The inclusion of retrieval latency does not influence the contribution of probability-based metrics which supports the view that they capture complementary aspects of clustering and switching in our data.

LM Comparison. When comparing all three testing conditions, attention-based metrics are the most robust predictors across different LM architectures. Their predictive power only decreases when added after the retrieval latency parameter, which suggests that attention-based predictors are highly aligned with retrieval latency in humans. For the final $m5$ model, the probability-based metrics from small German Bloom models remain highly significant. Interestingly, we observe a similar effect here to other studies on surprisal (Oh and Schuler, 2023), i.e. their predictive power decreases with increasing model size. Similarly, we see some advantages of the smaller XGLM560 over the larger XGLM1b7. Finally, next to model size, we see great differences between predictors computed from different transformer architectures (BERT, GPT2, T5). For instance, AH-JSD from BERT remains significant in $m5$, while the same is not true for T5 or GPT-2. This suggests that attention patterns learned in different architectures capture different aspects of humans' cognitive processes, supporting further research into novel LM architectures (Charpentier and Samuel, 2024).

Finally, we complement the chi-square-based evaluation with the model ranking according to AIC scores (quantifying model fitness) in Appendix Figure B.5. The AIC-based analysis confirms the pattern described above. Among all variations for the base model ($m3$), AH-JSD metric derived from BERT had the highest model fit. However, for the enriched models incorporating semantic similarity ($m4$) and retrieval latency ($m5$), larger models – particularly BLOOM1b5 and XGLM560 – demonstrate superior performance.

5.4 Discussion

Our experiments on verbal fluency add to the existing evidence that language models show some degree of human-likeness in their internal processing mechanisms (cf. Kuribayashi et al., 2025). Thus, we find that well-known predictors derived from LMs' word predictions, i.e., surprisal and related measures, as well as predictors computed from LMs' attention distributions, have strong statistical power when separating between clustering and switching in human verbal fluency responses.

For research on creativity in human cognition, this result supports the assumption that different processes are at play in creative semantic search tasks (Ovando-Tellez et al., 2022). When LMs regenerate humans' verbal fluency responses, they show clearly distinct attention and prediction patterns that neatly align with annotations of clustering and switching in these sequences. Previous studies identified these patterns based on distances in word embedding spaces (Alacam et al., 2022). Our study complements this with further metrics computed, in particular, from the LMs' internal attention distribution. These attention-based LM predictors remained significant even when added to a baseline model that included a semantic distance-based variable (IRS). This suggests that attention distributions capture processing-related mechanisms in verbal fluency beyond semantic distances.

The fact that attention-based predictors are superior to probability-based metrics in our verbal fluency setting supports previous work proposing that attention patterns in transformer LMs could reflect processes or retrieval and memory search (Ryu and Lewis, 2021; De Varda and Marelli, 2024). The creative search processes involved in verbal fluency pose particularly strong demands on memory and executive processes of working memory and inhibition (Shao et al., 2014). This further underlines the plausibility of our findings and explains why surprisal predictors, which are prominent in studies on processing difficulty in natural reading, show less consistent patterns than attention-based metrics.

While recent work on cognitive probing of LMs has mostly focused on autoregressive GPT-style architectures, our results show that attention predictors from encoder models like BERT outperform GPT models. This is surprising since autoregressive word prediction and causal, left-to-right self-attention seem intuitively more aligned with incremental sequence generation in verbal fluency. A hypothesis to explore in future work is that the bidirectional self-attention in the BERT encoder could allow the model to obtain a richer semantic space and account for more complex attention and retrieval operations involved in a challenging semantic search task.

Finally, our study points to new directions for the cognitive probing of LMs. Whereas most previous work looked at modelingreading times, our study shows the fitness of LM predictors in accounting for generative and creative tasks. Future tasks to consider could be related to naming (Silberer et al., 2020), reference (Junker and Zarrieß, 2024) or association (Chen and Ding, 2023).

6 Conclusion

Our work contributes to understanding the processing mechanisms of LMs with the help of verbal fluency, an established experimental task from cognitive science research. We showed that LMs can distinguish two central components of creative sematic search, clustering and switching, via their metrics derived from their attention and probability distributions. Our study is one of the first to show that distributions of attention weights in the internal layers and attention heads of the transformer architecture correlate to a great extent with processing difficulty in a creative semantic search task.

Limitations

We have employed the vanilla versions of the selected language models and all the metrics derived from the models were not subjected to heavy transformations except the basic soft-max, negative log-likelihood, or pooling over layers and attention heads. Since the evidence from the analysis points towards the advantage of using attention-based metrics, further investigation on calculating different attention scores (Oh and Schuler, 2022) is a promising line of research.

The verbal fluency data were processed using off-the-shelf NLP text processing tools. Compound words are generally common in German, and the vocabulary used by participants also frequently contains compound words such as "Klavierspielen" (piano playing), "Krankenpfleger" (health nurse), "Fahrradfahren" (bike riding). Unfortunately, many of the compounds do not exist in the vocabulary of the static embedding models such as ConceptNet, whereas BERT and succeeding language models can deal with out-of-vocabulary tokens due to their sub-word tokenization method.

Ethical Statement

Our study utilizes a published and openly available dataset with annotations on verbal fluency, without annotator-related information. Additionally, we ensure that our use of the dataset aligns with its intended purpose.

Acknowledgments

We thank the anonymous reviewers for their helpful and detailed comments. This research has been partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – CRC-1646, project no. 512393437, project B02. The authors also acknowledge financial support by the project "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine-Westphalia, Germany.

References

Joshua T Abbott, Joseph L Austerweil, and Thomas L Griffiths. 2015. Random walks on semantic networks can resemble optimal foraging. In *Neural Information Processing Systems Conference; A preliminary version of this work was presented at the aforementined conference.*, volume 122, page 558. American Psychological Association.

Özge Alacam, Simeon Schüz, Martin Wegrzyn, Johanna Kißler, and Sina Zarrieß. 2022. Exploring semantic spaces for detecting clustering and switching in verbal fluency. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 178–191, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Johnathan Avery and Michael N Jones. 2018. Comparing models of semantic fluency: Do humans forage optimally, or walk randomly? In *CogSci*.

Marco Baroni. 2022. On the proper role of linguistically-oriented deep net analysis in linguistic theorizing. *Algebraic structures in natural language*, pages 1–16.

Roger E. Beaty and Yoed N. Kenett. 2023. Associative thinking at the core of creativity. *Trends in Cognitive Sciences*, 27(7):671–683.

Roger E Beaty and Paul J Silvia. 2012. Why do ideas get more creative across time? an executive interpretation of the serial order effect in divergent thinking tasks. *Psychology of aesthetics, creativity, and the arts*, 6(4):309.

Roger E Beaty, Paul J Silvia, Emily C Nusbaum, Emanuel Jauk, and Mathias Benedek. 2014a. The roles of associative and executive processes in creative cognition. *Memory & cognition*, 42:1186–1197.

Roger E. Beaty, Paul J. Silvia, Emily C. Nusbaum, Emanuel Jauk, and Mathias Benedek. 2014b. The roles of associative and executive processes in creative cognition. *Memory amp; Cognition*, 42(7):1186–1197.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand gpt-3. *Proceedings of the National Academy of Sciences*, 120(6):e2218523120.

Tyler A. Chang and Benjamin K. Bergen. 2023. Language Model Behavior: A Comprehensive Survey. *Computational Linguistics*, pages 1–55.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. BERT or GPT: why not both? In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 262–283, Miami, FL, USA. Association for Computational Linguistics.

Honghua Chen and Nai Ding. 2023. Probing the "creativity" of large language models: Can models produce divergent semantic association? In *Findings* *of the Association for Computational Linguistics: EMNLP 2023*, pages 12881–12888, Singapore. 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.

Andrea De Varda and Marco Marelli. 2024. Locally biased transformers better align with human reading times. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 30–36, Bangkok, Thailand. Association for Computational Linguistics.

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

Eleni Demetriou and Roee Holtzer. 2017. Mild cognitive impairments moderate the effect of time on verbal fluency performance. *Journal of the International Neuropsychological Society*, 23(1):44–55.

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.

Kenneth J Gilhooly, Evridiki Fioratou, Susan H Anthony, and Victor Wynn. 2007. Divergent thinking: Strategies and executive involvement in generating novel uses for familiar objects. *British Journal of Psychology*, 98(4):611–625.

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, Salt Lake City, Utah. Association for Computational Linguistics.

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*.

David Heineman, Reba Koenen, and Sashank Varma. 2024. Towards a path dependent account of category fluency. In *Proceedings of the 46th Annual Meeting of the Cognitive Science Society*.

Thomas T Hills, Michael N Jones, and Peter M Todd. 2012. Optimal foraging in semantic memory. *Psychological review*, 119(2):431.

Simeon Junker and Sina Zarrieß. 2024. Resilience through scene context in visual referring expression generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 344–357, Tokyo, Japan. Association for Computational Linguistics.

Ivana Kajić, Jan Gosmann, Brent Komer, Ryan W. Orr, Terrence C. Stewart, and Chris Eliasmith. 2017. A biologically constrained model of semantic memory search. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*, London, UK. Cognitive Science Society.

Najoung Kim, Jung-Ho Kim, Maria K. Wolters, Sarah E. MacPherson, and Jong C. Park. 2019. Automatic scoring of semantic fluency. *Frontiers in Psychology*, 10.

Tatsuki Kuribayashi, Yohei Oseki, Ana Brassard, and Kentaro Inui. 2022. Context limitations make neural language models more human-like. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10421–10436, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tatsuki Kuribayashi, Yohei Oseki, Souhaib Ben Taieb, Kentaro Inui, and Timothy Baldwin. 2025. Large language models are human-like internally. *arXiv preprint arXiv:2502.01615*.

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

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Nicklas Linz, Johannes Tröger, Jan Alexandersson, and Alexandra Konig. 2017. Using Neural Word Embeddings in the Analysis of the Clinical Semantic Verbal Fluency Task. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS 2017)*, pages 1–7, Montpellier, France).

Martin Marko, Drahomír Michalko, Jozef Dragašek, Zuzana Vančová, Dominika Jarčušková, and Igor Riečanskỳ. 2023. Assessment of automatic and controlled retrieval using verbal fluency tasks. *Assessment*, 30(7):2198–2211.

Sarnoff Mednick. 1962. The associative basis of the creative process. *Psychological review*, 69(3):220.

Drahomír Michalko, Martin Marko, and Igor Riečanskỳ. 2023. Executive functioning moderates the decline of retrieval fluency in time. *Psychological Research*, 87(2):397–409.

Animesh Nighojkar, Anna Khlyzova, and John Licato. 2022. Cognitive modeling of semantic fluency using transformers. In *Cognitive Aspects of Knowledge Representation workshop at 31st International Joint Conference on Artificial Intelligence*.

Byung-Doh Oh, Christian Clark, and William Schuler. 2022. Comparison of structural parsers and neural language models as surprisal estimators. *Frontiers in Artificial Intelligence*, 5:777963.

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. 2023. Why does surprisal from larger transformer-based language models provide a poorer fit to human reading times? *Transactions of the Association for Computational Linguistics*, 11:336–350.

Marcela Ovando-Tellez, Mathias Benedek, Yoed N Kenett, Thomas Hills, Sarah Bouanane, Matthieu Bernard, Joan Belo, Theophile Bieth, and Emmanuelle Volle. 2022. An investigation of the cognitive and neural correlates of semantic memory search related to creative ability. *Communications Biology*, 5(1):604.

Felipe Paula, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. 2018. Similarity measures for the detection of clinical conditions with verbal fluency tasks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 231–235, New Orleans, Louisiana. Association for Computational Linguistics.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

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

Zeshu Shao, Esther Janse, Karina Visser, and Antje S. Meyer. 2014. What do verbal fluency tasks measure? predictors of verbal fluency performance in older adults. *Frontiers in Psychology*, 5:1–10.

Carina Silberer, Sina Zarrieß, Matthijs Westera, and Gemma Boleda. 2020. Humans meet models on object naming: A new dataset and analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1893–1905, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Paul J. Silvia, Roger E. Beaty, and Emily C. Nusbaum. 2013. Verbal fluency and creativity: General and specific contributions of broad retrieval ability (gr) factors to divergent thinking. *Intelligence*, 41(5):328–340.

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

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, et al. 2024. Testing theory of mind in large language models and humans. *Nature Human Behaviour*, 8(7):1285–1295.

Angela K Troyer, Morris Moscovitch, and Gordon Winocur. 1997. Clustering and switching as two components of verbal fluency: evidence from younger and older healthy adults. *neuropsychology*, 11(1):138.

Ye Wang, Yaling Deng, Ge Wang, Tong Li, Hongjiang Xiao, and Yuan Zhang. 2025. The fluency-based semantic network of llms differs from humans. *Computers in Human Behavior: Artificial Humans*, 3:100103.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Jill Weckerly, Beverly Wulfeck, and Judy Reilly. 2001. Verbal fluency deficits in children with specific language impairment: Slow rapid naming or slow to name? *Child Neuropsychology*, 7(3):142–152.

Douglas M Whiteside, Tammy Kealey, Matthew Semla, Hien Luu, Linda Rice, Michael R Basso, and Brad Roper. 2016. Verbal fluency: language or executive function measure? *Applied Neuropsychology: Adult*, 23(1):29–34.

Ethan Wilcox, Clara Meister, Ryan Cotterell, and Tiago Pimentel. 2023. Language model quality correlates with psychometric predictive power in multiple languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7503–7511, Singapore. Association for Computational Linguistics.

Jeffrey C Zemla and Joseph L Austerweil. 2017. Modeling semantic fluency data as search on a semantic network. In *CogSci... Annual Conference of the Cognitive Science Society. Cognitive Science Society (US). Conference*, volume 2017, page 3646. NIH Public Access.

Jeffrey C Zemla and Joseph L Austerweil. 2019. Analyzing knowledge retrieval impairments associated with alzheimer's disease using network analyses. *Complexity*, 2019.

Appendix

A BIEFU data

Table 4 presents basic statistics for word counts and retrieval latencies for BIEFU verbal fluency sequences within each category and across categories (as *global*). This overview highlights some characteristic differences between the categories: participants enumerated almost 11.5 items on average. For the *animals* and *countries*, the number is high as 19.11 and 18.5 respectively, while it is around or below 10 items for *fabrics*, *insects*, and *vessels*. Correspondingly, retrieval latency for *countries*, *animals*, *groceries* and *body parts* are significantly lower than categories that are less easy to enumerate such as *fabrics or insects*.

Table 4 also includes typicality and IRS scores that we will use as predictors in our baseline model. The IRS is the cosine similarity between consecutive words calculated with ConceptNet Numberbatch embeddings (Speer et al., 2017). We observe that the categories *insects* and *fabrics* which elicited the smallest number of words (tokens and types) across participants show the lowest typicality values, i.e. participants retrieved relatively few and rather divergent sets of words. Interestingly, *hobbies* and *occupations* exhibit high typicality, i.e. show more overlap between participants, but also show the lowest IRS scores, i.e. they contain words that have more distant embedding in semantic space. The categories *clothes*, *body parts*, *insects*, and *vessels* exhibit the highest IRS scores. Based on the provided dataset, we further calculate the retrieval latencies between each consecutive items. The mean retrieval latencies shown in Table 4 further differentiate the overall picture. Here, the categories *countries* and *animals*, the most widely used category in verbal fluency, show the lowest mean retrieval latencies, together with high typicality and medium IRS.

Task demands For creating the task demand categories for BIEFU in a similar way as in Michalko et al. (2023), we have looked at the held-out sequences (from another 100 participants on the same categories, but without retrieval latency scores) and calculated the basic statistics similar to Table 4 except the retrieval latency score. Based on these scores, we categorized the BIEFU categories into three groups depending on the cognitive effort needed to enumerate them. The low-demand category consists of *animals*, *body parts* and *countries*. *Hobbies*, *occupations*, *groceries* and *clothes* belong to the moderate category. Finally, the high demand category includes *fabrics*, *vessels* and *insects*.

B Language Models

We utilize the verbal fluency data in German by (Alacam et al., 2022) and we employ various distinct language models for German : (i) a pretrained German BERT model[1] (ii) a pretrained German GPT-2 model[2] , and (iii) a pretrained T5 model[3] for German.

In this way, we aim to minimize any potential impact of the training data's nature on the overall performance of our models. We generally use the Hugging Face[4] framework for reproducibility.

Next to these common LMs, we evaluate two more recent autoregressive models on the dataset, investigating the effects of model size and the difference between monolingual and multilingual language models. Specifically, we employ (i) a monolingual BLOOM model that is trained from scratch on German data, comprising 350M parameters[5],

---

[1] https://huggingface.co/dbmdz/bert-base-german-cased.

[2] https://huggingface.co/dbmdz/german-gpt2.

[3] https://huggingface.co/t5-base.

[4] https://huggingface.co/.

[5] https://huggingface.co/malteos/bloom-350m-german.

Table 4: BIEFU: Basic statistics (Max, min, and average values of sequences, retrieval latency and sub-category counts per semantic category)

Categories	Token Count in a Sequence	Mean Retrieval Latency (in sec.)	Total Token (Type) Count	Subcat. Count	Typicality (mean)	IRS Similarity (mean)
animals	**Max: 34, Min: 8, Mean: 19.11**	1,96	1548 (202)	22	4.53	.39
body parts	**Max: 28 , Min: 8 , Mean: 17.02**	2.50	1571 (144)	8	3.98	.50
clothes	Max: 24, Min: 7, Mean: 16.5	2.31	1434	15	4.04	.52
countries	**Max: 33, Min: 6, Mean: 18.5**	1.81	1688 (140)	6	4.19	.42
fabrics	Max: 14, Min: 5, Mean: 7.9	**5.06**	633 (142)	15	3.94	.39
groceries	Max: 28, Min: 7, Mean: 16.6	2.32	1550 (276)	14	4.69	.42
hobbies	Max: 25, Min: 6, Mean: 14.49	2.63	1333 (302)	31	4.86	.32
insects	Max: 17, Min: 5, Mean: 9.47	**4.21**	843 (99)	14	3.67	.49
occupations	Max: 20, Min: 6, Mean: 12.23	2.89	1113 (296)	19	4.91	.35
vessels	Max: 17, Min: 5, Mean: 10.13	3.83	902 (166)	9	4.13	.46
Global	Max: 34, Min: 5, Mean: 11.51	3.05	19518 (2763)	153	4.13	.43

(ii) a multilingual BLOOM model adapted to the German language via the CLP-Transfer method with 1.5B parameters[6], and (iii) a multilingual BLOOMoom with 1.7B parameters[7]. Furthermore, we use (iv) a multilingual XGLM model with 564M parameters[8], comparable in size to the monolingual BLOOM model, and (v) a multilingual XGLM model with 1.7B parameters[9], equivalent in size to the biggest multilingual BLOOM model.

We omit models like Chat-GPT or GPT-4 from our analysis since these do not generally provide token probabilities or attention distributions through their respective APIs and, hence, do make it possible to compute the type of measures and predictors we need for our investigation.

B.1 Tokenization

We first tokenize the masked prompt with the word w masked out by a single mask token m) and pass it through the model. We then restrict the output logits of the model to the position of the masked token and pass them through a softmax function to obtain a probability distribution over the model's vocabulary for the position of m. In the resulting distribution, we select the probability of w, the entropy of the distribution as well as the rank of w in the model's vocabulary sorted by the probability. In addition to this, we also store the attention map over the whole sequence. The subword tokenization of BERT and T5 complicates this process, i.e. w is not always represented by a single token in the model's vocabulary, but may consist of multiple subword tokens (such as *[Kol, ##ib, ##ri]* for the word *Kolibri (hummingbird)*). In such cases, we iteratively replace m with each subword token for w and take the average of the log probabilities of all subwords as well as the lowest rank of any subword as representative of the whole item w. Such a method is considered useful for extracting a more meaningful score for the multiword expressions like *[Großer Panda (Big Panda), Rote Paprika (Red paprika)]*. For the autoregressive GPT-2, BLOOM and XGLM models, where utilizing a masked token isn't feasible, we truncate the prompt at the position of the masked item and then pass it through the models. The process of extracting probabilities, ranks, surprisal scores, and entropies with GPT-2, BLOOM and XGLM models mirrors that are utilized for BERT and T5 models. This also extends to the handling of the subword tokens, as the autoregressive models employ the same tokenization strategy.

B.2 Prompt Design

Since existing LMs can be very sensitive to the specification of their prompts, we also test several prompt variations for the calculation of probabilities and attention distributions for verbal fluency sequences. Depending on the type of LM, these prompts can be divided into (i) unidirectional prompts that only include left context for masked tokens and (ii) bidirectional prompts where masked tokens are presented in a left and right context. In the following, we describe the design of the verbal fluency prompts.

---

[6] `https://huggingface.co/malteos/bloom-1b5-clp-german`.

[7] `https://huggingface.co/bigscience/bloom-1b7`.

[8] `https://huggingface.co/facebook/xglm-564M`.

[9] `https://huggingface.co/facebook/xglm-1.7B`.

Table 5: A sample of a human response and derived LM prompts for two subsequent steps in a verbal fluency sequence (1st step/left, 2nd step/right column), as input for autoregressive prompting. For T5, we use identical prompts to BERT but replace [MASK] with the sentinel token.

Original Sequence	Hund (dog), Katze (cat), Maus (mouse)	
Target token	Katze in the 1st step	Maus in the 2nd step
Prompt-0	*(Animals: Dog, [MASK])* Tiere: *Hund, [MASK]**	Tiere: *Hund, Katze, [MASK]**
Prompt-1	*(Animals I know are dog and [MASK].)* Tiere, die ich kenne, sind *Hund und* [MASK]*	Tiere, die ich kenne, sind *Hund, Katze und* [MASK]*
Prompt-2	*(Examples of animals are dog, [MASK])* Beispiele für Tiere sind *Hund und [MASK]*.*	Beispiele für Tiere sind *Hund, Katze, und [MASK]*.*
Prompt-3	*(The first animals that come to my mind are dog, [MASK], mouse.)* Die ersten Tiere, die mir einfallen, sind *Hund und [MASK]*.*	Die ersten Tiere, die mir einfallen, sind *Hund, Katze und [MASK]*.*
Prompt-4	*(Animals one can know are dog and [MASK].)* Tiere, die man kennt, sind *Hund und* [MASK]*	Tiere, die ich kenne, sind *Hund, Katze und* [MASK]*
Prompt-5	*(When I think of animals, I think of dog and [MASK].)* Wenn ich an Tiere denke, dann denke ich an *Hund und* [MASK]*	Wenn ich an Tiere denke, dann denke ich an *Hund, Katze und* [MASK]*

B.3 Hard Switches

Table 6 summarizes the results for the hard switch modeling when the LM metrics are added to $m3$, $m4$ and $m5$ models.

Unlike soft-switch modeling, the contribution of various metrics in this specific case of switches varies significantly, without exhibiting a consistent pattern across all conditions. A closer examination reveals that among the probability-based metrics, RANK and SURPRISAL are the most influential, often performing on par with AH-JSD or even surpassing it in modeling hard-switch cases. It is important to note that a hard switch occurs when a previously unmentioned subcategory appears in the enumeration. This necessitates metrics that are sensitive to a broader contextual lookback.

Overall, for detecting hard-switches, probability-based metrics demonstrate greater predictive power in decoder-only models, whereas models with encoders benefit substantially from AH-JSD. Further details on these results are provided in Appendix B.3.

Psycholinguistic parameters. In the hard switch condition, adding the retrieval order parameter (TEMP) improves model fit ($\mathcal{X}^2(1) = 11.58, p < .001$). The task demand also significantly improves the model ($\mathcal{X}^2(2) = 6.97.87, p < .0001$). The main effects of typicality (TYP) ($\mathcal{X}^2(1) = 19.76, p < .001$) and the inter-response similarity parameter (IRS) also significantly contributed to explaining the data ($\mathcal{X}^2(1) = 2990.75, p < .0001$) as well as the retrival latency.

m3 + LM predictors. It is obvious that A closer look reveals that among the probability-based metrics, Rank and Surprisal are the most prominent ones except the GPT-2, Bloom1b5 and Bloom1b7 models. Furthermore, all attention-based metrics contribute significantly to the model fit to a differing extent. Despite not having the highest contribution, almost all metrics derived from XGLM adds explanatory power.

m4 + LM predictors . When we look at the effect of LM metrics for the model with IRS, it is also difficult to see one distinct pattern. Again, Rank and Surprisal parameters are generally more informative than probability or entropy metrics. Bloom1b7 seems to have no contribution on top of basic psycholinguistic parameters. $Entropy$ only contributes to the fitness for Bloom350m.

m5 + LM predictors. In addition to the de facto psycholinguistic parameters, we investigate the effect of a less common parameter in verbal fluency analysis – the retrieval latency – as an indicator of lexical computation in explaining switching /clustering behavior. Then we also examine the alignment between retrieval latency with the LM predictors. To do that, we add the retrieval latency to the $m4$ model. In the both hard and soft switch conditions, we find that the retrieval latency RL further improves the model fitness significantly: ($\mathcal{X}^2(1) = 344.88, p < .001$) and ($\mathcal{X}^2(1) = 265.17, p < .001$) respectively.

As summarized in Table 6, the Bloom350 model continues to exhibit a significant effect for its probability-based metrics, followed by Bloom 1b5. Attention-based metrics continue to contribute to the model fitness only for the BERT model, on top of retrieval latency.

Table 6: Hard Switch: the individual contribution of LM-predictors on top of $m3$, $m4$ and $m5$ models (Chi-Square)

		BERT	**T5**	**GPT-2**	**Bloom350**	**Bloom1b5**	**Bloom1b7**	**XGLM560**	**XGLM1b7**
$m3$	**Prob**	49.67 ***	9.95 **	1.72	0.32	0.37	2.06	19.69 ***	27.02***
	Rank	12.75 **	**57.89*****	1.67	44.32 **	0.52	7.65**	**94.86*****	**66.05*****
	Surprisal	**107.08*****	0.06	9.86**	**66.82** **	31.30***	2.07	76.61 **	27.25***
	Entropy	5.24*	0.61	2.05	24.61***	0.89	0.33	21.87**	2.32
	AHE	37.12**	24.97***	18.17**	16.40**	16.40**	16.36**	24.64***	24.64***
	AH-JSD	73.34***	54.89***	**43.45*****	40.03***	**37.96****	**45.91****	53.31***	48.43***
$m4$	**Prob**	0.10	23.33 ***	18.62 ***	49.31 ***	41.76***	0.01	0.87	**28.64*****
	Rank	7.14 **	11.76 ***	**21.49** ***	32.58 ***	26.23***	**19.05****	**32.53*****	23.52***
	Surprisal	39.25 ***	9.73 **	0.52	**78.20*****	**89.56*****	0.8	14.11***	10.71**
	Entropy	1.50	0.04	2.37	30.98 ***	1.09	0.14	22.05***	4.87*
	AHE	24.40 ***	13.69 ***	8.49 **	7.40 **	7.40**	5.14*	11.65**	11.65***
	AH-JSD	**43.37** ***	**28.45** ***	16.12 ***	16.22 ***	14.09**	16.58**	21.68***	18.13***
$m5$	**Prob**	0.04	**24.80** ***	20.04 ***	48.93 ***	43.01***	0.003	0.48	31.65***
	Rank	7.25 **	9.77 **	**23.08** ***	29.23 ***	21.22***	**15.87*****	**34.57*****	22.44***
	Surprisal	**35.21** ***	8.62 **	1.55	**75.08** ***	**88.46*****	1.15	17.65***	11.8**
	Entropy	2.67	0.03	3.57	24.31 ***	1.77	0.07	20.69***	**38.9***
	AHE	2.28 ***	0.15	0.06	0.37	0.37	1.26	0.00	0.001
	AH-JSD	6.41 *	1.67	0.00	0.01	0.17	0.04	0.11	0.02

B.4 Effect of Vocabulary size on the Rank Parameter

Among the explored metrics, the rank score is highly dependent on the vocabulary size of the language model. Figure 2 plots the predictive power (Chi2) of the RANK parameter when added to the to $m3$, $m4$ and $m5$ models. On the left, the graph shows models with smaller vocabulary sizes (BERT, T5); in the center, models with (relatively) moderate vocabulary sizes (GPT-2, Bloom350m, Bloom1b5); and on the right side, multilingual models with substantially larger vocabulary sizes (Bloom1b7 and XGLM models). This graphs reveals that when the rank is added to $m3$ model, no clear pattern is observed with respect to vocabulary size. On the other hand, when the rank score is added to the m4 and m5 models, there is a slight upward, suggesting a possible relation between rank score and the vocabulary size. However, this trend is still not consistent across models with similar vocabulary sizes.

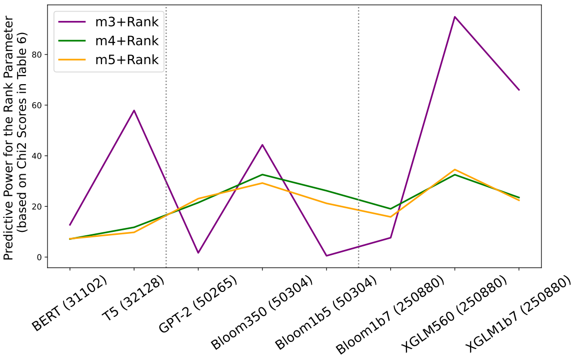
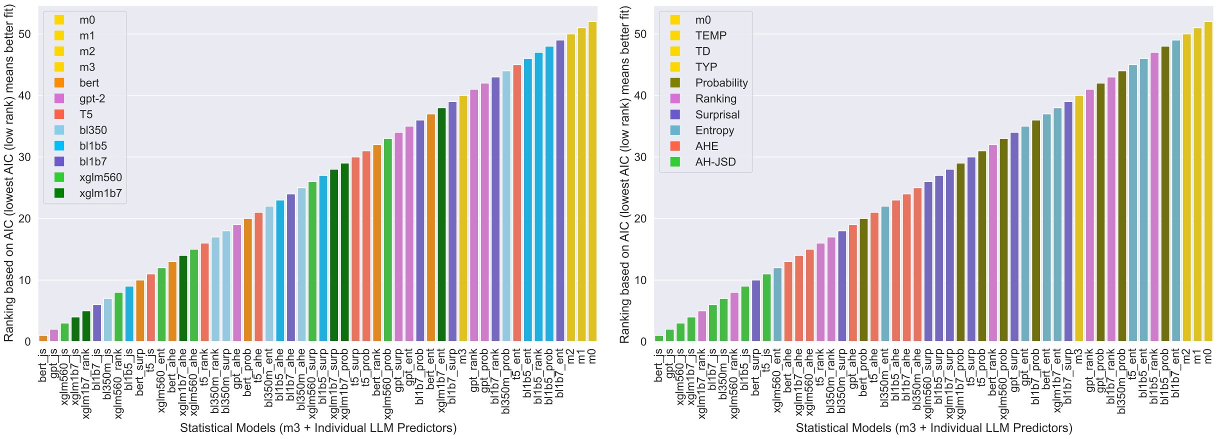


Figure 2: LMs architectures ordered w.r.t their vocabulary size. Y-axes denotes the Chi2 Scores for the rank parameter added to $m3$, $m4$ and $m5$ models.

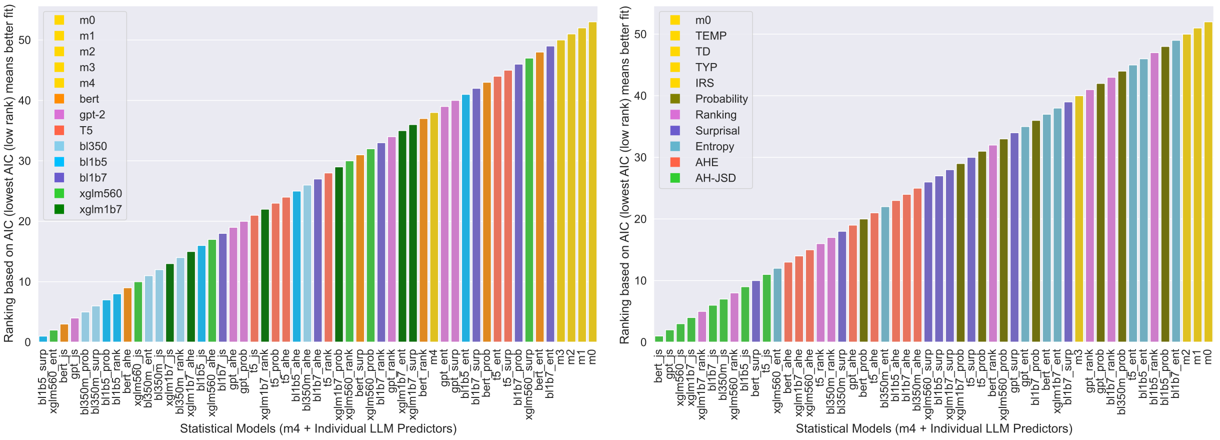
B.5 AIC Based Ranking

Complementary results for the Section 5.2. While the sub-figures positioned next to each other show the same data, they highlight the different aspects: for example, Figure B.5 (a) is color-coded with respect to the LM type, and Figure B.5 (b) for the effect of the metric. The lowest AIC corresponds to the lowest rank (1st rank/best model).



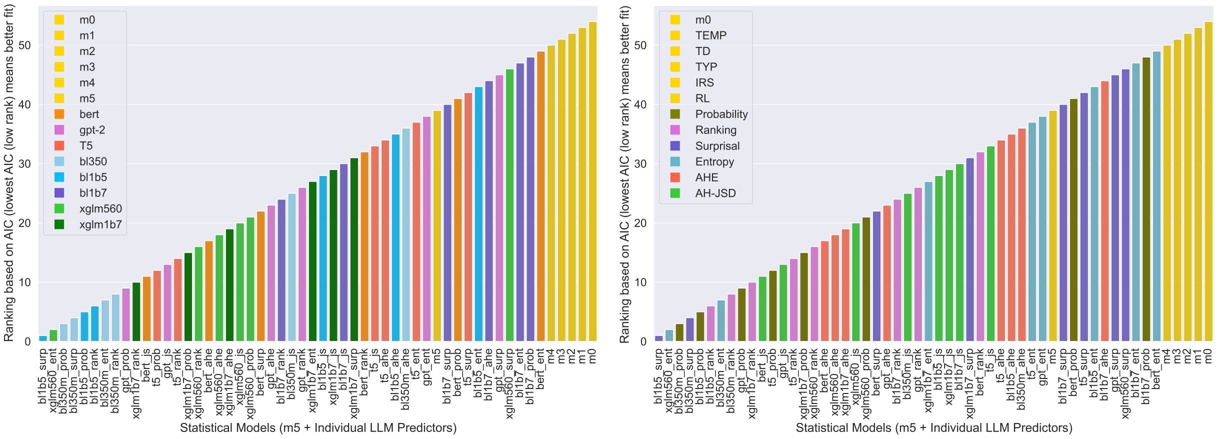
(a) $m3$: Model-based color coded

(b) $m4$: Metric-based color coded



(c) $m4$: Model-based color coded

(d) $m4$: Metric-based color coded



(e) $m5$: Model-based color coded

(f) $m5$: Metric-based color coded

Figure 3: Individual Models' fitness (based on AIC scores

An Appraisal Theoretic Approach to Modelling Affect Flow in Conversation Corpora

Alok Debnath and **Yvette Graham** and **Owen Conlan**
ADAPT Centre
School of Computer Science and Statistics
Trinity College Dublin
`{debnatha, ygraham, owen.conlan}@tcd.ie`

Abstract

This paper presents a model of affect in conversations by leveraging Appraisal Theory as a generalizable framework. We propose that the multidimensional cognitive model of Appraisal Theory offers significant advantages for analyzing emotions in conversational contexts, addressing the current challenges of inconsistent annotation methodologies across corpora. To demonstrate this, we present AppraisePLM, a regression and classification model trained on the crowd-EnVent corpus that outperforms existing models in predicting 21 appraisal dimensions including *pleasantness*, *self-control*, and *alignment with social norms*. We apply AppraisePLM to diverse conversation datasets spanning task-oriented dialogues, general-domain chit-chat, affect-specific conversations, and domain-specific affect analysis. Our analysis reveals that AppraisePLM successfully extrapolates emotion labels across datasets, while capturing domain-specific patterns in affect flow – change in conversational emotion over the conversation. This work highlights the entangled nature of affective phenomena in conversation and positions affect flow as a promising model for holistic emotion analysis, offering a standardized approach to evaluate and benchmark affective capabilities in conversational agents.[1]

1 Introduction

Affect, which encompasses both emotion and mood, is crucial in conversations, influencing dynamics such as empathy, sarcasm, and naturalness (Ruusuvuori, 2012). In the domain of conversational agents (CAs), recognizing and responding to affective cues is essential (Skowron and Paltoglou, 2011; Yang et al., 2019). Various methodologies are employed for incorporating affect into CAs, including emotion classification, dimensional ratings, intent annotations, and vicarious emotion ratings such as empathy and condolence (Busso et al., 2008; Ma et al., 2020; Karna et al., 2020). While affect-annotated datasets exist across general and specialized domains, inconsistencies in annotation schemas and objectives pose challenges for standardizing affect modelling in conversational AI (Liu et al., 2021; Islam et al., 2022).

These inconsistencies arise due to variations in annotation methodologies, including differences in unit-level labelling (e.g., turn-wise versus full-conversation annotations) and dataset construction depending on the domain (Liu et al., 2024). Moreover, evaluation metrics for contextual affect interactions remain limited, coercing a generalization of findings across datasets. Many domain-specific models, such as diff-EPITOME (Lee et al., 2022), are trained within a specific domain but later applied broadly, highlighting the need for standardized affect evaluation (Schaaff et al., 2023). A generalizable framework for modelling affect in conversations could address these challenges, ensuring more consistent benchmarking for conversational agents.

This paper proposes that **Appraisal Theory** provides such a generalizable framework. Appraisal theory conceptualizes emotions as responses to an individual's evaluation of a stimulus along multiple cognitive dimensions (Ellsworth and Smith, 1988; Scherer, 2005). For example, anger can be characterized as an unpleasant, short-lived emotion with low self-control (Roseman and Smith, 2001). Such an approach not only allows for the modelling of emotional intensity and duration but also enables the analysis of *affect flow*, or how emotions evolve throughout a conversation (Hendriks et al., 2014; Poria et al., 2019b).

In this paper, we hypothesize that: (**H1**) appraisal-theoretic emotion analysis aligns with existing emotion annotations; and that (**H2**) such a cognitive analysis captures affect flow: emotion

---

[1] Code is available here: `https://github.com/alokdebnath/appraise-PLM`

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 233–250
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

change over the course of a conversation. To test these hypotheses, the paper introduces **AppraisePLM**, a model for appraisal regression and emotion classification, trained on the crowd-EnVent corpus. Crowd-EnVENT is a benchmark emotion recognition and appraisal analysis corpus which provides fine-grained annotations of event descriptions on 21 appraisal dimensions including *pleasantness*, *self-control*, and *suddenness* (Troiano et al., 2023).[2]

Our model outperforms existing classifiers and regressors on this dataset and is subsequently applied to turn-wise appraisal annotation across four benchmark conversation corpora: EmoWOZ (Feng et al., 2022), EMPATHETICDIALOGUES (Rashkin et al., 2019), DailyDialog (Li et al., 2017), and EPITOME (Sharma et al., 2020). Our results show that AppraisePLM improves appraisal estimation performance on the crowd-EnVent corpus and can extrapolate categorical and emotion labels. Additionally, corpus domain influences affect flow, with distinct patterns emerging in specific domains (e.g., empathetic conversations improving *pleasantness*). Through this paper, we highlight the intertwined nature of affective phenomena and argue towards developing appraisal theory as an interpretable intradomain model of emotion in conversation.

2 Background and Motivation

2.1 Emotion Recognition in Conversation

Emotion recognition in conversation (ERC) often relies on Plutchik's wheel or Ekman's universal emotions for annotation (Plutchik, 1965; Ekman, 2000). Commonly used general-domain dialogue corpora, such as DailyDialog (Li et al., 2017), MELD (Poria et al., 2019a), and EmotionLines (Hsu et al., 2018), employ a set of basic emotions like joy, fear, sadness, anger, surprise, disgust, and neutral. However, some corpora use varying numbers of emotion categories, ranging from fine-grained annotations to broader affect labels (Qin et al., 2023). The veracity and similarity of emotions can differ significantly by domain, raising questions about the accessibility and identification of fine-grained emotions in conversation (Hancock et al., 2007; Machová et al., 2023).

The Valence-Arousal-Dominance (VAD) model is a prevalent dimensional model for emotion, with IEMOCAP serving as a reference corpus providing both dimensional and categorical emotion labels (Busso et al., 2008). The conversation corpus' domain heavily influences the taxonomy and distribution of emotion labels (Rajapakshe et al., 2024). For instance, mental health-focused corpora may prioritize certain emotions over others compared to general-domain corpora (Saha et al., 2022). Additionally, factors such as access to different modalities and the number of participants in the conversation can impact the emotion annotation methodology as well (Pereira et al., 2023, 2025). Appendix provides a table to show the inconsistencies across emotion annotations in conversation corpora.

2.2 Affective Phenomena in Conversation

The emotions expressed and perceived by interlocutors influence expected conversational behavior, though modeling "emotion shift" remains an open problem (Pereira et al., 2025). Corpora often use direct annotation methods to extract relevant affective features and behaviours. For instance, EMPATHETICDIALOGUES is a benchmark open-domain empathetic conversation corpus that uses 32 fine-grained emotion labels, also applied in EDOS (Rashkin et al., 2019).

Some domain-specific corpora, such as EPITOME (Sharma et al., 2020), ALoE (Yang and Jurgens, 2024), PAIR (Pérez-Rosas et al., 2022), and ESConv (Liu et al., 2021) in the mental health domain, do not directly annotate emotion. Instead, they assess characteristics of empathetic interactions using direct annotator ratings, like *Emotional Interpretation* in EPITOME. This approach allows models to access desirable interactional behaviours without relying solely on emotion (Lahnala et al., 2022). Metrics from PAIR and EPITOME have been used to benchmark open-domain conversational agents, expecting these behaviours in general-domain contexts (Lee et al., 2024). For example, a general-purpose conversational agent should provide condolence, implying expected linguistic behaviour with an affective signal (Zhou and Jurgens, 2020). The manner and display of empathy vary with context, relationship, and personality, as noted in the PEC corpus (Zhong et al., 2020).

2.3 Appraisal Theory in Language and Conversation Analysis

Appraisal theory posits that experienced emotions result from cognitive appraisals of event stimuli,

---

[2] The Appraisal dimensions are defined and categorized in Appendix B

such as pleasantness, suddenness, controllability, or alignment with social norms (Ellsworth and Smith, 1988). This theory offers a view of an experiencer's cognitive state by systematically choosing context-appropriate appraisals.

Appraisal theory has gained prominence in NLP and conversation analysis, enhancing emotion classification and interpersonal communication studies (Balahur et al., 2011; Hofmann et al., 2020). In NLP, it improves emotion classification accuracy through dimensional models and annotated corpora, aiding in understanding how emotions arise from event evaluations (Troiano et al., 2022; Resendiz and Klinger, 2023). In conversation analysis, appraisal theory reveals how speakers express attitudes and manage relationships through evaluative language.

The theory has also had use in analyzing motivational interviewing, with the ALoE dataset focusing on empathetic alignment in therapeutic conversations using appraisal theory (Yang and Jurgens, 2024). However, this work is domain-specific and lacks correspondence with other categorical or dimensional labels.

Troiano et al. (2023) introduced the crowd-enVENT dataset, consisting of 6,600 emotion-inducing event descriptions annotated with 21 appraisal dimensions, emotion labels, and author demographics. This dual-perspective annotation allows for comparing appraisal and emotion reconstruction by readers versus computational models, providing a human baseline for machine learning tasks. Unlike ISEAR, crowd-enVENT was compiled specifically for text analysis, studying the relationship between appraisals, emotions, and event descriptions.

3 The AppraisePLM Framework

In this section, we propose AppraisePLM, an appraisal-theoretic conversation analysis framework which estimates the aggregate change(s) and patterns in how the interlocutors appraise the conversation over time. First, we test the cross-comparability of appraisals with other representations (§3.1-3.3) then provide the methodology to do the same for conversations (§3.4).

3.1 Problem Definition

Given a dataset $D = \{(e_i, l_i, c_i)\}_{i=1}^{N}$ where e_i is the ith event description (text), $l_i = [l_i^1, l_i^2, ..., l_i^a]$ is a vector of a event description appraisals, and $c_i \in \mathbb{C}$ is a label from the set of n emotion class labels $\mathbb{C}$, we perform the following two tasks.

Appraisal Estimation Train a function $f_{app} : \mathbb{R}^d \rightarrow \mathbb{R}^a$ where d is the dimensionality of the encoded event description PLM(e_i) and a is the number of appraisals. The objective of this function is to find $\theta_{app}* = \arg\min_{\theta_{app}} \mathcal{L}_{app}$ such that:

$$\mathcal{L}_{app} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{a}\sum_{j=1}^{a}(f_{app}(\text{PLM}(e_i))_j - l_i^j)^2$$

Emotion Classification Upon appraisal estimation, train a function $f_{emo} : \mathbb{R}^d \times \mathbb{R}^a \rightarrow \mathbb{C}$, where d is the dimensionality of the encoded event description PLM(e_i), $\mathbb{C}$ is a set of n emotion class labels, and a is the number of appraisals. The objective of this function is to find $\theta_{emo}\star = \arg\min_{\theta_{emo}} \mathcal{L}_{emo}$ such that:

$$\text{comb} = \text{PLM}(e_i) \oplus f_{app}(\text{PLM}(e_i))_k$$

$$\mathcal{L}_{emo}^{comb} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{n} 1_{[c_i=k]} * \ln f_{emo}(\text{comb})$$

3.2 Dataset Characteristics

The crowd-EnVENT dataset consists of 6,600 event descriptions (550 event descriptions for 13 emotion labels). Each event is annotated with 21 appraisal variables, which are cognitive evaluations of the event by the event's author. The fine-grained emotion labels allow us to analyze how experiencers appraise various emotions (including a `no-emotion` label). The distribution of appraisal values is skewed, more than 33% of the corpus being either 1 or 5. Their approach for appraisal classification involves a two-class classificaiton, which we do not use as the differences in appraisal values are a critical step in AppraisePLM.

3.3 Model Framework

The proposed AppraisePLM multitask framework jointly performs appraisal regression and emotion classification using attention-attenuated pretrained language models (PLMs) such as RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2020), MP-Net (Song et al., 2020) and T5 (Raffel et al., 2020); with DeBERTa yielding the best performance. Figure 1 provides a simple schematic of the model architecture.

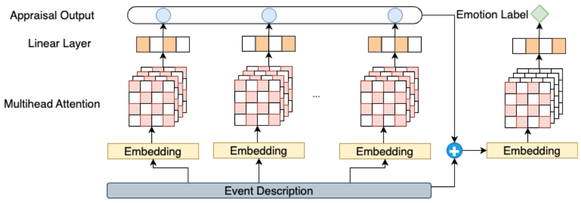


Figure 1: Model architecture for the AppraisePLM framework.

	T → A.	**T→ E**	**T + A → E**
Model	MSE ↓	F1 ↑	F1 ↑
Troiano et. al. (2022)	1.97	0.59	0.60
RoBERTa-large	1.62	0.59	0.67
T5-large	1.12	0.61	0.66
MPNet-base	**1.08**	0.64	0.70
DeBERTa-large	**1.08**	0.66	**0.71**

Table 1: Performance of the AppraisePLM architecture for appraisal estimation and emotion classification. Emotion classification is done in two modes; with only text (**T → E**) and both text and appraisals (**T + A → E**) on the crowd-EnVent corpus.

Corpus	**P**	**R**	**F1**
EMPATHETICDIALOGUES	0.77	0.79	0.78
DailyDialog	0.63	0.66	0.62
EmoWOZ	0.62	0.56	0.59

Table 2: Zero-shot emotion classification performance of AppraisePLM (DeBERTa-large; T + A → E) model on conversation corpora with emotion labels. Comparisons are done after label folding, a smaller subset of the crowd-EnVENT emotion labels are mapped to the labelling schema of the corpus.

The event description is embedded using a PLM encoder and a multihead attention layer. Each appraisal dimension has a task-specific multihead attention layer and linear head. Regression is trained with individual MSE losses for all appraisal values.

Emotion classification utilizes both the PLM representation and predicted appraisal values. The encoded event description is concatenated with the predicted appraisal values, normalized and regularized before being decoded by another multihead attention layer and a linear classification head. Classification is trained on cross-entropy loss.

We use an AdamW optimizer with a weight decay 0.01 and a learning rate 2e−5. We use a standard grid search for hyperparameter tuning. Training employs Distributed Data Parallel (DDP)[3] on four RTX 2080 Ti GPUs, with a batch size of 16 and gradient checkpointing, early stopping within three epochs with a maximum training of ten epochs. Reproducibility report is provided in **??**.

3.4 AppraisePLM Results and Performance

Table 1 presents the test set performance of the AppraisePLM architecture on the crowd-enVENT dataset, compared to the baseline model. While attention attenuation marginally improves appraisal estimation, DeBERTa-large achieves the highest performance. However, the limited improvement in regression scores reflects the task's complexity (see Appendix D).

Appraisal representations enhance categorical emotion detection, with event descriptions appended with appraisal information yielding a 0.11 macro avg. F1 improvement over the baseline. Multi-head attention slightly improves standard emotion classification (**T → E**. in Table 6), but the AppraisePLM architecture shows a more substantial boost when integrating both text and appraisal data.

Figure 6 visualizes appraisal estimates across emotions using DeBERTa-large AppraisePLM, with emotions ordered by *pleasantness*. As expected, *no-emotion* separates positive and negative emotions, with *joy* being the most pleasant and *disgust* the least. Unpleasantness follows the inverse trend, while *urgency*, *attention*, and *other-control* exhibit minimal variation across emotions.

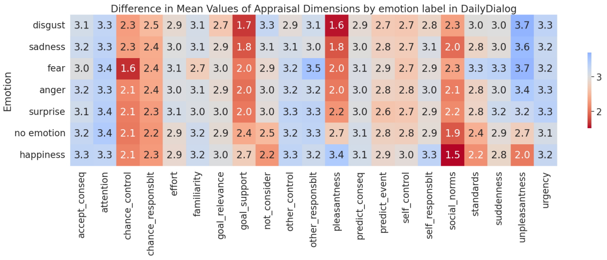
4 Affect Annotations in Dialogue Corpora

In this section, the applicability of AppraisePLM on conversational corpora is examined using four datasets: EmoWOZ, DailyDialog, EMPATHETICDIALOGUES, and EPITOME. These datasets vary in emotion annotation schemes, label counts, and domains, so the analysis considers each corpus individually while maintaining methodological consistency.

AppraisePLM estimates the appraisal dimension for dialogue turn and concatenates them with utterance embeddings for zero-shot emotion classification. The DeBERTa-large AppraisePLM model is used for annotation. Due to differing labelling schemas for some datasets, label folding is applied, and a co-occurrence Emotion category similarities with crowd-EnVENT are assessed, and relevant labels are retained for weighted F1-score evaluation.

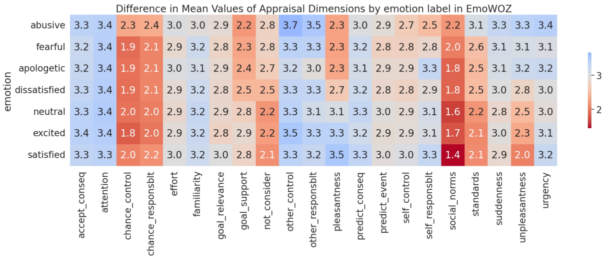
---

[3] `https://pytorch.org/docs/stable/generated/torch.nn.parallel.DistributedDataParallel.html`



Emotion	accept_conseq	attention	chance_control	chance_responsblt	effort	familiarity	goal_relevance	goal_support	not_consider	other_control	other_responsblt	pleasantness	predict_conseq	predict_event	self_control	self_responsblt	social_norms	standards	suddenness	unpleasantness	urgency
disgust	3.1	3.3	2.3	2.5	2.9	3.1	2.7	1.7	3.3	2.9	3.1	1.6	2.9	2.7	2.7	2.8	2.3	3.0	3.0	3.7	3.3
sadness	3.2	3.3	2.3	2.4	3.0	3.1	2.9	1.8	3.1	3.1	3.0	1.8	3.0	2.8	2.7	3.1	2.0	2.8	3.0	3.6	3.2
fear	3.0	3.4	1.6	2.4	3.1	2.7	3.0	2.0	2.9	3.2	3.5	2.0	3.1	2.9	2.7	2.9	2.3	3.3	3.3	3.7	3.2
anger	3.2	3.3	2.1	2.4	3.0	3.1	2.9	2.0	3.0	3.2	3.2	2.0	3.0	2.8	2.8	3.0	2.1	2.8	3.0	3.4	3.3
surprise	3.1	3.4	2.1	2.3	3.1	3.0	3.0	2.0	3.0	3.3	3.3	2.2	3.0	2.6	2.7	2.9	2.2	2.8	3.2	3.2	3.3
no emotion	3.2	3.4	2.1	2.2	2.9	3.2	2.9	2.4	2.5	3.2	3.3	2.7	3.1	2.8	2.8	2.9	1.9	2.4	2.9	2.7	3.1
happiness	3.3	3.3	2.1	2.3	2.9	3.2	3.0	2.7	2.2	3.3	3.2	3.4	3.1	2.9	3.0	3.3	1.5	2.2	2.8	2.0	3.2

(a) Dataset: DailyDialog



emotion	accept_conseq	attention	chance_control	chance_responsblt	effort	familiarity	goal_relevance	goal_support	not_consider	other_control	other_responsblt	pleasantness	predict_conseq	predict_event	self_control	self_responsblt	social_norms	standards	suddenness	unpleasantness	urgency
abusive	3.3	3.4	2.3	2.4	3.0	3.0	2.9	2.2	2.8	3.7	3.5	2.3	3.0	2.9	2.7	2.5	2.2	3.1	3.3	3.3	3.4
fearful	3.2	3.4	1.9	2.1	2.9	3.2	2.8	2.3	2.8	3.3	3.3	2.3	3.2	2.8	2.8	2.8	2.0	2.6	3.1	3.1	3.1
apologetic	3.2	3.4	1.9	2.1	3.0	3.1	2.9	2.4	2.7	3.2	3.0	2.3	3.1	2.9	2.9	3.3	1.8	2.5	3.1	3.2	3.2
dissatisfied	3.3	3.4	1.9	2.1	2.9	3.2	2.8	2.5	2.5	3.3	3.3	2.7	3.2	2.8	2.8	2.9	1.8	2.5	3.0	2.8	3.0
neutral	3.3	3.4	2.0	2.0	2.9	3.2	2.9	2.8	2.2	3.3	3.1	3.1	3.3	3.0	2.9	3.1	1.6	2.2	2.8	2.5	3.0
excited	3.4	3.4	1.8	2.0	2.9	3.2	2.8	2.9	2.2	3.5	3.3	3.3	3.2	2.9	2.9	3.1	1.7	2.1	3.0	2.3	3.1
satisfied	3.3	3.3	2.0	2.2	3.0	3.2	3.0	2.8	2.1	3.3	3.2	3.5	3.3	3.0	3.0	3.3	1.4	2.1	2.9	2.0	3.2

(b) Dataset: EmoWOZ

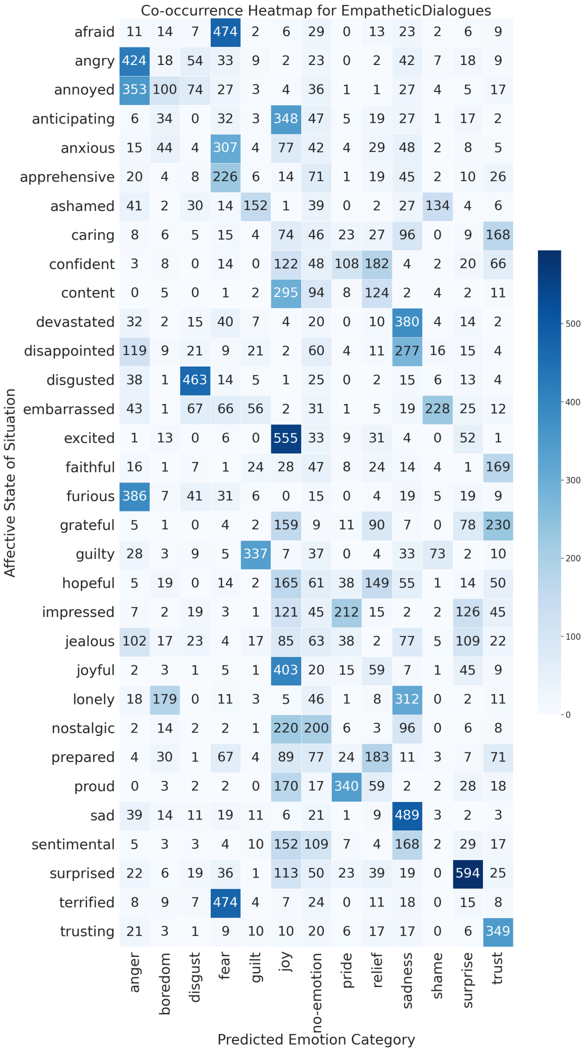
Figure 2: Average estimate of each appraisal from the DailyDialog and EmoWOZ test corpora using the best performing AppraisePLM DeBERTa-large. The emotion labels are ordered by *pleasantness* from low (red) to high (blue).

4.1 DailyDialog

The DailyDialog dataset is a high-quality, manually labeled, multi-turn dialogue dataset designed to reflect everyday communication. It contains 13,118 dialogues, with an average of approximately 8 speaker turns per dialogue. The dataset covers various topics related to daily life, providing a diverse range of conversational context and includes manual annotations for topics, dialogue acts, and emotion.

DailyDialog uses a six class emotion classification (*anger*, *fear*, *disgust*, *happiness*, *surprise*, *sadness*) along with a *no-emotion*. The latter is almost 80% of the corpus, while in the emotion labelled turns, 74% of them are labelled *happiness*. This label skew affected AppraisePLM's performance. Since the DailyDialog emotion categories are a subset of Plutchik's categories, no label folding or merging was performed, computing a strict macro weighted F1 score of 0.62 for emotion classification using AppraisePLM DeBERTa-large.

Figure 2a shows the average distribution of appraisal values across emotion labels for the DailyDialog corpus. We see that these appraisals are similar to the appraisal distribution by emotion label for crowd-EnVENT, except the average valence of the *no emotion* label and the slightly higher *pleasantness* and *unpleasantness* estimates of *disgust*- and *fear*-labelled conversation turns.



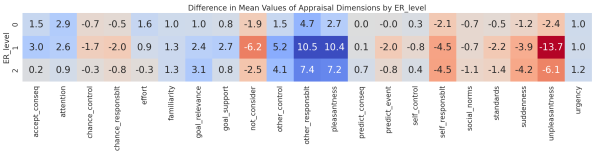
Affective State of Situation	anger	boredom	disgust	fear	guilt	joy	no-emotion	pride	relief	sadness	shame	surprise	trust
afraid	11	14	7	474	2	6	29	0	13	23	2	6	9
angry	424	18	54	33	9	2	23	0	2	42	7	18	9
annoyed	353	100	74	27	3	4	36	1	1	27	4	5	17
anticipating	6	34	0	32	3	348	47	5	19	27	1	17	2
anxious	15	44	4	307	4	77	42	4	29	48	2	8	5
apprehensive	20	4	8	226	6	14	71	1	19	45	2	10	26
ashamed	41	2	30	14	152	1	39	0	2	27	134	4	6
caring	8	6	5	15	4	74	46	23	27	96	0	9	168
confident	3	8	0	14	0	122	48	108	182	4	2	20	66
content	0	5	0	1	2	295	94	8	124	2	4	2	11
devastated	32	2	15	40	7	4	20	0	10	380	4	14	2
disappointed	119	9	21	9	21	2	60	4	11	277	16	15	4
disgusted	38	1	463	14	5	1	25	0	2	15	6	13	4
embarrassed	43	1	67	66	56	2	31	1	5	19	228	25	12
excited	1	13	0	6	0	555	33	9	31	4	0	52	1
faithful	16	1	7	1	24	28	47	8	24	14	4	1	169
furious	386	7	41	31	6	0	15	0	4	19	5	19	9
grateful	5	1	0	4	2	159	9	11	90	7	0	78	230
guilty	28	3	9	5	337	7	37	0	4	33	73	2	10
hopeful	5	19	0	14	2	165	61	38	149	55	1	14	50
impressed	7	2	19	3	1	121	45	212	15	2	2	126	45
jealous	102	17	23	4	17	85	63	38	2	77	5	109	22
joyful	2	3	1	5	1	403	20	15	59	7	1	45	9
lonely	18	179	0	11	3	5	46	1	8	312	0	2	11
nostalgic	2	14	2	2	1	220	200	6	3	96	0	6	8
prepared	4	30	1	67	4	89	77	24	183	11	3	7	71
proud	0	3	2	2	0	170	17	340	59	2	2	28	18
sad	39	14	11	19	11	6	21	1	9	489	3	2	3
sentimental	5	3	3	4	10	152	109	7	4	168	2	29	17
surprised	22	6	19	36	1	113	50	23	39	19	0	594	25
terrified	8	9	7	474	4	7	24	0	11	18	0	15	8
trusting	21	3	1	9	10	10	20	6	17	17	0	6	349

Figure 3: A co-occurrence heatmap of predicted emotion category and annotated emotion label for the EMPATHETICDIALOGUES corpus. Emotion categories are predicted for emotionally grounded situations.

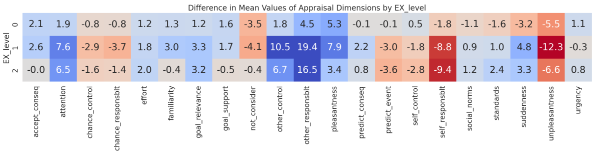
4.2 EMPATHETICDIALOGUES (ED)

The EmpatheticDialogues (ED) dataset comprises 24,850 one-to-one open-domain conversations, with 2,457 in the test set analyzed here. Each conversation features a speaker sharing a personal emotional experience and a listener responding empathetically. The dataset includes 32 fine-grained emotions, with 5.1% tagged as "surprised" and 1.9% as "faithful", and test set conversations averaging 4.2 turns.

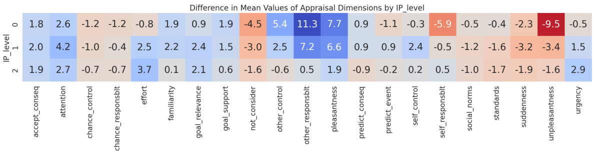
Since ED uses a custom emotion list, AppraisePLM's emotion detection is evaluated using a coarser emotion set. Figure 3 shows that the model effectively distinguishes broad emotional



(a) Appraisal Change by Emotional Reaction



(b) Appraisal Change by Emotion Expression



(c) Appraisal Change by Interpretation

Figure 4: The change in appraisal estimate between the speaker and response posts of the EPITOME dataset. Change in appraisal estimates is computed as $a_r^3 - a_s^3$ where a_r is the average response appraisal and a_s is the seeker appraisal, scaled for trend analysis.

categories (e.g., "afraid," "anxious," "apprehensive," and "terrified" all align with "fear"). It also identifies theoretical correlations across annotation schemas (e.g., "lonely" and "annoyed" strongly correlate with "boredom"). Synonym-based label folding results are reported in Table 5.

Appendix Figure 7 presents appraisal estimates of emotions, ordered by pleasantness, showing similarities with the crowd-EnVENT corpus (Figure 6). The ordering of emotions reflects their perceived intensity or arousal (e.g., "furious" vs. "angry" and "disgusted" vs. "annoyed"). Notably, while "devastated" is among the most unpleasant, it is not the least pleasant and exhibits higher goal support than more negatively valenced emotions. Additionally, the range of appraisal estimates in ED is narrower than in crowd-EnVENT.

4.3 EPITOME

The EPITOME dataset is designed to examine empathy in text-based, asynchronous conversations, incorporating both emotional and cognitive aspects. It consists of 10,000 post-response pairs sourced from online platforms such as Reddit and TalkLife, annotated along three dimensions—Emotional Reaction (ER), Interpretation (IP), and Exploration (EX)—each rated on a 0-2 scale: ER demonstrates warmth, compassion, or concern, IP reflects an understanding of inferred feelings and experiences, and EX explores aspects of the seeker's experience not explicitly stated. Since these annotations rely on comparisons between posts, whereas Appraise-PLM annotates individual turns, we reinterpret the dimensions through differences in cognitive appraisals. Specifically: **High ER** corresponds to increased *pleasantness* and *other-responsibility* while decreasing *unpleasantness* and *self-responsibility*; **High IP** implies minimal change in appraisal values, ensuring emotional alignment with the seeker, and **High EX** suggests differences in *other-control*, *other-responsibility*, and *self-responsibility* between seeker and response, showing a distinct but similar affect.

The heatmap analysis (Fig. 4) highlights two key findings: (1) ER and EX ratings of 1 show greater shifts in appraisals than ratings of 2, and (2) IP ratings of 2 correspond to the lowest average appraisal shifts, indicating stronger alignment between seeker and response posts.

4.4 EmoWOZ

The EmoWOZ dataset is a large-scale, manually emotion-annotated corpus of task-oriented dialogues, derived from MultiWOZ. It is designed to examine how user emotions impact task-oriented dialogue systems. EmoWOZ contains 11,434 dialogues, including both human-human (MultiWOZ) and human-machine (DialMage) dialogues. The analysis focuses on the test set.

EmoWOZ employs a custom emotion labelling scheme for task-oriented dialogues, with seven labels: neutral, satisfied, dissatisfied, excited, apologetic, fearful, and abusive, adapted from the OCC emotion model. Due to differences in domain and classification intent, these labels do not directly correlate between corpora, with "neutral" being overwhelmingly dominant.

Figure 2b shows distinct appraisal profiles across emotion labels. The "neutral" category serves as a separator between positive and negative states. We can see that emotion ordering by pleasantness aligns with emotional valence. The range of appraisal values in EmoWOZ is lower than in other conversational datasets, likely due to the task-oriented nature of dialogues, which exhibit less emotional variability than chit-chat. Categorical labeling alone would not highlight such differences effectively. Table 2 indicates that emotion detection is more challenging in EmoWOZ, partly due to label imbalance, with notably fewer "abusive" and "dissatisfied" conversation turns.

5 Affect Flow in Conversation

So far, we have applied appraisal theory to examine emotion at the level of individual dialogue turns. In this section, we extend our analysis to the conversation level by modelling *affect flow*—that is, how appraisal dimensions evolve throughout a multi-turn interaction. This expanded analysis introduces three additional considerations: speaker interaction dynamics, the gradient of appraisal change across conversation turns, and the emotion-specificity of appraisal dimensions.

Speaker interaction dynamics are addressed by distinguishing the initiator and responder roles in conversations. We track appraisal shifts separately for each interlocutor, allowing us to capture role-specific patterns in emotional evolution. As in our prior analysis with EPITOME (§4.3), we apply a power function transformation to enhance subtle but consistent variations in appraisal values, making it easier to detect meaningful affective trends over time.

Gradient analysis enables us to compare how specific appraisal values change turn-by-turn, particularly across corpora. While the average appraisal values in a corpus may exhibit minimal variation—largely reflecting the dominant emotional tone of that dataset (often neutral or mildly positive)—certain appraisals or their combinations remain predictive of emotion labels. Building on this, we align comparable emotions across corpora by identifying their most indicative appraisal dimensions, and then compare the rate and direction of change in these dimensions over conversation turns.

Emotion-specificity of appraisals, the notion that distinct appraisal dimensions are reliably associated with specific emotion labels, can be extended to study affect flow. By statistically comparing the gradient trajectories of these key appraisal dimensions, we identify emotion-specific patterns of conversational change. For instance, we observe that in the EMPATHETICDIALOGUES corpus, conversations labelled with *anger* tend to exhibit a rapid shift toward more neutral appraisals in the subsequent turn, indicating quick pacification. In contrast, conversations labelled with *furious* show more gradual shifts.

We refer to Figure 5 to examine the change in appraisals over time for a sample of the corpora.

EmoWOZ has the highest gradient and lowest central tendency for appraisal estimates, indicating large fluctuations in emotion appraisals over a conversation ((Figure 5a, 5b)). Conversations labelled satisfied exhibit strong positive valence shifts, with both speakers increasing appraisals of *pleasantness* and *goal support* over turns. In contrast, conversations labelled *dissatisfied* show an amplifying effect for positive appraisals and a dampening effect for negative appraisals by the second interlocutor, highlighting a different form of emotional adaptation compared to the other corpora.

EmpatheticDialogues (ED) contains the shortest conversations on average and shows low variation in appraisal shifts between turns (Figure 5e, 5b). Conversations in this corpus display empathic matching (Wondra and Ellsworth, 2015) for both positive and negative emotions: speakers and listeners tend to align their appraisals over time, leading to appraisal gradients closer to zero. The *happy/joyful* category exhibits strong alignment, consistent with theoretical expectations of interactional empathy, where interlocutors appraise events similarly over successive turns.

DailyDialog (DD) exhibits higher variation in appraisal gradients, particularly for negative emotions, suggesting that emotional shifts are more dynamic ((Figure 5c, 5d)). Unlike ED, where emotion directionality is clear (seeker vs. provider), DD does not enforce speaker roles. Either participant can elicit emotion, leading to non-uniform affect flow. Despite this variability, a general trend of appraisal convergence is observed over time, particularly for emotions like joy and sadness, although sadness shows a distinct decrease in unpleasantness near the end of conversations.

EPITOME Unlike the other corpora, EPITOME exhibits appraisal shifts where emotional convergence occurs but with different dominant appraisal dimensions. While *pleasantness* and *unpleasantness* remain key indicators, dimensions such as *self-responsibility*, *other-responsibility*, attention (for emotional expression), and *not consider* (for emotional reaction) play a larger role in distinguishing response quality. Higher quality responses, as measured by reaction, interpretation, and expression ratings, show distinct appraisal characteristics, reinforcing the importance of nuanced appraisal dynamics in emotion modelling.

6 Discussion

Appraisal theory, as a model of emotion realized in text, is based on post-hoc or simulated appraisals of cognitive dimensions correlated with universal emotion labels. We approximate conversation segments (situation, turn, or response) as event descriptions, assuming that post-hoc contextual rating of appraisals preserves the relationship between semantic and cognitive representations of affect. Our analysis of conversational corpora using appraisal estimation yields mixed quantitative results but offers promising qualitative insights. Label inconsistencies complicate the evaluation of emotion detection in AppraisePLM.

In this section, we examine the veracity of the hypotheses mentioned in §1.

6.1 H1: Aligning with Extant Affect Annotations

In **H1**, we hypothesized that appraisal-theoretic emotion analysis aligns with existing emotion annotations. We tested this at multiple levels by examining patterns of appraisal estimates for the overall corpus, characterized by its domain and annotation level (conversation, turn, or response).

We found that fully textual corpora, such as EMPATHETICDIALOGUES and DailyDialog, exhibit significantly higher alignment in categorical labels between the AppraisePLM emotion classification and existing annotations. This finding is notable, as both corpora have different approaches and goals for affect annotation. However, the domain of affect annotation poses challenges for quantitative analysis.

For instance, in the EPITOME corpus, changes in appraisal estimates between utterance and response align with the definitions of the annotated dimensions, while appraisal-informed emotion classification reflects the source of the conversation. A similar domain effect is observed in EmoWOZ, where emotion classification scores after label folding were baseline, but trends in appraisal by turn and speaker correspond to action states in the corpus.

In summary, appraisal theory shows reasonable alignment with existing affect annotations in conversational corpora, providing additional cognitive insights. Using appraisal theory as the grounding emotion annotation in general domain conversations would significantly improve the performance and reliability of the AppraisePLM approach.

6.2 H2: Appraisal Change as Affect Flow

In **H2**, we hypothesized that cognitive analysis captures affect flow, examined as emotion change over the course of a conversation. We observed that not all appraisals are relevant to a conversation or domain and may change minimally. However, those appraisals that do change exhibit a small but consistent gradient when aggregated over the conversation.

Section §5 details findings from one approach to examining affect flow using the power-amplified difference of appraisal estimates between conversation turns. Appraisal gradients differ by dataset: EMPATHETICDIALOGUES exhibits low appraisal shifts, DailyDialog shows greater variability in negative emotions, and EmoWOZ presents the highest appraisal gradient with distinct trends for satisfaction and dissatisfaction. Empathic matching, where interlocutors align appraisals over time, is evident in positive emotions across EMPATHETICDIALOGUES and DailyDialog. However, DailyDialog lacks directional speaker roles, leading to broader variability in emotional elicitation. The EPITOME corpus demonstrates distinct appraisal relevance, with dimensions like responsibility and attention influencing response quality.

In summary, changes in appraisal estimates represent emotion change in conversation. The multidimensionality and cognitive nature of appraisal theory reinforce its utility in emotion modeling and highlight corpus-specific affective dynamics.

7 Conclusion

In this paper, we introduced AppraisePLM, a multitask learning model designed to estimate appraisal dimensions and emotion categories using the crowd-EnVENT dataset. By applying our model to various conversational corpora, we leveraged its fine-grained dimensional representation of emotion to analyze affect flow—the subtle evolution of emotions within a conversation as it progresses.

Our findings demonstrate that appraisal theory provides a valuable framework for examining how emotions manifest in conversational data. While the crowd-EnVENT dataset is not a dialogue corpus, our results support the feasibility of using appraisal-based models to examine emotion dynamics in conversation. We observed not only improved appraisal estimation and emotion classification performance over baseline models but also reasonable success in appraisal-informed zero-shot emotion classification.

We identify two key applications for this approach. First, benchmarking affective conversational agents, such as those designed for emotional support or empathetic interaction, by assessing how their responses modulate appraisal dimensions. Second, informing agent response expectations when expressing specific emotions, offering insights into emotionally intelligent dialogue systems. These findings highlight the potential of AppraisePLM in advancing computational approaches to emotion modeling and affective dialogue analysis.

Limitations

This work has several important limitations that should be acknowledged. Firstly, we assume that appraisal annotation for conversations occurs in the same way as for statements or short-form text, which may not always be the case. Additionally, we presume that emotions are appraised similarly in human-human and human-machine interaction contexts, an assumption that requires further investigation. The granularity of our approach, while providing more detailed insights, also increases the potential for errors. We utilized 21 dimensions because it was possible, but future research should determine which of these dimensions are most applicable and relevant. Our current system employs power amplification of differences between values to identify interaction trends, which could be critiqued as potentially highlighting insignificant variations. A more robust approach would involve the development and use of conversationally defined and annotated corpora based on appraisal theory, given its relationship to and generality of emotion categorization systems. Lastly, the lack of longitudinal data prevents us from observing how appraisal patterns might change over time in ongoing human-machine interactions. Addressing these limitations in future research will be crucial for advancing our understanding of emotion appraisal in human-machine conversations.

Acknowledgement

This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at Trinity College Dublin. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by Science Foundation Ireland through the SFI Research Centres Programme. The authors also thank the anonymous reviewers for their insightful reviews and recommendations for improving the paper.

References

Alexandra Balahur, Jesus M Hermida, and Andres Montoyo. 2011. Building and exploiting emotinet, a knowledge base for emotion detection based on the appraisal theory model. *IEEE transactions on affective computing*, 3(1):88–101.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42:335–359.

Paul Ekman. 2000. Basic emotions. *Handbook of Cognition and Emotion*, page 45.

Phoebe C Ellsworth and Craig A Smith. 1988. From appraisal to emotion: Differences among unpleasant feelings. *Motivation and emotion*, 12(3):271–302.

Shutong Feng, Nurul Lubis, Christian Geishauser, Hsien-Chin Lin, Michael Heck, Carel van Niekerk, and Milica Gasic. 2022. Emowoz: A large-scale corpus and labelling scheme for emotion recognition in task-oriented dialogue systems. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4096–4113.

Jeffrey T Hancock, Christopher Landrigan, and Courtney Silver. 2007. Expressing emotion in text-based communication. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 929–932.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hanneke Hendriks, Bas van den Putte, and Gert-Jan de Bruijn. 2014. Changing the conversation: The influence of emotions on conversational valence and alcohol consumption. *Prevention Science*, 15:684–693.

Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. 2020. Appraisal theories for emotion classification in text. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 125–138.

Chao-Chun Hsu, Sheng-Yeh Chen, Chuan-Chun Kuo, Ting-Hao Huang, and Lun-Wei Ku. 2018. Emotionlines: An emotion corpus of multi-party conversations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Md Adnanul Islam, Md Saddam Hossain Mukta, Patrick Olivier, and Md Mahbubur Rahman. 2022. Comprehensive guidelines for emotion annotation. In *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents*, pages 1–8.

Mounika Karna, D Sujitha Juliet, and R Catherine Joy. 2020. Deep learning based text emotion recognition for chatbot applications. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 988–993. IEEE.

Allison Lahnala, Charles Welch, David Jurgens, and Lucie Flek. 2022. A critical reflection and forward perspective on empathy and natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2139–2158.

Andrew Lee, Jonathan Kummerfeld, Larry Ann, and Rada Mihalcea. 2024. A comparative multidimensional analysis of empathetic systems. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–189.

Young-Jun Lee, Chae-Gyun Lim, and Ho-Jin Choi. 2022. Does gpt-3 generate empathetic dialogues? a novel in-context example selection method and automatic evaluation metric for empathetic dialogue generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 669–683.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995.

Siyang Liu, Chujie Zheng, Orianna Demasi, Sahand Sabour, Yu Li, Zhou Yu, Yong Jiang, and Minlie Huang. 2021. Towards emotional support dialog systems. 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 3469–3483.

Tingting Liu, Salvatore Giorgi, Ankit Aich, Allison Lahnala, Brenda Curtis, Lyle Ungar, and João Sedoc. 2024. The illusion of empathy: How ai chatbots shape conversation perception. *arXiv preprint arXiv:2411.12877*.

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

Yukun Ma, Khanh Linh Nguyen, Frank Z Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

Kristína Machová, Martina Szabóova, Ján Paralič, and Ján Mičko. 2023. Detection of emotion by text analysis using machine learning. *Frontiers in Psychology*, 14:1190326.

Patrícia Pereira, Helena Moniz, and Joao Paulo Carvalho. 2025. Deep emotion recognition in textual conversations: A survey. *Artificial Intelligence Review*, 58(1):1–37.

Patrícia Pereira, Helena Moniz, Isabel Dias, and Joao Paulo Carvalho. 2023. Context-dependent embedding utterance representations for emotion recognition in conversations. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 228–236.

Verónica Pérez-Rosas, Kenneth Resnicow, Rada Mihalcea, et al. 2022. Pair: Prompt-aware margin ranking for counselor reflection scoring in motivational interviewing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 148–158.

Robert Plutchik. 1965. What is an emotion? *The Journal of psychology*, 61(2):295–303.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. Meld: A multimodal multi-party dataset for emotion recognition in conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 527–536.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE access*, 7:100943–100953.

Xiangyu Qin, Zhiyu Wu, Tingting Zhang, Yanran Li, Jian Luan, Bin Wang, Li Wang, and Jinshi Cui. 2023. Bert-erc: Fine-tuning bert is enough for emotion recognition in conversation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 13492–13500.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Thejan Rajapakshe, Rajib Rana, Sara Khalifa, and Björn W Schuller. 2024. Domain adapting deep reinforcement learning for real-world speech emotion recognition. *IEEE Access*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Yarik Menchaca Resendiz and Roman Klinger. 2023. Affective natural language generation of event descriptions through fine-grained appraisal conditions. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 375–387.

Ira J Roseman and Craig A Smith. 2001. Appraisal theory. *Appraisal processes in emotion: Theory, methods, research*, pages 3–19.

Johanna Ruusuvuori. 2012. Emotion, affect and conversation. *The handbook of conversation analysis*, pages 330–349.

Tulika Saha, Vaibhav Gakhreja, Anindya Sundar Das, Souhitya Chakraborty, and Sriparna Saha. 2022. Towards motivational and empathetic response generation in online mental health support. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 2650–2656.

Kristina Schaaff, Caroline Reinig, and Tim Schlippe. 2023. Exploring chatgpt's empathic abilities. In *2023 11th international conference on affective computing and intelligent interaction (ACII)*, pages 1–8. IEEE.

Klaus R Scherer. 2005. What are emotions? and how can they be measured? *Social science information*, 44(4):695–729.

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276.

Marcin Skowron and Georgios Paltoglou. 2011. Affect bartender—affective cues and their application in a conversational agent. In *2011 IEEE Workshop on Affective Computational Intelligence (WACI)*, pages 1–7. IEEE.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Enrica Troiano, Laura Oberländer, and Roman Klinger. 2023. Dimensional modeling of emotions in text with appraisal theories: Corpus creation, annotation reliability, and prediction. *Computational Linguistics*, 49(1):1–72.

Enrica Troiano, Laura Oberländer, Maximilian Wegge, and Roman Klinger. 2022. x-envent: A corpus of event descriptions with experiencer-specific emotion and appraisal annotations. In *13th International Conference on Language Resources and Evaluation Conference, LREC 2022*, pages 1365–1375. European Language Resources Association (ELRA).

Joshua D. Wondra and Phoebe C. Ellsworth. 2015. An appraisal theory of empathy and other vicarious emotional experiences. *Psychological Review 122.3*, pages 4–11.

Jiamin Yang and David Jurgens. 2024. Modeling empathetic alignment in conversation. 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 3127–3148.

Xi Yang, Marco Aurisicchio, and Weston Baxter. 2019. Understanding affective experiences with conversational agents. In *proceedings of the 2019 CHI conference on human factors in computing systems*, pages 1–12.

Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards persona-based empathetic conversational models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6556–6566.

Naitian Zhou and David Jurgens. 2020. Condolence and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 609–626.

A A Review of Emotion Annotations in Conversational Corpora

Table 3 shows the wide range of contemporary emotion classification and affect-annotated datasets. We see that there is little consistency in the emotion labelling, dimensionality, representation and expectation of emotion as a latent property of interaction. Standard

B Crowd-EnVent Dataset and Appraisal Definitions

B.1 Dataset Description

The crowd-EnVENT dataset consists of 6,600 instances of emotion-inducing event descriptions. Each event is annotated using 21 appraisals as well as the stable properties of text authors (demographics, personality traits). The dataset also captures categorical emotion. The data was collected from English native speakers from diverse backgrounds, not limited to college students. The dataset is annotated and validated by external crowdworkers who read the descriptions and inferred the original appraisals.

The distribution of labels for this corpus are provided in Table 4.

Dataset Name	Type	Annotation	Layer	Domain	Size
EMPATHETICDIALOGUES	Conversations	32 emo cat	dialogue	General	24.8K
EPITOME	Reddit	3 2pt dim	response	Mental Health	10.1K
WASSA	Conversations	3 5pt dim	response	News	12.6K
Condolence	Reddit	Distress/support	comment	Online Support	14.1M
ALoE	Reddit	Empathy levels	post	Mental Health	10K
ESConv	Conversations	Support strategies	turn	Mental Health	1K
DailyDialog	Conversations	5 emo cat	turn	General	13.1K
MELD	Conversations (M.P)	3 senti + 7 emo cat	turn	Movies	13.7K
IEMOCAP	Conversations	3 5pt dim + 7 emo cat	turn	Multimodal	10K
EmoWOZ	Conversations	3 senti + 7 emo cat	turn	Task Oriented	11.4K
Twitter-Customer	Tweet-Response	3 5pt dim + 7 emo cat	turn	Customer Service	9K

Table 3: Overview of benchmark conversation corpora with emotion or affect annotations, highlighting the disparity between them. Corpora marked in **bold** are studied extensively in this paper. *cat* refers to categorical labels; *npt dim* refers to an n point dimensional Likert scale; *senti* refers to sentiment categories; *emo* refers to emotion categories. The disparity in emotion and affect annotations is apparent, depending on source and context. M.P refers to multi-party conversations. Size is measured in number of dialogues/conversations

Label	Frequency
5	1197
1	1034
4	859
2	627
3	603

Table 4: Distribution of labels from 1 to 5 in crowd-EnVENT, showing the label skew towards 1 and 5.

B.2 Appraisal Definitions

The crowd-ENVENT corpus highlights 21 appraisal dimensions, which can be categorized based on four affective state responses as established by Scherer (2005). These categories, which the paper and subsequent model treat as evaluation objectives, can be described as:

1. **Relevance**: Relevance may be determined as a combination of novelty, intrinsic pleasantness, and importance towards an experiencer's goal or objective; i.e. the relevance appraisal criterion determines the experiencer's familiarity with the event responsible for the emotion as well as linguistic cues about its alignment with the expected goals and outcomes.

2. **Implication**: Implication is seen as a combination of the causality of the agent, conduciveness of the situation towards the goal, anticipation of the consequence of the event, and the relative urgency of response to a given situation.

3. **Coping**: Coping as an appraisal objective examines how an experiencer handles the situation both in terms of their experience of control over the situation as well as the adjustment "felt necessary" by the experiencer. Some formalisms of the coping objective account for the experiencer's "power" during the experience. Troiano et al. (2023) replaces this with the dimension of 'effort' instead.

4. **Normative Significance**: The normative significance of an event or situation is the degree of conformity that the response to that situation has to personal ideals as well as with external laws or norms, which may be based on the experiencer's social or cultural environment.

These definitions are based on two critical underpinnings: that the person examining the event is also contextually involved in the event and outcome, and that this is a retrospective cognitive outcome of a given event. Given the methods adopted by Troiano et al. (2023) for curating the corpus, such an assumption is justified. However, in its applicability to dialogue, a principally reformulated set of appraisals would have to be determined. For example, the event in question could be the statement made by another conversation participant, or the scoping of *other responsibility* and *others' control* would be limited to the other conversation participant, and any individual external to the conversation be treated as a part of the "situation". However, the suitability of appraisals is beyond the scope of a feasibility study and is a promising avenue for future work given that this work establishes the noticeable enrichment to dialogue done by an appraisal based approach.

Objectives	**Dimensions**	**Definition**
Relevance	Suddenness	The situation or event was sudden or abrupt to the experiencer.
	Familiarity	The situation or event was familiar to the experiencer.
	Predictability	The experiencer could have predicted that the event would occur or the situation would arise.
	Attention	The experiencer had to pay attention to the situation.
	Not Consider	The experiencer tried to shut the situation out of their mind.
	Pleasantness	The situation or event was a pleasant experience.
	Unpleasantness	The situation or event was an unpleasant experience.
	Goal Relevance	The experiencer expected the event to have important consequences for them.
Implication	Self Responsibility	The experiencer believes that the event occurred because of their behaviour.
	Other Responsibility	The experiencer believes that the event occurred because of somebody else's behaviour.
	Situational Responsibility	The experiencer believes that the event occurred because of circumstances external to them, such as chance, special circumstances, or natural forces.
	Goal Support	The experiencer expected a positive outcome of the event for them (this is different from goal relevance as an "important" event does not necessitate the belief of a positive outcome).
	Consequence Anticipation	The experiencer anticipated the consequences of the event.
	Urgency	The experiencer believes that the event requires an immediate response.
Coping	Own Control	The experiencer believes they can influence the ongoing of the event.
	Others' Control	The experiencer believes that someone other than them was influencing the ongoing event.
	Chance Control	The experiencer believes that the situation was the result of outside influences of which nobody had control.
	Anticipated Consequence	The experiencer anticipated the outcome of the event based on their past experiences.
	Effort	The experiencer believes that the event required additional ("a great deal of") effort to deal with.
N.S.	Standards	The event clashed with the experiencer's standards or ideals.
	Social Norms	The actions that produced the event violated laws or socially accepted norms.

Table 5: With appraisal objectives defined, each appraisal dimension can be examined based on the appraisal objective they contribute to. The appraisal dimensions can be defined based on the questions asked to annotators to examine a specific situation or event. **N.S.** refers to the Normative Significance objective.

C AppraisePLM: Implementation Details

C.1 Experimental Setup

All experiments were conducted using the PyTorch deep learning framework in conjunction with the Hugging Face `transformers` library. Model training was performed on a system equipped with four NVIDIA RTX 2080 Ti GPUs, employing mixed precision training (FP16) to enhance computational efficiency and memory utilization.

C.2 Dataset and Preprocessing

Text inputs were tokenized using a maximum sequence length of either 128 or 256 tokens, depending on the specific model configuration. No additional preprocessing steps, such as lowercasing, stopword removal, or normalization, were applied.

For the appraisal prediction task, appraisal values were directly used as regression labels. In the emotion classification task, when incorporating appraisal features, these values were projected through a linear transformation to ensure dimensional compatibility before classification.

C.3 Model Architectures and Training

C.3.1 Appraisal Prediction Model

Four pretrained language models (PLMs) were utilized: RoBERTa-large, DeBERTa-large, MPNet-base, and T5-large. Each PLM was augmented with a multihead attention layer comprising 8 attention heads and 2 layers, with a hidden size equal to that of the PLM embedding layer. The output of the attention mechanism was subsequently passed through a fully connected layer for final label prediction.

Optimization was performed using the AdamW optimizer with a linear learning rate decay schedule. The models were trained using Mean Squared Error (MSE) loss with balanced class weighting. An attention weight decay of 1×10^{-3} was applied, and a dropout rate of 0.01 was employed between sequential layers, except for T5, where a dropout rate of 0.001 yielded superior performance. To mitigate exploding gradients, gradient clipping was applied after the attention layer. Training was conducted for a maximum of 10 epochs, with early stopping enforced using a patience of 3 epochs. On average, model convergence was achieved in 4.6 epochs.

C.3.2 Emotion Detection Model

Two variations of the emotion detection model were developed: a text-only model and a text + appraisal model. The text-only model followed the architecture: PLM embeddings → attention layer → classification layer. The text + appraisal model incorporated appraisal features by concatenating them with text-based embeddings after passing them through a linear projection layer to ensure dimensional alignment before classification.

For classification, cross-entropy loss with balanced class weighting was utilized. Model performance was evaluated using Precision, Recall, and F1-score.

C.4 Hyperparameter Selection

A comprehensive grid search was conducted to determine optimal values for batch size, maximum sequence length, dropout rate, and attention weight decay. The final hyperparameter selections were as follows:

- **Batch size:** 16, except for RoBERTa, where a batch size of 8 was optimal.
- **Maximum sequence length:** 128, except for RoBERTa, where a length of 256 performed best.
- **Dropout rate:** 0.01, except for T5, where 0.001 was more effective.
- **Attention weight decay:** 0.01.
- **Learning Rate:** 2e-5

All models employed a linear decay learning rate schedule, with gradient clipping applied after the attention layer to prevent gradient explosion.

C.5 Evaluation and Baselines

For appraisal prediction, model performance was assessed using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Emotion classification performance was evaluated using Precision, Recall, and F1-score.

As a baseline, our models were compared against a simple RoBERTa classifier released by the dataset authors. This baseline does not incorporate an attention mechanism and can be interpreted as an ensemble of single-task models rather than a fully integrated multitask model.

Model	Text → Appr.			Text → Emo.			Text + Appr. → Emo.		
	MSE ↓	MAE ↓	RMSE ↓	P ↑	R ↑	F1 ↑	P ↑	R ↑	F1 ↑
Troiano et. al. (2022)	1.97	3.22	1.40	0.62	0.59	0.59	0.62	0.60	0.60
RoBERTa-large	1.62	2.96	1.11	0.62	0.59	0.60	0.66	0.67	0.67
T5-large	1.64	2.77	1.12	0.63	0.61	0.61	0.63	0.65	0.66
MPNet-base	1.49	2.68	**1.08**	0.66	0.64	0.62	0.64	**0.70**	0.70
DeBERTa-large	**1.44**	**2.60**	**1.08**	0.67	0.65	0.66	**0.73**	**0.71**	**0.71**

Table 6: Performance of the AppraisePLM architecture for the appraisal regression and emotion classification models on regression. Categorical emotion detection is done in two modes; with only text (**Text → Emo.**) and both text and appraisals (**Text + Appr. → Emo.**) on the crowd-EnVent corpus. *Baseline* refers to the baseline RoBERTa-large regressor used in Troiano et. al. (2022). Per-appraisal performance and comparisons for **Text → Appr.** are provided in Appendix C.

To determine statistical significance, paired t-tests and ANOVA tests were conducted to compare model performance. These tests were performed both across different PLM architectures and before and after hyperparameter tuning. The results demonstrated statistically significant improvements in model performance following hyperparameter optimization.

To ensure the reproducibility of our results, random seeds were set for model initialization, data shuffling, and optimizer state. Additionally, all hyperparameters, training procedures, and evaluation metrics are comprehensively documented in this report. All models were trained under controlled computational conditions to facilitate consistency and comparability across experimental runs.

D AppraisePLM: Performance Analysis Details

Since different datasets had a differing number of labels and we did not employ a semantic space implementation, we perform label folding in order to evaluate the AppraisePLM model. Here, we detail the emotion mapping used. Given the label skew in the EmoWOZ and DailyDialog datasets, the emotion detection metrics were computed *excluding* the neutral emotion label.

EmpatheticDialogues (ED) Since the ED corpus has 32 fine-grained emotions to the 13 (12 without no-emotion), we had to label fold from ED into crowd-EnVENT, i.e. predictions made by AppraisePLM would be considered true positive for more than one label of the ED corpus. We folded by synonymy, where each crowd-EnVENT emotion label was mapped as follows:

We do preserve the labels for qualitative tessting, as can be seen for Figure 3.

crowd-EnVENT	EmpatheticDialogues
anger	angry, annoyed, furious, disappointed
boredom	**None**
disgust	disgusted
fear	afraid, anxious, apprehensive, terrified
guilt	guilty
joy	joyful, excited, content
no-emotion	**None**
pride	proud
relief	prepared, hopeful
sadness	sad, devastated
shame	ashamed, embarrassed
surprise	surprised
trust	trust, grateful, faithful, caring
Removed	confident, nostalgic, sentimental

Table 7: Emotion categories and their associated terms from the crowd-EnVENT to the ED corpus

DailyDialog (DD) uses Plutchik's emotion labels: anger, disgust, fear, happiness, sadness, and surprise. However, from crowd-EnVENT, it is missing the labels *boredom*, *guilt*, *shame*, *trust*, *pride*, and *relief*. Therefore, we had to label fold from DD out of crowd-EnVENT, i.e. one or more predictions made by AppraisePLM would be considered true positive for the same label of the DD corpus. We folded here by affective synonymy, where each crowd-EnVENT emotion label was mapped as follows:

crowd-EnVENT	DailyDialog
anger	anger
boredom	no emotion
disgust	disgust
fear	fear
guilt	sadness
joy	happiness
no-emotion	no emotion
pride	happiness
relief	happiness
sadness	sadness
shame	sadness
surprise	surprise
trust	happiness

Table 8: Emotion categories and their corresponding mapped categories from crowd-EnVENT to the DD corpus.

EmoWOZ EmoWOZ uses a novel emotion labelling scheme tailored to task-oriented dialogues, with seven emotion labels: Neutral, Satisfied, Dissatisfied, Excited, Apologetic, Fearful, and Abusive. Interestingly, this system is adopted from the OCC emotion annotation schema (), which has its roots in early cognitive emotion theory. In fact, appraisal dimensions could theoretically be directly mapped to certain labels. However, practically, due to the presence of an overwhelmingly large category of no emotion, and the difference in source corpus of event descriptions and rarget corpus of textual instructional conversation, we do not use a semantic space representation of the OCC model, though we leave it up to future work. Instead, we follow an OCC mapping elicited by Steunebrink, Dastani, and Meyer (2009, Figure 2) .

crowd-EnVENT	EmoWOZ
anger	dissatisfied
boredom	**None**
disgust	abusive
fear	fearful
guilt	apologetic
joy	satisfied
no-emotion	no emotion
pride	satisfied
relief	satisfied
sadness	fearful
shame	apologetic
surprise	**None**
trust	satisfied

Table 9: Emotion categories and their corresponding mapped categories from the crowd-EnVENT to the EmoWOZ corpus.

The model performed worst on this dataset, partially because of the label skew, partially because the OCC mapping from crowd-EnVENT to EmoWOZ is less than satisfactory. The goal of an appraisal-based model is to have an interpretable semantic space adaptable to the affective lexicon of a domain in order to avoid doing label mapping or using an uninterretible semantic space instead.

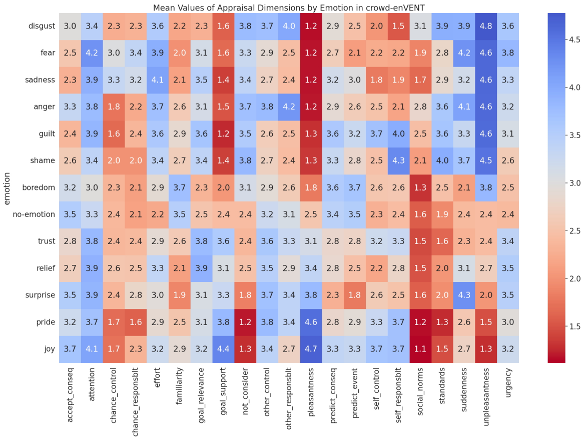


Figure 6: Average estimate of each appraisal from the crowd-enVENT test corpus. The emotion labels are ordered by *pleasantness* from low (red) to high (blue).

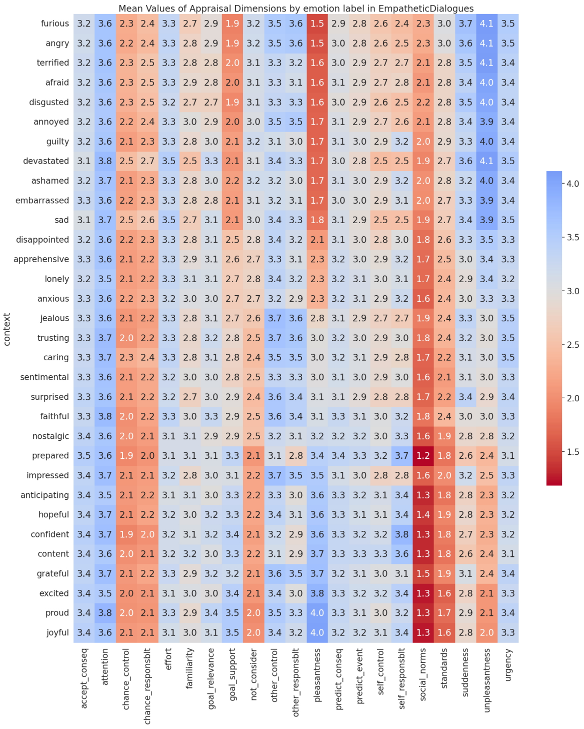
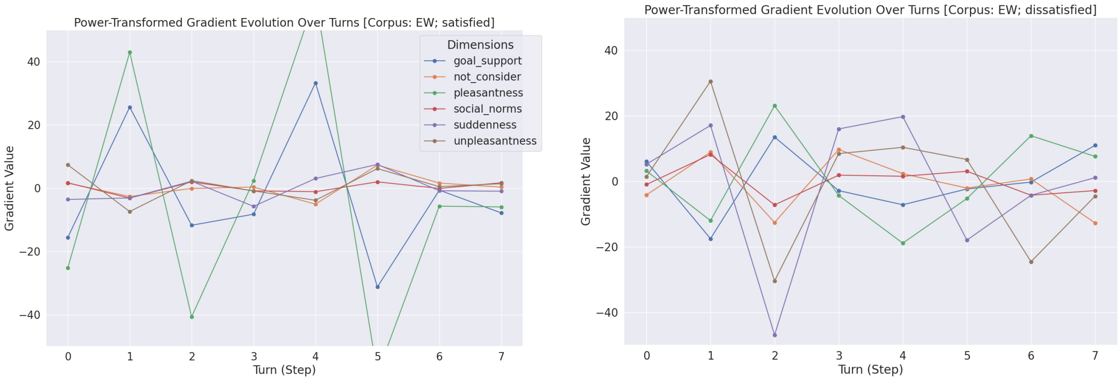


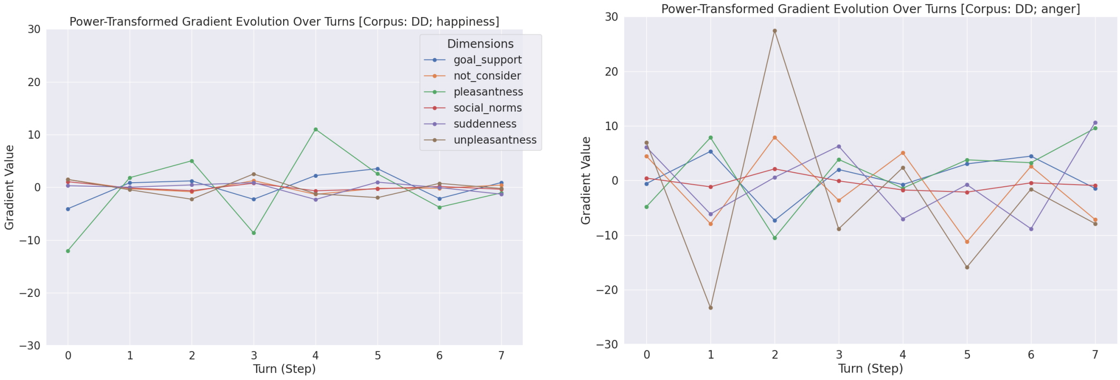
Figure 7: Average estimate of each appraisal from the EMPATHETICDIALOGUES test corpus using the best performing AppraisePLM DeBERTa-large. The emotion labels are ordered by *pleasantness* from low (red) to high (blue).

E Appraisal Distributions by Emotion Label for Conversational Corpora

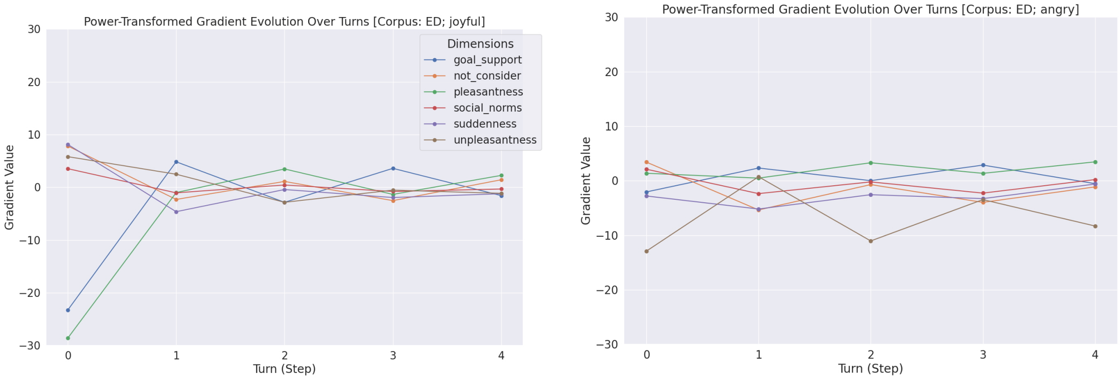
In Section §4, we presented the mean appraisal estimates of emotion in the EmoWOZ and DailyDialog dataset. Figure 6 and 7 show the distribution of appraisal values by emotion category for crowd-EnVENT and estimates for ED respectively.



(a) Dataset: EmoWOZ; Emotion Category: Satisfied (b) Dataset: EmoWOZ; Emotion Category: Dissatisfied

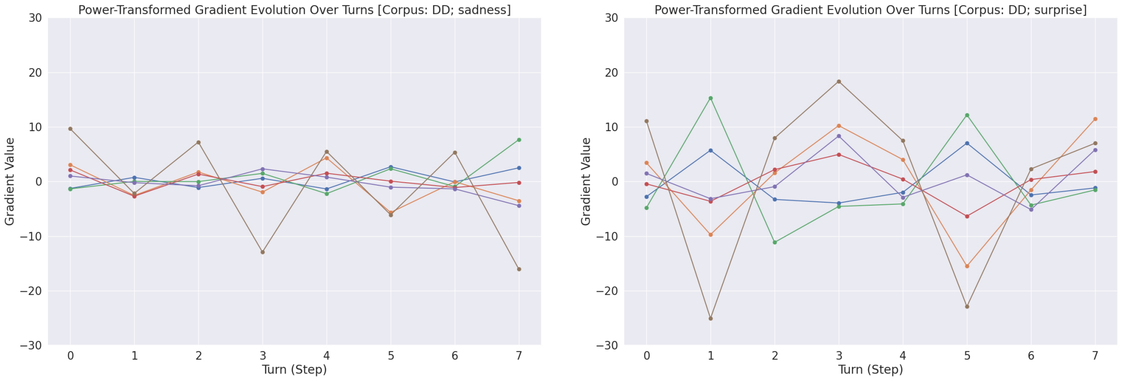


(c) Dataset: DailyDialog; Emotion Category: Happy (d) Dataset: DailyDialog; Emotion Category: Angry



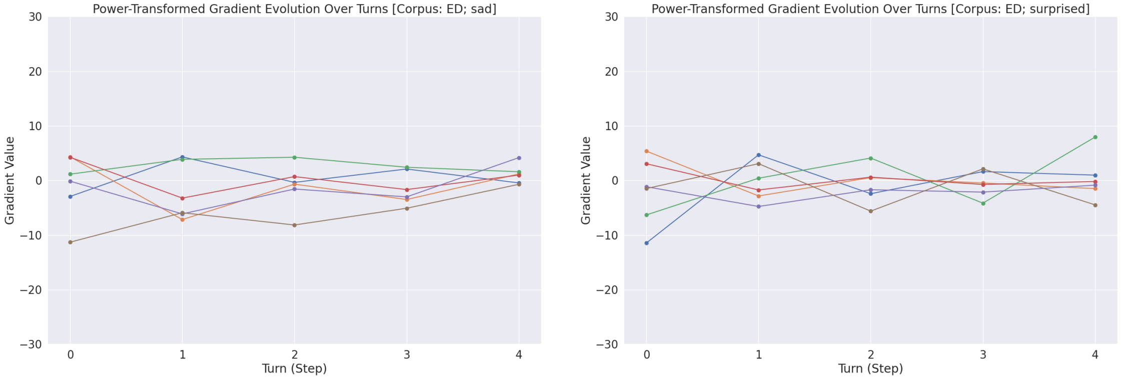
(e) Dataset: EmpatheticDialogues; Emotion Category: Joyful (f) Dataset: EmpatheticDialogues; Emotion Category: Anger

Figure 5: The average gradient of change between appraisal estimates for an average number of turns isolated by emotion category. Each turn shows the gradient, i.e. the amplified power difference between the speaker and listener across conversational turns. We see that the way corpora expect models to handle the same emotion differs greatly based on the dataset and context. The legend is shared across all graphs.



(a) Dataset: DailyDialog; Emotion Category: Sad

(b) Dataset: DailyDialog; Emotion Category: Surprise



(c) Dataset: EmpatheticDialogues; Emotion Category: Sad

(d) Dataset: EmpatheticDialogues; Emotion Category: Surprise

Figure 8: Comparing gradients from some other emotion labels in the EMPATHETICDIALOGUES and DailyDialog corpora.

Principal Parts Detection for Computational Morphology: Task, Models and Benchmark

Dorin Keshales, Omer Goldman, Reut Tsarfaty
Bar-Ilan University
`{dorinkeshales14, omer.goldman}@gmail.com, reut.tsarfaty@biu.ac.il`

Abstract

Principal parts—defined as the minimal set of cells from which all other forms within a lexeme's inflectional paradigm can be deduced—are an important concept in theoretical morphology. This concept, which outlines the minimal memorization needed for a perfect inflector, has been largely overlooked in computational morphology despite impressive advances in the field over the past decade. In this work, we formalize PRINCIPAL PARTS DETECTION as a computational task under the static scheme assumption, identifying a single set of cells as principal parts uniformly applicable across lexemes within a syntactic category. We construct a multilingual dataset of verbal inflection tables with gold principal parts annotations for ten typologically diverse languages. We evaluate several computational models for PRINCIPAL PARTS DETECTION, each implementing the same three-stage framework: characterizing relations between each pair of cells, clustering the resulting vector representations, and selecting a representative cell from each cluster as a predicted principal part. Our best-performing model, combining Edit Scripts between inflections with Hierarchical K-Means clustering, achieves an average F1 score of 55.05%, significantly outperforming a random baseline of 21.20%. While these results demonstrate initial success, further research is needed to advance PRINCIPAL PARTS DETECTION, which could optimize inputs for morphological inflection models and encourage exploration into the theoretical and practical significance of compact morphological representations.

1 Introduction

Morphological analysis is essential for understanding natural language, particularly in languages with complex inflectional systems. In both linguistic theory and language pedagogy, the concept of *principal parts* plays a central role in structuring and simplifying inflectional paradigms (Finkel and Stump, 2007; Stump and Finkel, 2013). Principal parts form the minimal subset of paradigm cells from which all other forms can be systematically derived.

By identifying these key forms, principal parts provide a compact representation of inflection tables and facilitate the analysis of morphologically rich languages. Despite their theoretical significance, the detection of principal parts remains largely unexplored in computational morphology. While they have inspired research in inflection and reinflection (Cotterell et al., 2017; Liu and Hulden, 2020), they are rarely used explicitly. Most computational approaches instead rely on a single citation form, the lemma (Cotterell et al., 2016; Goldman et al., 2023), or select input forms randomly (Cotterell et al., 2016; Kann et al., 2017). This reliance on suboptimal input representations overlooks the potential of principal parts as a more efficient foundation for inflectional modeling.

In this paper, we formalize PRINCIPAL PARTS DETECTION as a computational task under the static principal-parts scheme assumption: given a collection of inflection tables belonging to the same syntactic category, the goal is to identify a single, minimal set of cells that uniformly serve as principal parts across all lexemes. Crucially, inflection tables typically contain standard morphological annotations but are not explicitly labeled with principal parts, making this an unsupervised learning problem. To promote research in this area, we deliver a standardized dataset covering the verbal paradigms of ten diverse languages. We sourced principal parts for each language from online dictionaries, where they are often listed to aid language learners, and obtained full inflection tables from UniMorph (Batsuren et al., 2022).

We develop several computational approaches for PRINCIPAL PARTS DETECTION, leveraging the defining property of principal parts: their encapsulation of implicative relations existing among cells in the paradigm. Our models character-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 251–267
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

ize inter-cell similarity and cluster cells into *sub-paradigms*, selecting a representative cell from each sub-paradigm as predicted principal parts. We explore different methods for *characterizing* inter-cell relations, including Edit Distance, Edit Script, and Reinflection Accuracy, and we experiment with *clustering* techniques such as Affinity Propagation and a Hierarchical K-Means algorithm. Our best-performing model, combining Edit Script similarity measure + Hierarchical K-Means clustering, achieves an average F1 score of 55.05% across the ten languages in our dataset, significantly outperforming a random baseline of 21.20%.

By formalizing PRINCIPAL PARTS DETECTION as a computational task, we lay the groundwork for future research on more efficient morphological representations. To the best of our knowledge, this is the first work to deliver a standardized benchmark of PRINCIPAL PARTS DETECTION alongside a fully-operational detection framework. Successfully solving this task could enhance applications in morphological inflection and analysis by providing more informative input forms. Our findings suggest that principal parts can be computationally identified with reasonable accuracy, but further improvements are necessary to fully realize their potential.

2 The PRINCIPAL PARTS DETECTION Task and Dataset

The PRINCIPAL PARTS DETECTION Task. The task of PRINCIPAL PARTS DETECTION is defined as identifying the minimal set of cells within a paradigm that, when known, allow the derivation of all other paradigm forms. For instance, in English, the principal parts of the verbal paradigm are the cells corresponding to the infinitive, simple past and past participle (for example, *eat*, *ate*, and *eaten*), as these forms are not predictable from one another, especially for strong verbs. On the other hand, the forms corresponding to the present participle and the 3rd person singular present are deterministically predictable from the infinitive and they therefore provide no additional information for inflection if the infinitive is known.

Formally, the task of PRINCIPAL PARTS DETECTION is defined under the static principal-parts scheme assumption. Specifically, given a language L, a syntactic category POS, and their associated paradigm$P^L_{POS} = \{c_1, c_2, \ldots, c_n\}$, where each cell c_i corresponds exactly to one coherent morpho-syntactic feature set associated with POS, alongside a set of lexeme-specific inflection tables:

$$\mathcal{T}^L_{POS} = \{t^L_{P_{POS},\ell_1}, t^L_{P_{POS},\ell_2}, \ldots, t^L_{P_{POS},\ell_k}\},$$

each table instantiating the paradigm P^L_{POS} for a specific lexeme ℓ_i.

Then, the task is to identify the minimal subset of cells $C_{PP} \subseteq P^L_{POS}$ from which all remaining forms within each inflection table $t^L_{P_{POS},\ell_i} \in \mathcal{T}^L_{POS}$ can be deterministically deduced.

The PRINCIPAL PARTS DETECTION Dataset. To empirically evaluate methods for the detection of principal parts, we first need a dataset to evaluate against. To this end, we constructed the multilingual PRINCIPAL PARTS DETECTION dataset, containing verbal inflectional paradigms from ten typologically diverse languages: Hebrew, English, French, German, Spanish, Danish, Swedish, Finnish, Turkish, and Latin. These languages were selected based on the availability of comprehensive inflectional data and suitable resources for identifying principal parts.

The input side of the task comprises complete inflection tables sourced from the UniMorph corpus (Batsuren et al., 2022), a large-scale morphological resource providing comprehensive inflectional data across languages, organized by lexeme and morpho-syntactic features.

Gold principal parts annotations — the target output for evaluation — were primarily obtained from two online resources. For five languages (English, German, French, Latin, and Spanish), we directly adopted principal parts from Wikipedia's dedicated principal-parts page.[1] For the other languages, where principal parts were not explicitly documented, we identified them directly based on the forms presented in Wiktionary's standardized verb conjugation templates, except for Finnish, for which we consulted a specialized language-learning resource.[2]

The dataset preparation involved rigorous normalization and error correction applied specifically to the inflection tables. We retained only strictly inflectional forms, excluding derivational forms, and ensured exactly one form per feature set. Sparse, marginal, or inconsistent feature sets were removed, and problematic entries originating from the original sources were manually reviewed and corrected

---

[1] https://en.wikipedia.org/wiki/Principal_parts

[2] https://ielanguages.com

to ensure a reliable dataset (see Appendix A for details).

The PRINCIPAL PARTS DETECTION dataset provides a strong empirical foundation for computational modeling, bridging linguistic theory and practical applications, and constitutes a robust resource for future research on morphological inflection and principal parts detection. The next section shifts focus to computational methods for detecting principal parts, drawing on the linguistic insights outlined in the literature.[3]

3 Translating Linguistic Insights into Computational Methods

The linguistic principle underlying PRINCIPAL PARTS DETECTION is that principal parts encapsulate the implicative relations among cells within a lexeme's inflectional paradigm, constituting the minimal subset of cells from which all remaining cells can be deduced (Finkel and Stump, 2007; Stump and Finkel, 2013). In this work, we adopt the static principal-parts scheme, framing PRINCIPAL PARTS DETECTION as the automatic identification of a minimal, uniform subset of paradigm cells applicable consistently across all lexemes within a given syntactic category.

However, linguistic theory alone does not provide a direct computational method for systematically generalizing or approximating these implicative relations across multiple lexemes at the syntactic-category level. To operationalize PRINCIPAL PARTS DETECTION computationally, we hypothesize that implicative relationships across lexemes can be indirectly approximated through measurable morphological patterns observable within lexemes' inflectional paradigms. Specifically, we propose three types of measurable morphological patterns: (i) surface-level similarities, observed as orthographic overlap, shared morphological markers, or recurring affixation patterns across multiple cells; (ii) structural correspondences, represented by minimal transformations converting one cell's form into another; and (iii) predictive relations, characterized by the consistent ability of one cell's realization to predict another's.

By quantifying the morphological relations among cells based on these measurable patterns, we obtain empirical evidence enabling the organization of cells into meaningful subsets. We introduce the notion of sub-paradigms, computational abstractions (not formally defined in linguistic theory) that group cells whose realizations consistently display morphological and functional similarities. Cells grouped into a sub-paradigm thus implicitly share similar morphological and functional roles across lexemes, indirectly reflecting broader implicative trends, even though exact implicative relationships vary between individual lexemes.

Selecting principal parts thus naturally corresponds to choosing exactly one representative cell from each identified sub-paradigm. This ensures the resulting principal-part set compactly and effectively captures the generalized morphological roles identified through sub-paradigm formation.

This conceptualization leads directly to a three-phase computational methodology for PRINCIPAL PARTS DETECTION: First, we *characterize* morphological relationships between pairs of cells through similarity measures. Next, we *cluster* these cells into coherent sub-paradigms based on their quantified morphological similarities. Finally, we *select* exactly one representative cell from each sub-paradigm as its principal part. Together, these principal parts constitute a minimal and comprehensive set capable of systematically deriving all remaining paradigm cells across lexemes.

4 Framework and Task Empirical Design

The PRINCIPAL PARTS DETECTION framework we propose in this work is composed of three interconnected stages: *characterization*, *clustering*, and *principal-parts selection*, each implemented using well-defined computational methods. These stages operate independently, meaning that different configurations of the framework can mix and match methods in seeking the best combination. Below, we briefly review the computational methods considered for each stage.

4.1 Characterization: Quantifying Morphological Relations Between Cells

The characterization stage quantifies morphological relationships between paradigm cells by computing numerical similarity scores between them. This work explores three distinct characterization methods, each offering a different perspective on morphological relations between cells.

Edit Distance A metric that measures surface-level similarity between forms based on minimal

---

[3] The data is publicly available in `https://github.com/DorinK/Principal-Parts-Detection`.

edit operations — insertions, deletions, or substitutions — required to transform one form into another (Levenshtein, 1966). This method is implemented by computing the average Edit Distance from each paradigm cell to all others (calculated across all lexemes in the data), treating one as the source and the rest as destinations. The resulting vector representations store these averaged distances, capturing surface-level similarity between cells. Pairs of paradigm cells with low Edit Distance scores exhibit orthographic overlap.

Edit Script A metric that captures transformational diversity by analyzing character-level transformations between paradigm cells. Unlike traditional Edit Script approaches (Wagner and Fischer, 1974; Myers, 1986), which focus on the exact sequence of operations needed to transform one string into another, this approach computes the number of unique character-level transformations observed across all surface realizations of each paradigm-cell pair. Each transformation is counted only once per cell pair (calculated across all lexemes), capturing distinct transformational patterns rather than repeatedly occurring character changes. The result is a vector representation for each cell pair, where each entry encodes the number of unique transformations required to convert one cell to another, representing their transformational distance. This method provides insight into the variation in morphological transformations within a paradigm. Cells with lower transformation diversity may exhibit more stable morphological patterns, making them stronger principal part candidates. In contrast, higher transformation diversity may signal greater variability in inflectional behavior, affecting predictability.

Reinflection Accuracy A metric that evaluates the functional predictability of paradigm cells. It leverages the Base LSTM reinflection model (Goldman et al., 2021) trained to generate a target form given a source form and the morpho-syntactic features of the target. Unlike edit-based methods that focus on surface similarity and transformational diversity, Reinflection Accuracy captures functional dependencies between cells, reflecting their predictive capacity within a paradigm.

Reinflection Accuracy is particularly effective in languages with complex inflectional systems, where orthographic similarity alone is not a reliable predictor of implicative relations. By capturing functional dependencies rather than surface transformations, it provides a direct measure of a cell's ability to generate other forms. However, its performance depends on training data quality and resource availability. In low-resource settings, data sparsity may lead to biased results, and the approach is computationally intensive, as it requires training multiple models—one model per paradigm cell. Despite these challenges, its ability to model functional predictability makes it a valuable tool for identifying paradigm cells suitable as principal parts, particularly in morphologically complex languages.

Each characterization method produces a similarity table, where rows represent source paradigm cells and columns represent target paradigm cells, encoding pairwise morphological relationships (see Appendix B). Before clustering, all similarity tables are standardized by removing the mean and scaling to unit variance to ensure comparability across methods. These standardized characterization tables form the empirical basis for the clustering stage.

4.2 Clustering: Structuring Cells into Sub-Paradigms

The clustering stage groups paradigm cells based on their quantified morphological relationships, forming computational abstractions termed sub-paradigms. These sub-paradigms approximate the internal morphological organization of paradigms. The framework implements two clustering algorithms, each offering distinct computational properties. As with characterization, only one clustering algorithm is employed at a time.

Affinity Propagation A message-passing clustering algorithm that dynamically determines the number of clusters based on pairwise similarity scores (Frey and Dueck, 2007). Unlike traditional clustering methods, it does not require a predefined number of clusters. Instead, it iteratively updates responsibility and availability values, which determine how well a paradigm cell serves as an exemplar (cluster center), until the algorithm converges on a final set of exemplars. This property makes it particularly suitable for paradigms with high morphological variability. The algorithm is implemented using scikit-learn's AffinityPropagation module, with similarity scores computed as negative squared Euclidean distances. The preference parameter is set to the median similarity value, allowing clusters to emerge naturally. Additional parameters include a convergence iteration limit of 30 and a random state value of 10.

Hierarchical K-Means A hierarchical variant of K-Means that recursively partitions paradigm cells into two clusters per iteration until a well-defined clustering structure is reached. The stopping criterion is determined using the Calinski–Harabasz Index (CHI) (Caliński and Harabasz, 1974), which evaluates clustering quality by comparing between-cluster dispersion to within-cluster cohesion. At each step, the CHI is computed across the entire clustering structure to assess how well-separated the clusters are relative to their internal cohesion. To prevent over-segmentation, clustering stops if the number of clusters in the new best CHI solution exceeds that of the previous best CHI solution by more than one cluster. The algorithm is implemented using scikit-learn's KMeans module with a random state value of 10. By grouping paradigm cells into sub-paradigms, the clustering stage provides a data-driven approximation of generalized morphological and functional roles. The resulting sub-paradigms form the structured basis for the principal-parts selection stage.

4.3 Principal Parts Selection: Identifying Representative Cells

The principal-parts selection stage finalizes the PRINCIPAL PARTS DETECTION framework by transforming sub-paradigms into a compact, generative summary of the paradigm structure. In this stage, exactly one representative cell from each sub-paradigm is selected, capturing the morphological and functional properties that characterize its sub-paradigm. These representative cells collectively constitute the principal parts, ensuring comprehensive morphological coverage while maintaining compactness and predictive capacity.

Specifically, we adopt the *Minimum Average Inflectional Length* criterion. Under this criterion, the principal part selected from each sub-paradigm is the paradigm cell whose realizations exhibit the minimal average length, computed across all lexemes. This selection ensures that the chosen cell is both structurally central and morphologically efficient within its sub-paradigm. Such a criterion aligns with linguistic insights suggesting that shorter inflectional paths often correspond to central morphological roles, enhancing their suitability as principal parts.

Together, the principal parts derived from this selection process form a minimal and comprehensive set capable of systematically deriving all remaining paradigm cells across lexemes, in accordance with the static principal-parts scheme assumption.

5 Experimental Setup and Results

We conduct a series of experiments to evaluate the effectiveness of the PRINCIPAL PARTS DETECTION framework across ten typologically diverse languages. The evaluation compares six model configurations, each formed by pairing one of three characterization methods—Edit Distance, Edit Script, and Reinflection Accuracy—with one of two clustering algorithms—Affinity Propagation and Hierarchical K-Means. To establish a performance threshold, we include a random baseline, selecting principal parts at random.

5.1 Dataset

The PRINCIPAL PARTS DETECTION dataset utilized in our experiments comprises ten typologically diverse languages, structured into two subsets to rigorously assess our framework's cross-linguistic generalization.

The first subset (Hebrew, English, French, German, Spanish) was used during method development, providing a broad and representative morphological foundation. Hebrew exhibits synthetic morphology, encoding multiple grammatical features within single inflected forms. English, in contrast, is predominantly analytic, relying primarily on word order and function words to indicate grammatical relations. French and Spanish, as fusional languages, embed tense, mood, and person distinctions within single inflectional forms, exhibiting varying degrees of morphological regularity. Finally, German presents a hybrid morphological system, integrating analytic and fusional characteristics.

The second subset (Danish, Swedish, Finnish, Turkish, Latin) was reserved exclusively for independent validation of the finalized methods' generalizability, evaluating their performance on languages not encountered during development. Finnish and Turkish exemplify agglutinative morphology, expressing grammatical information through clearly segmentable morphemes arranged sequentially. Latin, a highly inflected classical language, provides a challenging scenario due to extensive distinctions in case, number, and gender. Danish and Swedish, characterized by regular and predictable inflectional paradigms, allow us to assess method robustness in languages with simpler morphological structures.

This structuring enables a rigorous and unbiased evaluation of our framework's adaptability, robustness, and cross-linguistic generalization across diverse morphological systems.

5.2 Evaluation Metric

To evaluate model effectiveness, we utilize the F1 score, balancing precision (correctness of predicted principal parts) and recall (coverage of gold principal parts) to assess both accuracy and completeness in PRINCIPAL PARTS DETECTION.

In addition to reporting F1 scores, we benchmark our models against a random baseline, which selects principal parts randomly within each paradigm. Given a paradigm with x cells and y gold principal parts, the probability of randomly selecting a correct principal part is $\frac{y}{x}$. As the baseline selects exactly y principal parts, the expected number of correct predictions is $y \times \frac{y}{x} = \frac{y^2}{x}$. Thus, the expected precision, recall, and consequently the F1 score, are all equal to $\frac{y}{x}$.

Since principal parts are inherently sparse within most paradigms, the random baseline represents a challenging threshold. Models that significantly exceed this baseline demonstrate an ability to detect principal parts systematically rather than relying on chance.

5.3 Reinflection Settings

For models utilizing Reinflection Accuracy, we train a separate reinflection model for each paradigm cell, treating it as the source while all other cells serve as targets. The model is based on the Base LSTM architecture (Goldman et al., 2021), a character-based sequence-to-sequence model comprising a one-layer bidirectional LSTM encoder and a one-layer unidirectional LSTM decoder with a global soft attention layer (Bahdanau et al., 2014). Each model is trained for 50 epochs, optimizing categorical cross-entropy.

The dataset is split 70%-30%, ensuring test lexemes remain unseen during training. Each paradigm cell is trained using a dedicated dataset, where it serves as the source inflection across different lexemes. Since each cell is evaluated on its ability to generate all other cells within the paradigm, corresponding test sets are created—one per target cell.

Model	**Algorithmic Evaluation**
Random Baseline	21.20
Edit Distance + Affinity Propagation	31.29
Edit Distance + Hierarchical K-Means	32.51
Reinflection Accuracy + Hierarchical K-Means	42.43
Edit Script + Affinity Propagation	44.62
Reinflection Accuracy + Affinity Propagation	45.56
Edit Script + Hierarchical K-Means	**55.05**

Table 1: Average F1 scores across the ten languages of our PRINCIPAL PARTS DETECTION dataset for different model configurations. The best-performing model configuration is highlighted.

Each trained model is evaluated on how accurately it inflects from its assigned source cell to each target cell. The resulting accuracy scores form a representation vector, capturing a cell's proficiency in generating others. Cells with high Reinflection Accuracy scores demonstrate strong predictive capacity, making them effective candidates for principal parts.

5.4 Results

Table 1 presents the average F1 scores across the ten languages, providing a comparative evaluation of model performance. All models outperform the random baseline, which achieves the lowest F1 score of 21.20%. The best-performing model, Edit Script + Hierarchical K-Means, achieves an F1 score of 55.05%, highlighting its ability to effectively characterize morphological relationships among paradigm cells and cluster these cells across diverse languages.

Reinflection Accuracy-based models perform competitively, with F1 scores of 45.56% (Affinity Propagation) and 42.43% (Hierarchical K-Means). In contrast, Edit Distance-based models yield lower scores of 31.29% and 32.51%, indicating that surface-level similarity alone is insufficient for PRINCIPAL PARTS DETECTION.

Overall, all tested methods surpass the random baseline by at least 10.09 points, with the best-performing model exceeding it by 33.85 points. These results confirm the effectiveness of the proposed methodology, highlighting a substantial improvement over random selection.

Table 2 provides a language-specific breakdown of F1 scores, offering further insights into models' performance across morphological typologies. Edit Script + Hierarchical K-Means, our best-performing model overall, achieves the highest scores in Hebrew, French, Spanish, Turkish, and Latin. This highlights its effectiveness in capturing systematic morphological transformations—particularly beneficial in languages with root-and-pattern morphology (e.g., Hebrew), fusional systems (e.g., French, Spanish, Latin), where single inflections encode multiple grammatical features simultaneously, and in Turkish, an agglutinative language characterized by clearly segmentable, predictable morphological sequences.

While the Reinflection Accuracy + Affinity Propagation model ranks second-best overall (45.56%), it does not consistently outperform other models across languages. Its strongest results appear specifically in languages characterized by relatively transparent, regular, and predictable inflectional paradigms, such as Danish and Swedish, where the exemplar-based clustering method effectively organizes paradigm cells. Conversely, its performance drops in morphologically opaque or fusional languages (e.g., Spanish, Finnish). However, the Reinflection Accuracy + Hierarchical K-Means model achieves notably stronger results in Finnish and English, indicating differences in how clustering methods handle morphological predictability. These contrasting patterns underscore the importance of carefully matching characterization methods and clustering algorithms to linguistic properties.

In contrast to the previously discussed models, the weaker performance of Edit Distance-based models is particularly evident in morphologically opaque or highly fusional languages (e.g., Spanish, Finnish), where subtle or irregular morphological variations encode multiple grammatical features simultaneously.

6 Analysis

We analyze how methodological factors shape model performance, focusing on transformations in characterization data and the effectiveness of clustering strategies. This evaluation highlights structural patterns influencing clustering quality and examines the extent to which clustering results align with ideal principal-parts selection.

6.1 Transpose Ablation: Evaluating the Impact of Data Orientation

The Transpose Ablation study investigates whether swapping the rows and columns of the characterization tables influences clustering quality and principal-parts selection. This transformation is particularly relevant for Reinflection Accuracy, where original tables encode directional relationships—rows indicate how easily a paradigm cell can inflect from itself to others, while columns represent the reverse relationship. Unlike Edit Distance and Edit Script methods, which produce symmetric similarity matrices, Reinflection Accuracy matrices are inherently asymmetric. Thus, transposing these tables meaningfully changes their directional structure and potentially impacts clustering results.

Transposition is applied only to Reinflection Accuracy models, as Edit Distance and Edit Script methods generate symmetric similarity tables, making transposition redundant. We evaluate two models: Reinflection Accuracy + Affinity Propagation and Reinflection Accuracy + Hierarchical K-Means, comparing their performance before and after transposition.

The results in Table 3 show that transposition affects models differently. Reinflection Accuracy + Affinity Propagation experiences a slight decrease in performance (45.56% $\rightarrow$ 44.05%), while Reinflection Accuracy + Hierarchical K-Means improves marginally (42.43% $\rightarrow$ 43.14%). This suggests that transposition does not universally enhance clustering effectiveness and that its impact depends on the underlying clustering strategy.

Despite the minor improvement for Hierarchical K-Means, transposed results are excluded from the main evaluation due to their limited effect and misalignment with the principal-parts definition. Because original (non-transposed) cells encode generative properties crucial for inflection, preserving this structure remains preferable. These findings suggest that alternative data transformations, better aligned with the linguistic task, may offer greater benefits.

6.2 Oracle Evaluation

To estimate the theoretical upper bound of our models' performance, we conduct an Oracle Evaluation, where principal parts are selected directly from the gold principal parts annotations rather than relying on clustering results. This evaluation disentangles the contribution of clustering quality from principal-parts selection effectiveness: a low Oracle score indicates fundamental limitations in clustering, while a significant gap between Oracle and Algorithmic scores highlights inefficiencies specifically in the principal-parts selection stage. By providing this performance ceiling, the Ora-

Model	Hebrew	English	French	German	Spanish	Danish	Swedish	Finnish	Turkish	Latin
Random Baseline	20.68	60.00	14.28	16.66	2.53	62.50	26.30	2.48	28.00	6.25
Edit Distance + Affinity Propagation	33.30	66.70	37.50	46.20	15.40	57.10	40.00	0.00	0.00	16.70
Edit Distance + Hierarchical K-Means	25.00	57.10	44.40	44.40	0.00	57.10	57.10	0.00	0.00	40.00
Reinflection Accuracy + Hierarchical K-Means	25.00	85.70	44.40	28.60	50.00	57.10	43.50	50.00	0.00	40.00
Edit Script + Affinity Propagation	50.00	80.00	54.50	66.70	36.40	50.00	60.00	23.50	6.90	18.20
Reinflection Accuracy + Affinity Propagation	36.40	80.00	26.70	60.00	16.70	75.00	75.00	46.20	17.40	22.20
Edit Script + Hierarchical K-Means	**50.00**	80.00	**54.50**	60.00	**50.00**	72.70	60.00	33.30	**50.00**	**40.00**

Table 2: Language-specific F1 scores illustrating variations in effectiveness of different model configurations across morphological typologies. Top results are marked, with a unique color used for each language.

Model	Transpose	Algorithmic Evaluation
Reinflection Accuracy + Affinity Propagation	✗	**45.56**
	✓	44.05
Reinflection Accuracy + Hierarchical K-Means	✗	42.43
	✓	**43.14**

Table 3: Algorithmic evaluation of Reinflection Accuracy models with and without transposition across ten languages. The averaged F1 scores highlight varying impacts depending on the clustering algorithm.

Model	Evaluation	
	Oracle	Algorithmic
Edit Distance + Affinity Propagation	40.08	31.29
Edit Distance + Hierarchical K-Means	50.57	32.51
Reinflection Accuracy + Affinity Propagation	58.78	45.56
Reinflection Accuracy + Hierarchical K-Means	65.64	42.43
Edit Script + Affinity Propagation	54.16	44.62
Edit Script + Hierarchical K-Means	**76.21**	**55.05**

Table 4: Oracle and Algorithmic evaluations of PRINCIPAL PARTS DETECTION models across languages. Oracle evaluation assumes perfect knowledge of principal parts, establishing an upper bound on performance; Algorithmic evaluation reflects actual model performance.

Model	Transpose	Evaluation	
		Oracle	Algorithmic
Reinflection Accuracy + Affinity Propagation	✗	**58.78**	**45.56**
	✓	58.51	44.05
Reinflection Accuracy + Hierarchical K-Means	✗	65.64	42.43
	✓	**67.70**	**43.14**

Table 5: Oracle and Algorithmic evaluations of Reinflection Accuracy models before and after transposition, assessing clustering quality under ideal (Oracle) and practical (algorithmic) conditions.

cle Evaluation identifies which components of the PRINCIPAL PARTS DETECTION framework require targeted improvement.

Table 4 reveals substantial gaps between Oracle and Algorithmic scores, underscoring clustering limitations and principal-parts selection inefficiencies. Edit Script + Hierarchical K-Means achieves the highest Oracle score (76.21%), confirming strong clustering performance. However, the 21.16-point gap suggests that principal-parts selection remains a limiting factor.

Conversely, Edit Distance + Affinity Propagation exhibits the lowest Oracle score (40.08%), indicating fundamental clustering challenges. Reinflection Accuracy + Hierarchical K-Means shows a notably large Oracle-Algorithmic gap (65.64% → 42.43%), highlighting that while clustering is effective, principal-parts selection still requires refinement.

These findings emphasize the importance of optimizing both clustering effectiveness and principal-parts selection to bridge the gap between Oracle and Algorithmic performance.

6.3 Interplay Between Transposition and Oracle Performance

Table 5 examines the impact of transposition on Reinflection Accuracy models under both Oracle and Algorithmic evaluations.

The results indicate that while transposition improves Oracle performance for Hierarchical K-Means (65.64% → 67.70%), it has a negligible effect on Algorithmic scores, indicating that while transposition enhances clustering under ideal conditions, it does not meaningfully improve principal-parts selection. Additionally, Affinity Propagation exhibits sensitivity to data orientation, showing a slight decline in Oracle performance (58.78% → 58.51%), suggesting that its clustering mechanism relies on specific directional patterns that transposition may disrupt. Conversely, Hierarchical K-Means benefits from transposed data, likely due to its iterative refinement of clusters. However, since Algorithmic scores remain largely unchanged across models, these findings reinforce that refining selection heuristics, rather than adjusting data orientation, is the key to improving model performance.

7 Related Work

Early computational approaches to paradigm completion predominantly relied on the lemma as the central reference form, treating it as the sole input for generating full inflectional paradigms (Durrett and DeNero, 2013; Hulden, 2014; Nicolai et al., 2015; Ahlberg et al., 2015; Faruqui et al., 2016). However, Cotterell et al. (2017) highlighted the limitations of this approach, noting that forcing transformations to pass exclusively through the lemma can introduce unnecessary complexity. Instead, more flexible models leveraging multiple inflected forms have been proposed, allowing transformations to occur directly or via intermediary forms, rather than constraining them to a single privileged form. This shift aligns with the concept of principal parts, defined as the minimal set of paradigm cells required to deduce all others (Finkel and Stump, 2007; Stump and Finkel, 2013).

Cotterell et al. (2017) introduced a directed graphical model that probabilistically generates missing inflected forms by modeling dependencies within paradigms. This approach enables the prediction of a form from multiple inflected forms rather than exclusively from the lemma. Around the same time, Kann et al. (2017) introduced multi-source reinflection, demonstrating that using multiple inflected forms as input improves accuracy. Their work explicitly references principal parts as a linguistic motivation, reinforcing the idea that certain cells within a paradigm hold stronger predictive capacity. Additionally, Cotterell et al. (2019) examined the structural complexity of inflectional paradigms, proposing a neural method for ordering paradigm slots based on their predictability—an indirect computational realization of the principal parts concept.

Liu and Hulden (2020) extended these ideas by reformulating morphological inflection as a Paradigm Cell Filling Problem (PCFP), where missing forms are inferred from a partially observed set of paradigm cells. While their work does not explicitly model principal parts, it aligns with their predictive role in improving inflectional accuracy, particularly in low-resource settings.

Despite these advancements, no prior work has proposed a systematic, data-driven approach to PRINCIPAL PARTS DETECTION. Existing studies have either assumed pre-defined principal parts or incorporated them indirectly within broader inflectional tasks. In contrast, we have introduced PRINCIPAL PARTS DETECTION as a formal computational task, developed a multilingual benchmark, and proposed a principled methodology for automatic PRINCIPAL PARTS DETECTION. By integrating linguistic insights with computational modeling, we establish a structured framework for PRINCIPAL PARTS DETECTION.

8 Conclusions

This work introduces PRINCIPAL PARTS DETECTION as a computational task, formalizing the detection of principal parts within inflectional paradigms under the static principal-parts scheme assumption. We construct a multilingual dataset covering ten typologically diverse languages and develop a structured framework to automatically detect principal parts uniformly applicable across all lexemes belonging the verb syntactic category.

Our empirical evaluation demonstrates that quantifying morphological relationships between cells, clustering these cells into sub-paradigms, and selecting representative cells from each sub-paradigm provide a viable strategy for identifying principal parts. Our best-performing approach — Edit Script similarity combined with Hierarchical K-Means clustering — achieves an F1 score of 55.05%, significantly surpassing the random baseline of 21.20%. However, results across evaluated models indicate that while clustering effectively organizes paradigm cells into meaningful subsets, principal-parts selection remains a key bottleneck.

Beyond theoretical interest, successfully addressing PRINCIPAL PARTS DETECTION has practical implications for computational morphology. By identifying compact, generative subsets of paradigm cells, principal parts can be leveraged to optimize morphological inflection models, reduce annotation costs, and improve low-resource language modeling. The structured computational approach presented here lays the foundation for future advancements, highlighting the relevance of linguistic insights in shaping more efficient NLP methodologies.

Limitations

Despite the progress demonstrated in this study, several open challenges remain. Irregular paradigms, as seen in Latin, continue to pose difficulties, highlighting the need for methods that can better capture morphological unpredictability. Additionally, our reliance on UniMorph, while offering broad linguistic coverage, exposes inconsistencies that impact model generalization. More curated linguistic resources could improve dataset reliability and refine the evaluation of principal parts across languages.

Future work could explore alternative clustering strategies better suited to morphological structures, such as graph-based methods or neural clustering approaches. Transformer-based models may hold potential for capturing deeper morphological dependencies, offering an avenue for enhancing both clustering accuracy and principal-parts selection. These challenges are beyond the scope of this paper and reserved for future research.

Finally, our dataset currently includes only ten languages. Expanding the dataset to include additional morphologically rich and underrepresented languages, such as polysynthetic languages, would more comprehensively capture typological diversity and potentially further validate the robustness of PRINCIPAL PARTS DETECTION methods.

Acknowledgments

This project is funded by a generous grant from the Israeli Science Foundation (#ISF-670/23), and a KAMIN grant (#81698) from the Israeli Innovation Authority (IAA), for which we are grateful. In addition, the first author is funded by a VATAT grant, and the second author is partially funded by the UniDive COST Action (#CA21167), which we thankfully acknowledge. We finally with to thank three anonymous reviewers for their insightful comments on an earlier draft.

References

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, Jaime Rafael Montoya Samame, Delio Siticonatzi Camaiteri, Esaú Zumaeta Rojas, Didier López Francis, Arturo Oncevay, Juan López Bautista, Gema Celeste Silva Villegas, Lucas Torroba Hennigen, Adam Ek, David Guriel, Peter Dirix, Jean-Philippe Bernardy, Andrey Scherbakov, Aziyana Bayyr-ool, Antonios Anastasopoulos, Roberto Zariquiey, Karina Sheifer, Sofya Ganieva, Hilaria Cruz, Ritván Karahóǧa, Stella Markantonatou, George Pavlidis, Matvey Plugaryov, Elena Klyachko, Ali Salehi, Candy Angulo, Jatayu Baxi, Andrew Krizhanovsky, Natalia Krizhanovskaya, Elizabeth Salesky, Clara Vania, Sardana Ivanova, Jennifer White, Rowan Hall Maudslay, Josef Valvoda, Ran Zmigrod, Paula Czarnowska, Irene Nikkarinen, Aelita Salchak, Brijesh Bhatt, Christopher Straughn, Zoey Liu, Jonathan North Washington, Yuval Pinter, Duygu Ataman, Marcin Wolinski, Totok Suhardijanto, Anna Yablonskaya, Niklas Stoehr, Hossep Dolatian, Zahroh Nuriah, Shyam Ratan, Francis M. Tyers, Edoardo M. Ponti, Grant Aiton, Aryaman Arora, Richard J. Hatcher, Ritesh Kumar, Jeremiah Young, Daria Rodionova, Anastasia Yemelina, Taras Andrushko, Igor Marchenko, Polina Mashkovtseva, Alexandra Serova, Emily Prud'hommeaux, Maria Nepomniashchaya, Fausto Giunchiglia, Eleanor Chodroff, Mans Hulden, Miikka Silfverberg, Arya D. McCarthy, David Yarowsky, Ryan Cotterell, Reut Tsarfaty, and Ekaterina Vylomova. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Tadeusz Caliński and Jerzy Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1):1–27.

Ryan Cotterell, Christo Kirov, Mans Hulden, and Jason Eisner. 2019. On the complexity and typology of inflectional morphological systems. *Transactions of the Association for Computational Linguistics*, 7:327–342.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The sigmorphon 2016 shared task—morphological reinflection. In *Proceedings of the 14th SIGMORPHON workshop on computational research in phonetics, phonology, and morphology*, pages 10–22.

Ryan Cotterell, John Sylak-Glassman, and Christo Kirov. 2017. Neural graphical models over strings for principal parts morphological paradigm completion. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 759–765, Valencia, Spain. Association for Computational Linguistics.

Greg Durrett and John DeNero. 2013. Supervised learning of complete morphological paradigms. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1185–1195, Atlanta, Georgia. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Graham Neubig, and Chris Dyer. 2016. Morphological inflection generation using character sequence to sequence learning. *arXiv preprint arXiv:1512.06110*.

Raphael Finkel and Gregory Stump. 2007. Principal parts and morphological typology. *Morphology*, 17:39–75.

Brendan J. Frey and Dmitri Dueck. 2007. Clustering by passing messages between data points. *Science*, 315(5814):972–976.

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, Toronto, Canada. Association for Computational Linguistics.

Omer Goldman, David Guriel, and Reut Tsarfaty. 2021. (un) solving morphological inflection: Lemma overlap artificially inflates models' performance. *arXiv preprint arXiv:2108.05682*.

Mans Hulden. 2014. Generalizing inflection tables into paradigms with finite state operations. In *Proceedings of the 2014 Joint Meeting of SIGMORPHON and SIGFSM*, pages 29–36, Baltimore, Maryland. Association for Computational Linguistics.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2017. Neural multi-source morphological reinflection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 514–524, Valencia, Spain. Association for Computational Linguistics.

Vladimir I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10(8):707–710.

Ling Liu and Mans Hulden. 2020. Leveraging principal parts for morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 153–161.

Eugene W Myers. 1986. An o (nd) difference algorithm and its variations. *Algorithmica*, 1(1):251–266.

Garrett Nicolai, Colin Cherry, and Grzegorz Kondrak. 2015. Inflection generation as discriminative string transduction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 922–931, Denver, Colorado. Association for Computational Linguistics.

Gregory Stump and Raphael A. Finkel. 2013. *Principal parts*, Cambridge Studies in Linguistics, page 9–39. Cambridge University Press.

Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)*, 21(1):168–173.

Appendix

A Technical Overview of the PRINCIPAL PARTS DETECTION Dataset

This section provides the technical details of the PRINCIPAL PARTS DETECTION dataset, including the number of samples per feature set in each language's verb paradigm and the total number of gold principal parts for each language. In some cases, specific feature sets were removed for various reasons, as explained in subsection A.2.

Additionally, we list the gold principal parts for each language, formatted as `feature_set (e.g., form)`. When two feature sets share the same form, the gold principal parts are listed in square brackets `[ ]`. The first feature set corresponds to the principal part identified in linguistic literature, while the second represents a feature set that consistently shares the same form across all samples in the dataset. In such cases, the second feature set is included as a possible principal part, as the algorithm's choice between them does not affect the analysis. To avoid redundancy, no principal part is counted more than once in these scenarios.

A.1 Dataset Summary and Illustrative Lexeme Examples

For each language, we provide an example lexeme to illustrate the principal parts, formatted as `feature_set (e.g., form)`. These examples are illustrative and may not share the same meanings across languages.

A.2 Explanatory Notes

The following explanatory notes clarify decisions made during dataset preparation and supplement the information presented in Table 6:

- **Spanish**: PRO feature sets, representing verbs with object clitic pronouns, were removed.
- **Swedish**: The V-IMP-PASS feature set was excluded due to insufficient samples (only three).
- **Latin**:
  - Passive feature sets were excluded.
  - Feature sets starting with V.PTCP (instead of V-V.PTCP) were removed.
  - Feature sets with 30 or fewer samples were excluded.
  - The first-person-singular-perfect-active-indicative feature set was excluded from the gold principal parts list due to insufficient data (only two samples).

B Characterization Tables for Selected Languages

To illustrate the structure of the characterization methods, we present detailed characterization tables for three representative languages from our dataset. These tables demonstrate how different feature sets relate within the verb morphology of each language, showcasing the variation across Edit Distance, Edit Script, and Reinflection Accuracy characterization methods.

Each language is represented by three tables, corresponding to the distinct characterization methods, with principal parts highlighted in yellow for clarity. Additionally, cases where two feature sets consistently share the same form and are interchangeable as principal parts are marked with a distinct color. Since these feature sets carry identical information, the model's selection between them does not impact the results.

Interpretation of Tables. The provided tables exemplify the structure of the characterization methods rather than an exhaustive display of all ten languages in our study. While specific lexeme examples are shown in the rows and columns, the quantified relationships they capture apply to the entire verb morphology of each language. These examples serve to illustrate the broader implicative patterns identified during the characterization process.

B.1 Characterization Tables for English

Figures 1, 2, and 3 illustrate the Edit Distance, Edit Script, and Reinflection Accuracy characterization tables for English, respectively.

B.2 Characterization Tables for German

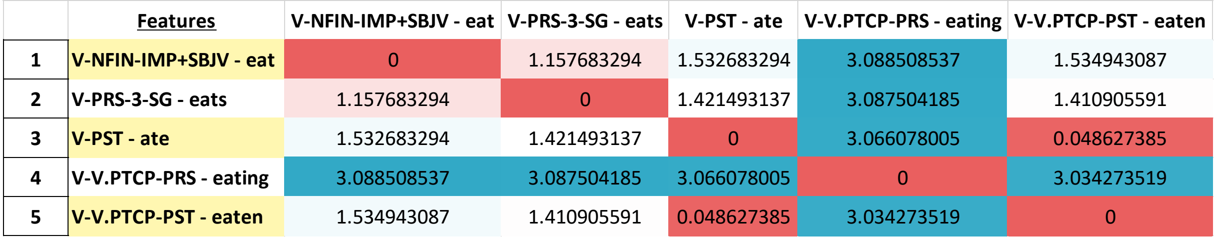
Figures 4, 5, and 6 illustrate the characterization tables for German.

B.3 Characterization Tables for Swedish

Figures 7, 8, and 9 present the characterization tables for Swedish.

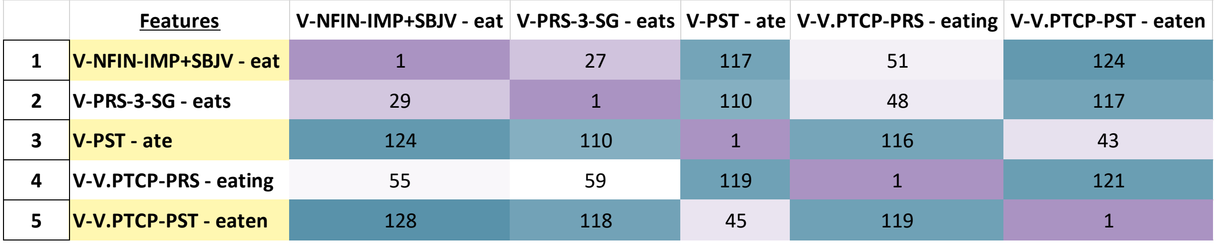
Language	**Features**	**Samples per Feature Set**	**# of Gold Principal Parts**	**Gold Principal Parts**
Hebrew	29	848–1,042	6	V-NFIN, (e.g., le’echol), V-2-SG-IMP-MASC, (e.g., echol!), V-3-SG-FUT-MASC, (e.g., yochal), V-3-SG-PST-MASC, (e.g., achal), V-SG-PRS-MASC, (e.g., ochel), V.MSDR (e.g., achila)
English	5	23,896–31,848	3	V-NFIN-IMP+SBJV (e.g., eat), V-PST (e.g., ate), V-V.PTCP-PST (e.g., eaten)
French	49	7,483–7,535	7	V-NFIN (e.g., mangier), V-IND-PRS-1-PL (e.g., manjons), V.PTCP-PST (e.g., mangié), V-IND-FUT-1-SG (e.g., mangerai), V-IND-PRS-1-SG (e.g., manju), V-IND-PRS-3-PL (e.g., manjüent), V-IND-PST-1-SG-PFV (e.g., manjai)
German	30	2,307–6,661	5	V-NFIN (e.g., essen), V.PTCP-PST (e.g., gegessen), [V-IND-SG-3-PST, V-IND-SG-1-PST (e.g., aß)], V-IND-SG-3-PRS (e.g., isst), [V-SBJV-SG-3-PST, V-SBJV-SG-1-PST (e.g., äße)]
Spanish	79	6,676–6,695	2	V-NFIN (e.g., comer), V-IND-PRS-1-SG (e.g., como)
Danish	8	162	5	V-ACT-NFIN (e.g., danse), V-ACT-IND-PRS (e.g., danser), V-ACT-IND-PST (e.g., dansede), V-ACT-IMP (e.g., dans), V.PTCP-PASS-PST (e.g., danset)
Swedish	19	2,114–2,536	5	[V-NFIN-ACT, V-IND-PL-ACT-PRS (e.g., äta)], V-IND-SG-ACT-PRS (e.g., äter), V-IND-SG-ACT-PST (e.g., åt), V-V.CVB-ACT (e.g., ätit), V-IMP-ACT (e.g., ät)
Finnish	161	7,221–7,226	4	V-NFIN-ACT+PASS (e.g., syödä), V-ACT-PRS-POS-IND-1-SG (e.g., syön), V-ACT-PST-POS-IND-3-SG (e.g., söi), V.PTCP-ACT-PST (e.g., syönyt)
Turkish	703	588	2	V-NFIN (e.g., içmek), V-IND-PRS-HAB-3-SG-POS-DECL (e.g., içer)
Latin	48	450–947	3	V-IND-ACT-PRS-1-SG (e.g., -pleō), V-NFIN-ACT-PRS (e.g., -plēre), V-V.MSDR-ACC-LGSPEC1 (e.g., -plētum)

Table 6: Summary of the Principal Parts Detection dataset by language, including gold principal parts and illustrative lexeme examples.



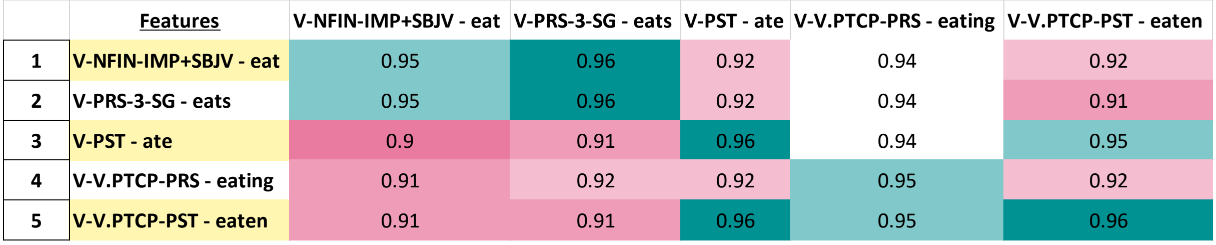
	Features	V-NFIN-IMP+SBJV - eat	V-PRS-3-SG - eats	V-PST - ate	V-V.PTCP-PRS - eating	V-V.PTCP-PST - eaten
1	V-NFIN-IMP+SBJV - eat	0	1.157683294	1.532683294	3.088508537	1.534943087
2	V-PRS-3-SG - eats	1.157683294	0	1.421493137	3.087504185	1.410905591
3	V-PST - ate	1.532683294	1.421493137	0	3.066078005	0.048627385
4	V-V.PTCP-PRS - eating	3.088508537	3.087504185	3.066078005	0	3.034273519
5	V-V.PTCP-PST - eaten	1.534943087	1.410905591	0.048627385	3.034273519	0

Figure 1: Average edit distances for the English verb paradigm. Values range from 0 to 3.088. Darker red shades indicate closer relationships between feature sets, while darker turquoise shades represent greater differences.



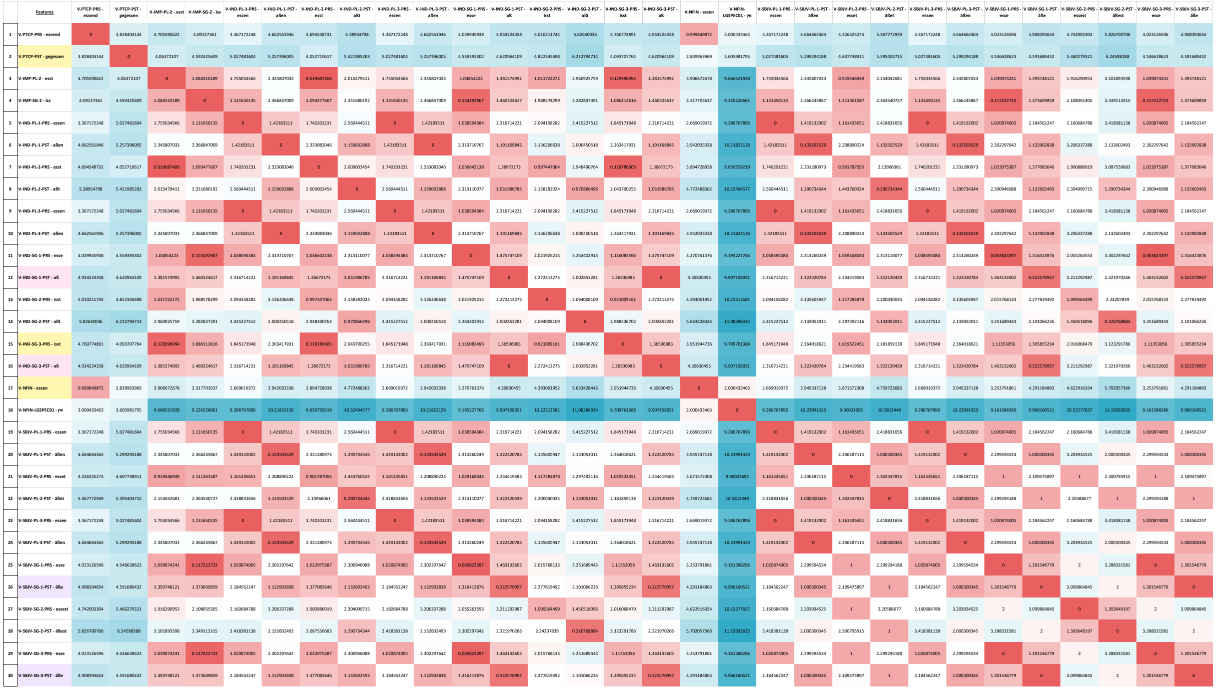
	Features	V-NFIN-IMP+SBJV - eat	V-PRS-3-SG - eats	V-PST - ate	V-V.PTCP-PRS - eating	V-V.PTCP-PST - eaten
1	V-NFIN-IMP+SBJV - eat	1	27	117	51	124
2	V-PRS-3-SG - eats	29	1	110	48	117
3	V-PST - ate	124	110	1	116	43
4	V-V.PTCP-PRS - eating	55	59	119	1	121
5	V-V.PTCP-PST - eaten	128	118	45	119	1

Figure 2: Edit Script scores for the English verb paradigm. Values range from 1 to 128. Darker purple shades indicate fewer unique character sets (closer relationships), while darker air-force-blue shades reflect greater variation.



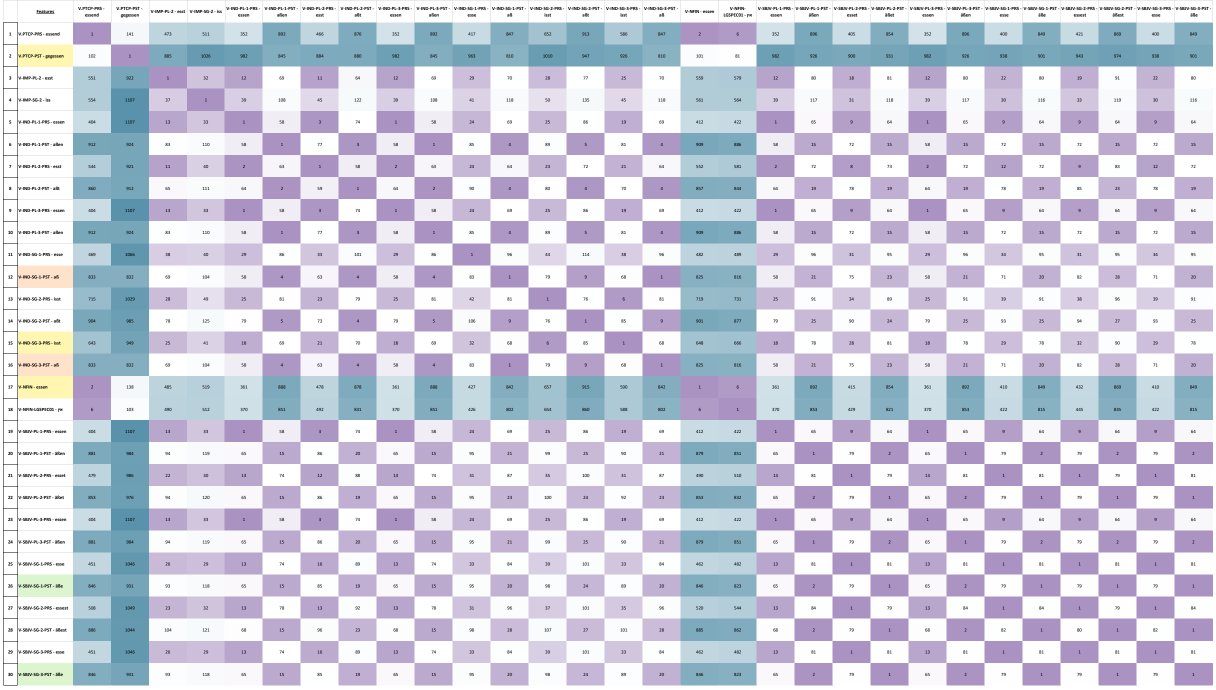
	Features	V-NFIN-IMP+SBJV - eat	V-PRS-3-SG - eats	V-PST - ate	V-V.PTCP-PRS - eating	V-V.PTCP-PST - eaten
1	V-NFIN-IMP+SBJV - eat	0.95	0.96	0.92	0.94	0.92
2	V-PRS-3-SG - eats	0.95	0.96	0.92	0.94	0.91
3	V-PST - ate	0.9	0.91	0.96	0.94	0.95
4	V-V.PTCP-PRS - eating	0.91	0.92	0.92	0.95	0.92
5	V-V.PTCP-PST - eaten	0.91	0.91	0.96	0.95	0.96

Figure 3: Reinflection Accuracy scores for the English verb paradigm. Values range from 0.9 to 0.96. Darker teal shades indicate higher accuracy, while darker pink shades reflect lower performance.



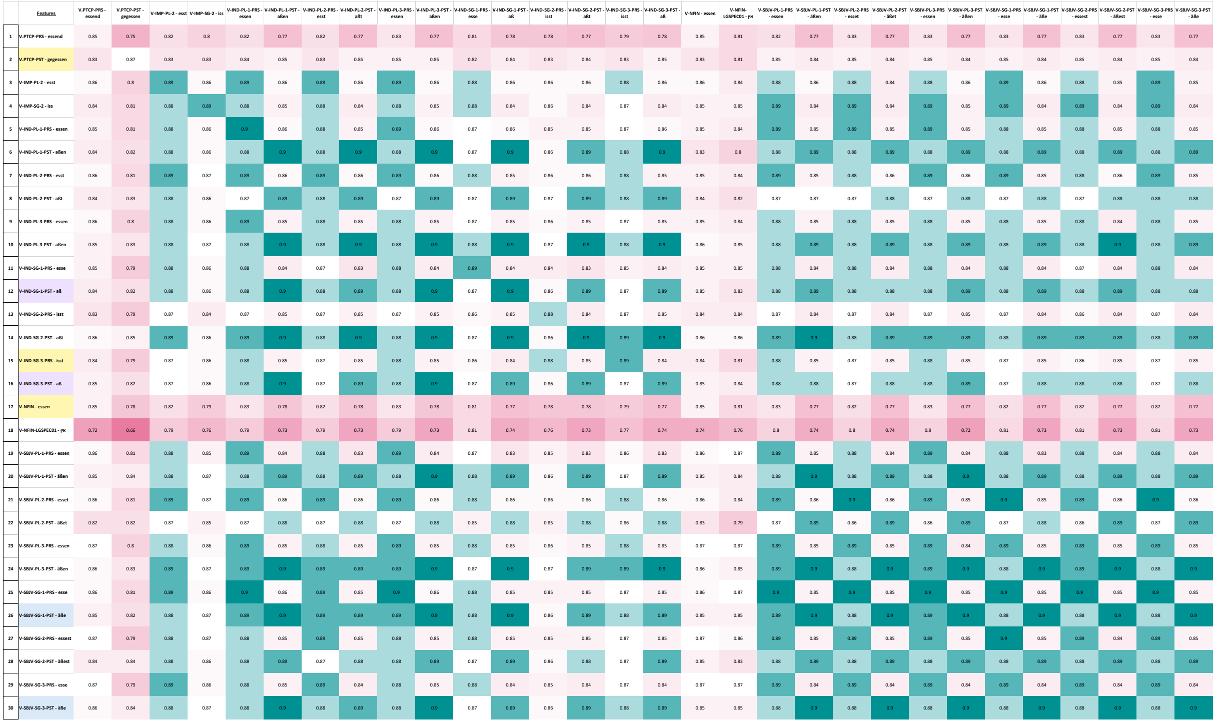
	Features	V.PTCP-PRS - essend	V.PTCP-PST - gegessen	V-IMP-PL-2 - esst	V-IMP-SG-2 - iss	V-IND-PL-1-PRS - essen	V-IND-PL-1-PST - aßen	V-IND-PL-2-PRS - esst	V-IND-PL-2-PST - aßt	V-IND-PL-3-PRS - essen	V-IND-PL-3-PST - aßen	V-IND-SG-1-PRS - esse	V-IND-SG-1-PST - aß	V-IND-SG-2-PRS - isst	V-IND-SG-2-PST - aßt	V-IND-SG-3-PRS - isst	V-IND-SG-3-PST - aß	V-NFIN - essen	V-NFIN-LGSPEC01 - אין	V-SBJV-PL-1-PRS - essen	V-SBJV-PL-1-PST - äßen	V-SBJV-PL-2-PRS - esset	V-SBJV-PL-2-PST - äßet	V-SBJV-PL-3-PRS - essen	V-SBJV-PL-3-PST - äßen	V-SBJV-SG-1-PRS - esse	V-SBJV-SG-1-PST - äße	V-SBJV-SG-2-PRS - essest	V-SBJV-SG-2-PST - äßest	V-SBJV-SG-3-PRS - esse	V-SBJV-SG-3-PST - äße
1	V.PTCP-PRS - essend	0	3.828404144	4.705590622	4.09137361	3.367172248	4.662561946	4.694548731	5.38954798	3.367172248	4.662561946	4.039945938	4.934224358	5.010211744	5.82640036	4.760774891	4.934224358	0.999849872	3.000433463	3.367172248	4.664664364	4.326325274	5.367772939	3.367172248	4.664664364	4.023126596	4.908394654	4.742003304	5.826700706	4.023126596	4.908394654
2	V.PTCP-PST - gegessen	3.828404144	0	4.06372107	4.592425609	5.027481604	5.257396005	4.052710617	5.421985283	5.027481604	5.257396005	4.559393302	4.620964109	4.822345698	6.212794714	4.093707764	4.620964109	2.839963969	3.605981795	5.027481604	5.299294188	4.607748911	5.395404715	5.027481604	5.299294188	4.546628623	4.591680432	5.460279321	6.24598288	4.546628623	4.591680432
3	V-IMP-PL-2 - esst	4.705590622	4.06372107	0	1.084310189	1.755034566	2.345807033	0.019687406	2.015479411	1.755034566	2.345807033	1.04854223	1.381574992	1.011722272	2.960925759	0.129996994	1.381574992	3.906672678	9.666521928	1.755034566	2.345807033	0.919446949	2.154042681	1.755034566	2.345807033	1.039074241	1.393748121	1.916290953	3.101893598	1.039074241	1.393748121
4	V-IMP-SG-2 - iss	4.09137361	4.592425609	1.084310189	0	1.131650135	2.366847009	1.093477607	2.331680192	1.131650135	2.366847009	0.154192967	1.460324617	1.988578299	3.282837391	1.086113616	1.460324617	3.317703637	9.226226661	1.131650135	2.366245867	1.111361587	2.363540727	1.131650135	2.366245867	0.117222723	1.373609859	2.108055305	3.349113315	0.117222723	1.373609859
5	V-IND-PL-1-PRS - essen	3.367172248	5.027481604	1.755034566	1.131650135	0	1.42183511	1.740201231	2.560444511	0	1.42183511	1.038594384	2.316714221	2.094158282	3.415227512	1.845171948	2.316714221	2.669019372	9.286767896	0	1.419132002	1.161435651	2.418831656	0	1.419132002	1.020874005	2.184562247	2.160684788	3.418381138	1.020874005	2.184562247
6	V-IND-PL-1-PST - aßen	4.662561946	5.257396005	2.345807033	2.366847009	1.42183511	0	2.333083046	1.159032888	1.42183511	0	2.313710767	1.191169845	3.136206638	2.000450518	2.363417931	1.191169845	3.942033338	10.21822126	1.42183511	0.133503529	2.208890224	1.133503529	1.42183511	0.133503529	2.302297642	1.132902838	3.206337288	2.132602493	2.302297642	1.132902838
7	V-IND-PL-2-PRS - esst	4.694548731	4.052710617	0.019687406	1.093477607	1.740201231	2.333083046	0	2.003003454	1.740201231	2.333083046	1.036642138	1.36672173	0.997447064	2.948490764	0.118786605	1.36672173	3.894728938	9.650759219	1.740201231	2.331280973	0.901787055	2.13966061	1.740201231	2.331280973	1.022075387	1.377083646	1.900886019	3.087550683	1.022075387	1.377083646
8	V-IND-PL-2-PST - aßt	5.38954798	5.421985283	2.015479411	2.331680192	2.560444511	1.159032888	2.003003454	0	2.560444511	1.159032888	2.313110077	1.031986785	2.158282024	0.970866496	2.043700255	1.031986785	4.772488362	10.52494577	2.560444511	1.290734344	1.443760324	0.290734344	2.560444511	1.290734344	2.300946088	1.132602493	2.304099715	1.290734344	2.300946088	1.132602493
9	V-IND-PL-3-PRS - essen	3.367172248	5.027481604	1.755034566	1.131650135	0	1.42183511	1.740201231	2.560444511	0	1.42183511	1.038594384	2.316714221	2.094158282	3.415227512	1.845171948	2.316714221	2.669019372	9.286767896	0	1.419132002	1.161435651	2.418831656	0	1.419132002	1.020874005	2.184562247	2.160684788	3.418381138	1.020874005	2.184562247
10	V-IND-PL-3-PST - aßen	4.662561946	5.257396005	2.345807033	2.366847009	1.42183511	0	2.333083046	1.159032888	1.42183511	0	2.313710767	1.191169845	3.136206638	2.000450518	2.363417931	1.191169845	3.942033338	10.21822126	1.42183511	0.133503529	2.208890224	1.133503529	1.42183511	0.133503529	2.302297642	1.132902838	3.206337288	2.132602493	2.302297642	1.132902838
11	V-IND-SG-1-PRS - esse	4.039945938	4.559393302	1.04854223	0.154192967	1.038594384	2.313710767	1.036642138	2.313110077	1.038594384	2.313710767	0	1.475747109	2.021925214	3.263402913	1.116083496	1.475747109	3.270761376	9.195227766	1.038594384	2.313260249	1.059168043	2.313110077	1.038594384	2.313260249	0.063823397	1.316413876	2.055263553	3.302297642	0.063823397	1.316413876
12	V-IND-SG-1-PST - aß	4.934224358	4.620964109	1.381574992	1.460324617	2.316714221	1.191169845	1.36672173	1.031986785	2.316714221	1.191169845	1.475747109	0	2.272413275	2.002853281	1.36506983	0	4.30830455	9.907158351	2.316714221	1.322420784	2.234419583	1.322120439	2.316714221	1.322420784	1.463132602	0.322570957	3.211292987	2.321970266	1.463132602	0.322570957
13	V-IND-SG-2-PRS - isst	5.010211744	4.822345698	1.011722272	1.988578299	2.094158282	3.136206638	0.997447064	2.158282024	2.094158282	3.136206638	2.021925214	2.272413275	0	2.094008109	0.921009161	2.272413275	4.393001952	10.22212581	2.094158282	3.135605947	1.117284878	2.200030035	2.094158282	3.135605947	2.015768133	2.277819492	1.099564499	2.24207839	2.015768133	2.277819492
14	V-IND-SG-2-PST - aßt	5.82640036	6.212794714	2.960925759	3.282837391	3.415227512	2.000450518	2.948490764	0.970866496	3.415227512	2.000450518	3.263402913	2.002853281	2.094008109	0	2.988436702	2.002853281	5.633428443	11.08286334	3.415227512	2.133053011	2.297492116	1.133053011	3.415227512	2.133053011	3.251689443	2.101066226	1.450518096	0.320768884	3.251689443	2.101066226
15	V-IND-SG-3-PRS - isst	4.760774891	4.093707764	0.129996994	1.086113616	1.845171948	2.363417931	0.118786605	2.043700255	1.845171948	2.363417931	1.116083496	1.36506983	0.921009161	2.988436702	0	1.36506983	3.951944736	9.709761388	1.845171948	2.364018621	1.019522451	2.181859138	1.845171948	2.364018621	1.11353056	1.395855234	2.016068479	3.123291786	1.11353056	1.395855234
16	V-IND-SG-3-PST - aß	4.934224358	4.620964109	1.381574992	1.460324617	2.316714221	1.191169845	1.36672173	1.031986785	2.316714221	1.191169845	1.475747109	0	2.272413275	2.002853281	1.36506983	0	4.30830455	9.907158351	2.316714221	1.322420784	2.234419583	1.322120439	2.316714221	1.322420784	1.463132602	0.322570957	3.211292987	2.321970266	1.463132602	0.322570957
17	V-NFIN - essen	0.999849872	2.839963969	3.906672678	3.317703637	2.669019372	3.942033338	3.894728938	4.772488362	2.669019372	3.942033338	3.270761376	4.30830455	4.393001952	5.633428443	3.951944736	4.30830455	0	2.000433463	2.669019372	3.945337138	3.671572308	4.759723682	2.669019372	3.945337138	3.253791861	4.291184863	4.622916354	5.702057366	3.253791861	4.291184863
18	V-NFIN-LGSPEC01 - אין	3.000433463	3.605981795	9.666521928	9.226226661	9.286767896	10.21822126	9.650759219	10.52494577	9.286767896	10.21822126	9.195227766	9.907158351	10.22212581	11.08286334	9.709761388	9.907158351	2.000433463	0	9.286767896	10.23991323	9.90021692	10.5813449	9.286767896	10.23991323	9.161388286	9.966160521	10.52277657	11.19392625	9.161388286	9.966160521
19	V-SBJV-PL-1-PRS - essen	3.367172248	5.027481604	1.755034566	1.131650135	0	1.42183511	1.740201231	2.560444511	0	1.42183511	1.038594384	2.316714221	2.094158282	3.415227512	1.845171948	2.316714221	2.669019372	9.286767896	0	1.419132002	1.161435651	2.418831656	0	1.419132002	1.020874005	2.184562247	2.160684788	3.418381138	1.020874005	2.184562247
20	V-SBJV-PL-1-PST - äßen	4.664664364	5.299294188	2.345807033	2.366245867	1.419132002	0.133503529	2.331280973	1.290734344	1.419132002	0.133503529	2.313260249	1.322420784	3.135605947	2.133053011	2.364018621	1.322420784	3.945337138	10.23991323	1.419132002	0	2.206187115	1.000300345	1.419132002	0	2.299594534	1.000300345	3.203934525	2.000300345	2.299594534	1.000300345
21	V-SBJV-PL-2-PRS - esset	4.326325274	4.607748911	0.919446949	1.111361587	1.161435651	2.208890224	0.901787055	1.443760324	1.161435651	2.208890224	1.059168043	2.234419583	1.117284878	2.297492116	1.019522451	2.234419583	3.671572308	9.90021692	1.161435651	2.206187115	0	1.302447815	1.161435651	2.206187115	1	2.109475897	1	2.300795915	1	2.109475897
22	V-SBJV-PL-2-PST - äßet	5.367772939	5.395404715	2.154042681	2.363540727	2.418831656	1.133503529	2.13966061	0.290734344	2.418831656	1.133503529	2.313110077	1.322120439	2.200030035	1.133053011	2.181859138	1.322120439	4.759723682	10.5813449	2.418831656	1.000300345	1.302447815	0	2.418831656	1.000300345	2.299294188	1	2.20588677	1	2.299294188	1
23	V-SBJV-PL-3-PRS - essen	3.367172248	5.027481604	1.755034566	1.131650135	0	1.42183511	1.740201231	2.560444511	0	1.42183511	1.038594384	2.316714221	2.094158282	3.415227512	1.845171948	2.316714221	2.669019372	9.286767896	0	1.419132002	1.161435651	2.418831656	0	1.419132002	1.020874005	2.184562247	2.160684788	3.418381138	1.020874005	2.184562247
24	V-SBJV-PL-3-PST - äßen	4.664664364	5.299294188	2.345807033	2.366245867	1.419132002	0.133503529	2.331280973	1.290734344	1.419132002	0.133503529	2.313260249	1.322420784	3.135605947	2.133053011	2.364018621	1.322420784	3.945337138	10.23991323	1.419132002	0	2.206187115	1.000300345	1.419132002	0	2.299594534	1.000300345	3.203934525	2.000300345	2.299594534	1.000300345
25	V-SBJV-SG-1-PRS - esse	4.023126596	4.546628623	1.039074241	0.117222723	1.020874005	2.302297642	1.022075387	2.300946088	1.020874005	2.302297642	0.063823397	1.463132602	2.015768133	3.251689443	1.11353056	1.463132602	3.253791861	9.161388286	1.020874005	2.299594534	1	2.299294188	1.020874005	2.299594534	0	1.301546779	2	3.288331581	0	1.301546779
26	V-SBJV-SG-1-PST - äße	4.908394654	4.591680432	1.393748121	1.373609859	2.184562247	1.132902838	1.377083646	1.132602493	2.184562247	1.132902838	1.316413876	0.322570957	2.277819492	2.101066226	1.395855234	0.322570957	4.291184863	9.966160521	2.184562247	1.000300345	2.109475897	1	2.184562247	1.000300345	1.301546779	0	3.099864845	2	1.301546779	0
27	V-SBJV-SG-2-PRS - essest	4.742003304	5.460279321	1.916290953	2.108055305	2.160684788	3.206337288	1.900886019	2.304099715	2.160684788	3.206337288	2.055263553	3.211292987	1.099564499	1.450518096	2.016068479	3.211292987	4.622916354	10.52277657	2.160684788	3.203934525	1	2.20588677	2.160684788	3.203934525	2	3.099864845	0	1.303649197	2	3.099864845
28	V-SBJV-SG-2-PST - äßest	5.826700706	6.24598288	3.101893598	3.349113315	3.418381138	2.132602493	3.087550683	1.290734344	3.418381138	2.132602493	3.302297642	2.321970266	2.24207839	0.320768884	3.123291786	2.321970266	5.702057366	11.19392625	3.418381138	2.000300345	2.300795915	1	3.418381138	2.000300345	3.288331581	2	1.303649197	0	3.288331581	2
29	V-SBJV-SG-3-PRS - esse	4.023126596	4.546628623	1.039074241	0.117222723	1.020874005	2.302297642	1.022075387	2.300946088	1.020874005	2.302297642	0.063823397	1.463132602	2.015768133	3.251689443	1.11353056	1.463132602	3.253791861	9.161388286	1.020874005	2.299594534	1	2.299294188	1.020874005	2.299594534	0	1.301546779	2	3.288331581	0	1.301546779
30	V-SBJV-SG-3-PST - äße	4.908394654	4.591680432	1.393748121	1.373609859	2.184562247	1.132902838	1.377083646	1.132602493	2.184562247	1.132902838	1.316413876	0.322570957	2.277819492	2.101066226	1.395855234	0.322570957	4.291184863	9.966160521	2.184562247	1.000300345	2.109475897	1	2.184562247	1.000300345	1.301546779	0	3.099864845	2	1.301546779	0

Figure 4: Average edit distances for the German verb paradigm. Values range from 0 to 11.19. Darker red shades indicate closer relationships between feature sets, while darker ball-blue shades represent greater distances.



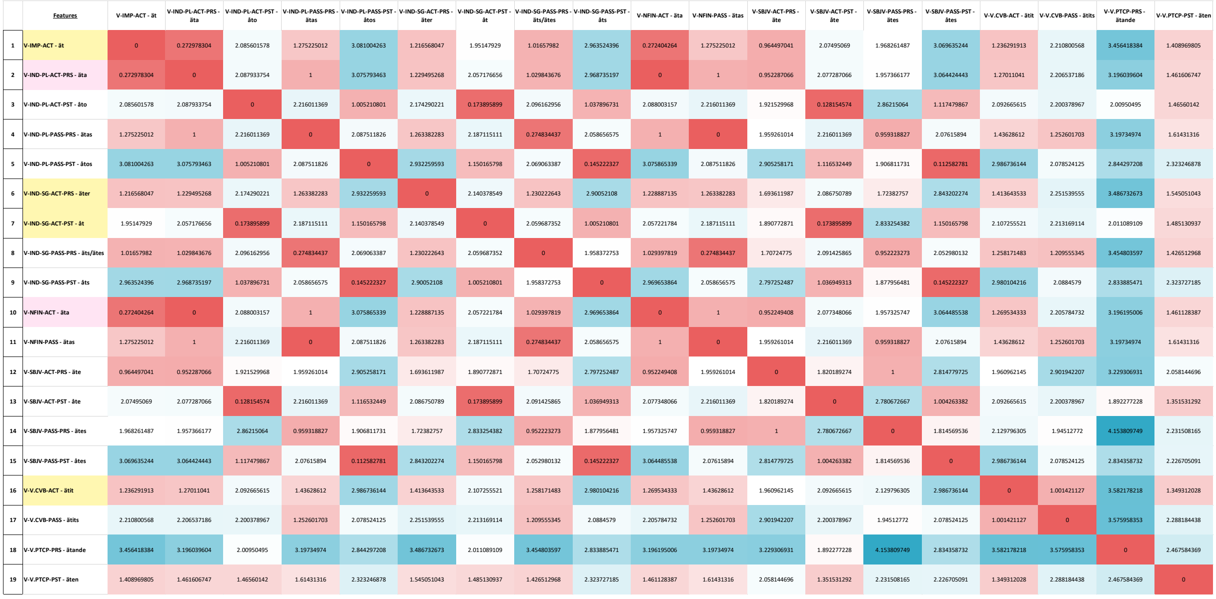
	Features	V.PTCP-PRS - essend	V.PTCP-PST - gegessen	V-IMP-PL-2 - esst	V-IMP-SG-2 - iss	V-IND-PL-1-PRS - essen	V-IND-PL-1-PST - aßen	V-IND-PL-2-PRS - esst	V-IND-PL-2-PST - aßt	V-IND-PL-3-PRS - essen	V-IND-PL-3-PST - aßen	V-IND-SG-1-PRS - esse	V-IND-SG-1-PST - aß	V-IND-SG-2-PRS - isst	V-IND-SG-2-PST - aßt	V-IND-SG-3-PRS - isst	V-IND-SG-3-PST - aß	V-NFIN - essen	V-NFIN-LGSPEC01 - אין	V-SBJV-PL-1-PRS - essen	V-SBJV-PL-1-PST - äßen	V-SBJV-PL-2-PRS - esset	V-SBJV-PL-2-PST - äßet	V-SBJV-PL-3-PRS - essen	V-SBJV-PL-3-PST - äßen	V-SBJV-SG-1-PRS - esse	V-SBJV-SG-1-PST - äße	V-SBJV-SG-2-PRS - essest	V-SBJV-SG-2-PST - äßest	V-SBJV-SG-3-PRS - esse	V-SBJV-SG-3-PST - äße
1	V.PTCP-PRS - essend	1	141	473	511	352	892	466	876	352	892	417	847	652	913	586	847	2	6	352	896	405	854	352	896	400	849	421	869	400	849
2	V.PTCP-PST - gegessen	102	1	885	1026	982	845	884	880	982	845	963	810	1010	947	926	810	101	81	982	926	900	931	982	926	938	901	943	974	938	901
3	V-IMP-PL-2 - esst	551	922	1	32	12	69	11	64	12	69	29	70	28	77	25	70	559	579	12	80	18	81	12	80	22	80	19	91	22	80
4	V-IMP-SG-2 - iss	554	1107	37	1	39	108	45	122	39	108	41	118	50	135	45	118	561	564	39	117	31	118	39	117	30	116	33	119	30	116
5	V-IND-PL-1-PRS - essen	404	1107	13	33	1	58	3	74	1	58	24	69	25	86	19	69	412	422	1	65	9	64	1	65	9	64	9	64	9	64
6	V-IND-PL-1-PST - aßen	912	924	83	110	58	1	77	3	58	1	85	4	89	5	81	4	909	886	58	15	72	15	58	15	72	15	72	15	72	15
7	V-IND-PL-2-PRS - esst	544	921	11	40	2	63	1	58	2	63	24	64	23	72	21	64	552	581	2	72	8	73	2	72	12	72	9	83	12	72
8	V-IND-PL-2-PST - aßt	860	912	65	111	64	2	59	1	64	2	90	4	80	4	70	4	857	844	64	19	78	19	64	19	78	19	85	23	78	19
9	V-IND-PL-3-PRS - essen	404	1107	13	33	1	58	3	74	1	58	24	69	25	86	19	69	412	422	1	65	9	64	1	65	9	64	9	64	9	64
10	V-IND-PL-3-PST - aßen	912	924	83	110	58	1	77	3	58	1	85	4	89	5	81	4	909	886	58	15	72	15	58	15	72	15	72	15	72	15
11	V-IND-SG-1-PRS - esse	469	1066	38	40	29	86	33	101	29	86	1	96	44	114	38	96	482	489	29	96	31	95	29	96	34	95	31	95	34	95
12	V-IND-SG-1-PST - aß	833	832	69	104	58	4	63	4	58	4	83	1	79	9	68	1	825	816	58	21	75	23	58	21	71	20	82	28	71	20
13	V-IND-SG-2-PRS - isst	715	1029	28	49	25	81	23	79	25	81	42	81	1	76	6	81	719	731	25	91	34	89	25	91	39	91	38	96	39	91
14	V-IND-SG-2-PST - aßt	904	985	78	125	79	5	73	4	79	5	106	9	76	1	85	9	901	877	79	25	90	24	79	25	93	25	94	27	93	25
15	V-IND-SG-3-PRS - isst	643	949	25	41	18	69	21	70	18	69	32	68	6	85	1	68	648	666	18	78	28	81	18	78	29	78	32	90	29	78
16	V-IND-SG-3-PST - aß	833	832	69	104	58	4	63	4	58	4	83	1	79	9	68	1	825	816	58	21	75	23	58	21	71	20	82	28	71	20
17	V-NFIN - essen	2	138	485	519	361	888	478	878	361	888	427	842	657	915	590	842	1	6	361	892	415	854	361	892	410	849	432	869	410	849
18	V-NFIN-LGSPEC01 - אין	6	103	490	512	370	851	492	831	370	851	426	802	654	860	588	802	6	1	370	853	429	821	370	853	422	815	445	835	422	815
19	V-SBJV-PL-1-PRS - essen	404	1107	13	33	1	58	3	74	1	58	24	69	25	86	19	69	412	422	1	65	9	64	1	65	9	64	9	64	9	64
20	V-SBJV-PL-1-PST - äßen	881	984	94	119	65	15	86	20	65	15	95	21	99	25	90	21	879	851	65	1	79	2	65	1	79	2	79	2	79	2
21	V-SBJV-PL-2-PRS - esset	479	986	22	30	13	74	12	88	13	74	31	87	35	100	31	87	490	510	13	81	1	79	13	81	1	81	1	79	1	81
22	V-SBJV-PL-2-PST - äßet	853	976	94	120	65	15	86	19	65	15	95	23	100	24	92	23	853	832	65	2	79	1	65	2	79	1	79	1	79	1
23	V-SBJV-PL-3-PRS - essen	404	1107	13	33	1	58	3	74	1	58	24	69	25	86	19	69	412	422	1	65	9	64	1	65	9	64	9	64	9	64
24	V-SBJV-PL-3-PST - äßen	881	984	94	119	65	15	86	20	65	15	95	21	99	25	90	21	879	851	65	1	79	2	65	1	79	2	79	2	79	2
25	V-SBJV-SG-1-PRS - esse	451	1046	26	29	13	74	16	89	13	74	33	84	39	101	33	84	462	482	13	81	1	81	13	81	1	81	1	81	1	81
26	V-SBJV-SG-1-PST - äße	846	931	93	118	65	15	85	19	65	15	95	20	98	24	89	20	846	823	65	2	79	1	65	2	79	1	79	1	79	1
27	V-SBJV-SG-2-PRS - essest	508	1049	23	32	13	78	13	92	13	78	31	96	37	101	35	96	520	544	13	84	1	79	13	84	1	84	1	79	1	84
28	V-SBJV-SG-2-PST - äßest	886	1044	104	121	68	15	96	23	68	15	98	28	107	27	101	28	885	862	68	2	79	1	68	2	82	1	80	1	82	1
29	V-SBJV-SG-3-PRS - esse	451	1046	26	29	13	74	16	89	13	74	33	84	39	101	33	84	462	482	13	81	1	81	13	81	1	81	1	81	1	81
30	V-SBJV-SG-3-PST - äße	846	931	93	118	65	15	85	19	65	15	95	20	98	24	89	20	846	823	65	2	79	1	65	2	79	1	79	1	79	1

Figure 5: Edit Script scores for the German Verb Paradigm. Values range from 1 to 1,107. Darker purple shades indicate fewer unique character sets (closer relationships), while darker air-force-blue shades reflect greater variation.



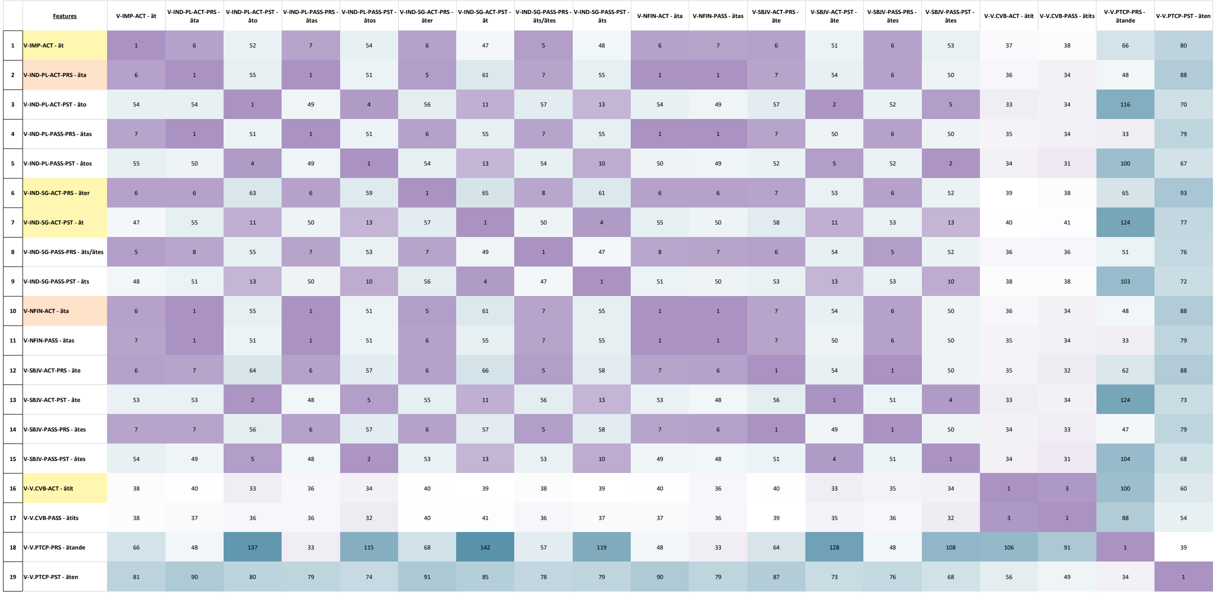
	Features	V.PTCP-PRS - essend	V.PTCP-PST - gegessen	V-IMP-PL-2 - esst	V-IMP-SG-2 - iss	V-IND-PL-1-PRS - essen	V-IND-PL-1-PST - aßen	V-IND-PL-2-PRS - esst	V-IND-PL-2-PST - aßt	V-IND-PL-3-PRS - essen	V-IND-PL-3-PST - aßen	V-IND-SG-1-PRS - esse	V-IND-SG-1-PST - aß	V-IND-SG-2-PRS - isst	V-IND-SG-2-PST - aßt	V-IND-SG-3-PRS - isst	V-IND-SG-3-PST - aß	V-NFIN - essen	V-NFIN-LGSPEC01 - אין	V-SBJV-PL-1-PRS - essen	V-SBJV-PL-1-PST - äßen	V-SBJV-PL-2-PRS - esset	V-SBJV-PL-2-PST - äßet	V-SBJV-PL-3-PRS - essen	V-SBJV-PL-3-PST - äßen	V-SBJV-SG-1-PRS - esse	V-SBJV-SG-1-PST - äße	V-SBJV-SG-2-PRS - essest	V-SBJV-SG-2-PST - äßest	V-SBJV-SG-3-PRS - esse	V-SBJV-SG-3-PST - äße
1	V.PTCP-PRS - essend	0.85	0.75	0.82	0.8	0.82	0.77	0.82	0.77	0.83	0.77	0.81	0.78	0.78	0.77	0.79	0.78	0.85	0.81	0.82	0.77	0.83	0.77	0.83	0.77	0.83	0.77	0.83	0.77	0.83	0.77
2	V.PTCP-PST - gegessen	0.83	0.87	0.83	0.83	0.84	0.85	0.83	0.85	0.85	0.85	0.82	0.84	0.83	0.84	0.83	0.85	0.83	0.81	0.85	0.84	0.85	0.84	0.85	0.84	0.85	0.84	0.85	0.84	0.85	0.84
3	V-IMP-PL-2 - esst	0.86	0.8	0.89	0.86	0.89	0.86	0.89	0.86	0.89	0.86	0.88	0.86	0.86	0.86	0.88	0.86	0.86	0.84	0.88	0.86	0.88	0.84	0.88	0.86	0.89	0.86	0.88	0.85	0.89	0.85
4	V-IMP-SG-2 - iss	0.84	0.81	0.88	0.89	0.88	0.85	0.88	0.84	0.88	0.85	0.88	0.84	0.86	0.84	0.87	0.84	0.85	0.85	0.89	0.84	0.89	0.84	0.89	0.85	0.89	0.84	0.89	0.84	0.89	0.84
5	V-IND-PL-1-PRS - essen	0.85	0.81	0.88	0.86	0.9	0.86	0.88	0.85	0.89	0.86	0.87	0.86	0.85	0.85	0.87	0.86	0.85	0.84	0.89	0.85	0.89	0.85	0.89	0.85	0.88	0.85	0.88	0.85	0.88	0.85
6	V-IND-PL-1-PST - aßen	0.84	0.82	0.88	0.86	0.88	0.9	0.88	0.9	0.88	0.9	0.87	0.9	0.86	0.89	0.88	0.9	0.83	0.8	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89
7	V-IND-PL-2-PRS - esst	0.86	0.81	0.89	0.87	0.89	0.86	0.89	0.86	0.89	0.86	0.88	0.85	0.86	0.86	0.88	0.85	0.85	0.84	0.89	0.85	0.88	0.86	0.89	0.86	0.89	0.85	0.88	0.86	0.89	0.85
8	V-IND-PL-2-PST - aßt	0.84	0.83	0.88	0.86	0.87	0.89	0.88	0.89	0.87	0.89	0.87	0.89	0.87	0.89	0.88	0.89	0.84	0.82	0.87	0.87	0.87	0.88	0.87	0.88	0.87	0.88	0.87	0.88	0.88	0.88
9	V-IND-PL-3-PRS - essen	0.86	0.8	0.88	0.86	0.89	0.85	0.88	0.85	0.88	0.85	0.87	0.85	0.86	0.85	0.87	0.85	0.85	0.84	0.88	0.85	0.88	0.85	0.88	0.85	0.88	0.85	0.88	0.84	0.88	0.85
10	V-IND-PL-3-PST - aßen	0.85	0.83	0.88	0.87	0.88	0.9	0.88	0.9	0.88	0.9	0.88	0.9	0.87	0.9	0.88	0.9	0.86	0.85	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.9	0.88	0.89
11	V-IND-SG-1-PRS - esse	0.85	0.79	0.88	0.86	0.88	0.84	0.87	0.83	0.88	0.84	0.89	0.84	0.84	0.83	0.85	0.84	0.85	0.85	0.88	0.84	0.88	0.84	0.88	0.84	0.88	0.84	0.87	0.84	0.88	0.84
12	V-IND-SG-1-PST - aß	0.84	0.82	0.88	0.86	0.88	0.9	0.88	0.89	0.88	0.9	0.87	0.9	0.86	0.89	0.87	0.89	0.85	0.83	0.88	0.89	0.88	0.88	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.88
13	V-IND-SG-2-PRS - isst	0.83	0.79	0.87	0.84	0.87	0.85	0.87	0.85	0.87	0.85	0.86	0.85	0.88	0.84	0.87	0.85	0.84	0.84	0.87	0.84	0.87	0.84	0.87	0.85	0.87	0.84	0.86	0.84	0.87	0.84
14	V-IND-SG-2-PST - aßt	0.86	0.85	0.89	0.86	0.89	0.9	0.88	0.9	0.88	0.9	0.87	0.9	0.86	0.9	0.89	0.9	0.86	0.86	0.89	0.9	0.88	0.89	0.89	0.89	0.88	0.89	0.89	0.89	0.88	0.89
15	V-IND-SG-3-PRS - isst	0.84	0.79	0.87	0.86	0.88	0.85	0.87	0.85	0.88	0.85	0.86	0.84	0.88	0.85	0.89	0.84	0.84	0.81	0.88	0.85	0.87	0.85	0.88	0.85	0.87	0.85	0.86	0.85	0.87	0.85
16	V-IND-SG-3-PST - aß	0.85	0.82	0.87	0.86	0.88	0.9	0.87	0.89	0.88	0.9	0.87	0.89	0.86	0.89	0.87	0.89	0.85	0.84	0.88	0.88	0.87	0.88	0.88	0.89	0.87	0.88	0.88	0.88	0.87	0.88
17	V-NFIN - essen	0.85	0.78	0.82	0.79	0.83	0.78	0.82	0.78	0.83	0.78	0.81	0.77	0.78	0.78	0.79	0.77	0.85	0.81	0.83	0.77	0.82	0.77	0.83	0.77	0.82	0.77	0.82	0.77	0.82	0.77
18	V-NFIN-LGSPEC01 - אין	0.72	0.66	0.79	0.76	0.79	0.73	0.79	0.73	0.79	0.73	0.81	0.74	0.76	0.73	0.77	0.74	0.74	0.76	0.8	0.74	0.8	0.74	0.8	0.72	0.81	0.73	0.81	0.73	0.81	0.73
19	V-SBJV-PL-1-PRS - essen	0.86	0.81	0.88	0.85	0.89	0.84	0.88	0.83	0.89	0.84	0.87	0.83	0.85	0.83	0.86	0.83	0.86	0.87	0.89	0.85	0.88	0.84	0.89	0.84	0.88	0.83	0.88	0.84	0.88	0.84
20	V-SBJV-PL-1-PST - äßen	0.85	0.84	0.88	0.87	0.88	0.89	0.88	0.89	0.88	0.9	0.88	0.89	0.86	0.89	0.87	0.89	0.85	0.84	0.88	0.9	0.88	0.89	0.88	0.9	0.88	0.89	0.88	0.89	0.88	0.89
21	V-SBJV-PL-2-PRS - esset	0.86	0.81	0.89	0.87	0.89	0.86	0.89	0.86	0.89	0.86	0.88	0.86	0.86	0.86	0.88	0.86	0.86	0.84	0.89	0.86	0.9	0.86	0.89	0.85	0.9	0.85	0.89	0.86	0.9	0.86
22	V-SBJV-PL-2-PST - äßet	0.82	0.82	0.87	0.85	0.87	0.88	0.87	0.88	0.87	0.88	0.85	0.88	0.85	0.88	0.86	0.88	0.83	0.79	0.87	0.89	0.86	0.89	0.86	0.89	0.87	0.88	0.86	0.89	0.87	0.89
23	V-SBJV-PL-3-PRS - essen	0.87	0.8	0.88	0.86	0.89	0.85	0.88	0.85	0.89	0.85	0.88	0.85	0.86	0.85	0.88	0.85	0.87	0.87	0.89	0.85	0.89	0.85	0.89	0.84	0.89	0.85	0.89	0.85	0.89	0.85
24	V-SBJV-PL-3-PST - äßen	0.86	0.83	0.89	0.87	0.89	0.9	0.89	0.89	0.89	0.9	0.87	0.9	0.87	0.89	0.89	0.9	0.86	0.85	0.89	0.9	0.88	0.9	0.89	0.9	0.88	0.9	0.89	0.9	0.88	0.9
25	V-SBJV-SG-1-PRS - esse	0.86	0.81	0.89	0.86	0.9	0.86	0.89	0.85	0.9	0.86	0.88	0.85	0.85	0.85	0.87	0.85	0.86	0.87	0.9	0.85	0.9	0.85	0.9	0.85	0.9	0.85	0.9	0.85	0.9	0.85
26	V-SBJV-SG-1-PST - äße	0.85	0.82	0.88	0.87	0.89	0.9	0.89	0.89	0.89	0.9	0.88	0.9	0.86	0.89	0.88	0.89	0.85	0.85	0.89	0.9	0.88	0.9	0.89	0.9	0.88	0.9	0.88	0.9	0.88	0.9
27	V-SBJV-SG-2-PRS - essest	0.87	0.79	0.88	0.87	0.88	0.85	0.89	0.85	0.88	0.85	0.88	0.85	0.85	0.85	0.87	0.85	0.87	0.86	0.89	0.85	0.89	0.85	0.89	0.85	0.9	0.85	0.89	0.84	0.89	0.85
28	V-SBJV-SG-2-PST - äßest	0.84	0.84	0.88	0.86	0.88	0.89	0.87	0.88	0.88	0.89	0.87	0.89	0.86	0.88	0.87	0.89	0.85	0.83	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89	0.88	0.89
29	V-SBJV-SG-3-PRS - esse	0.87	0.79	0.89	0.86	0.88	0.85	0.89	0.84	0.88	0.85	0.88	0.84	0.85	0.84	0.87	0.84	0.87	0.87	0.89	0.84	0.89	0.84	0.89	0.84	0.89	0.84	0.89	0.84	0.89	0.84
30	V-SBJV-SG-3-PST - äße	0.86	0.84	0.88	0.87	0.88	0.9	0.88	0.89	0.88	0.9	0.87	0.89	0.86	0.89	0.88	0.89	0.85	0.85	0.88	0.9	0.88	0.9	0.88	0.9	0.88	0.9	0.88	0.9	0.88	0.9

Figure 6: Reinflection Accuracy scores for the German verb paradigm. Values range from 0.66 to 0.9. Darker teal shades indicate higher accuracy, while darker pink shades reflect lower performance.



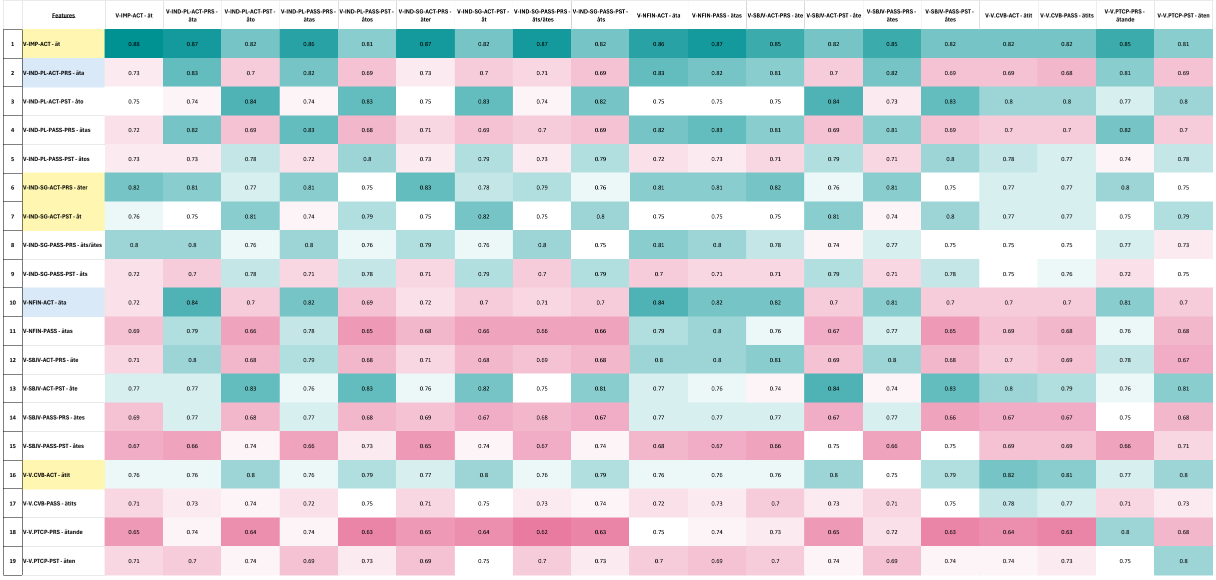
	Features	V-IMP-ACT - ät	V-IND-PL-ACT-PRS - äta	V-IND-PL-ACT-PST - åto	V-IND-PL-PASS-PRS - ätas	V-IND-PL-PASS-PST - åtos	V-IND-SG-ACT-PRS - äter	V-IND-SG-ACT-PST - åt	V-IND-SG-PASS-PRS - äts/ätes	V-IND-SG-PASS-PST - åts	V-NFIN-ACT - äta	V-NFIN-PASS - ätas	V-SBJV-ACT-PRS - äte	V-SBJV-ACT-PST - åte	V-SBJV-PASS-PRS - ätes	V-SBJV-PASS-PST - åtes	V-V.CVB-ACT - ätit	V-V.CVB-PASS - ätits	V-V.PTCP-PRS - ätande	V-V.PTCP-PST - äten
1	V-IMP-ACT - ät	0	0.272978304	2.085601578	1.275225012	3.081004263	1.216568047	1.95147929	1.01657982	2.963524396	0.272404264	1.275225012	0.964497041	2.07495069	1.968261487	3.069635244	1.236291913	2.210800568	3.456418384	1.408969805
2	V-IND-PL-ACT-PRS - äta	0.272978304	0	2.087933754	1	3.075793463	1.229495268	2.057176656	1.029843676	2.968735197	0	1	0.952287066	2.077287066	1.957366177	3.064424443	1.27011041	2.206537186	3.196039604	1.461606747
3	V-IND-PL-ACT-PST - åto	2.085601578	2.087933754	0	2.216011369	1.005210801	2.174290221	0.173895899	2.096162956	1.037896731	2.088003157	2.216011369	1.921529968	0.128154574	2.86215064	1.117479867	2.092665615	2.200378967	2.00950495	1.46560142
4	V-IND-PL-PASS-PRS - ätas	1.275225012	1	2.216011369	0	2.087511826	1.263382283	2.187115111	0.274834437	2.058656575	1	0	1.959261014	2.216011369	0.959318827	2.07615894	1.43628612	1.252601703	3.19734974	1.61431316
5	V-IND-PL-PASS-PST - åtos	3.081004263	3.075793463	1.005210801	2.087511826	0	2.932259593	1.150165798	2.069063387	0.145222327	3.075865339	2.087511826	2.905258171	1.116532449	1.906811731	0.112582781	2.986736144	2.078524125	2.844297208	2.323246878
6	V-IND-SG-ACT-PRS - äter	1.216568047	1.229495268	2.174290221	1.263382283	2.932259593	0	2.140378549	1.230222643	2.90052108	1.228887135	1.263382283	1.693611987	2.086750789	1.72382757	2.843202274	1.413643533	2.251539555	3.486732673	1.545051043
7	V-IND-SG-ACT-PST - åt	1.95147929	2.057176656	0.173895899	2.187115111	1.150165798	2.140378549	0	2.059687352	1.005210801	2.057221784	2.187115111	1.890772871	0.173895899	2.833254382	1.150165798	2.107255521	2.213169114	2.011089109	1.485130937
8	V-IND-SG-PASS-PRS - äts/ätes	1.01657982	1.029843676	2.096162956	0.274834437	2.069063387	1.230222643	2.059687352	0	1.958372753	1.029397819	0.274834437	1.70724775	2.091425865	0.952223273	2.052980132	1.258171483	1.209555345	3.454803597	1.426512968
9	V-IND-SG-PASS-PST - åts	2.963524396	2.968735197	1.037896731	2.058656575	0.145222327	2.90052108	1.005210801	1.958372753	0	2.969653864	2.058656575	2.797252487	1.036949313	1.877956481	0.145222327	2.980104216	2.0884579	2.833885471	2.323727185
10	V-NFIN-ACT - äta	0.272404264	0	2.088003157	1	3.075865339	1.228887135	2.057221784	1.029397819	2.969653864	0	1	0.952249408	2.077348066	1.957325747	3.064485538	1.269534333	2.205784732	3.196195006	1.461128387
11	V-NFIN-PASS - ätas	1.275225012	1	2.216011369	0	2.087511826	1.263382283	2.187115111	0.274834437	2.058656575	1	0	1.959261014	2.216011369	0.959318827	2.07615894	1.43628612	1.252601703	3.19734974	1.61431316
12	V-SBJV-ACT-PRS - äte	0.964497041	0.952287066	1.921529968	1.959261014	2.905258171	1.693611987	1.890772871	1.70724775	2.797252487	0.952249408	1.959261014	0	1.820189274	1	2.814779725	1.960962145	2.901942207	3.229306931	2.058144696
13	V-SBJV-ACT-PST - åte	2.07495069	2.077287066	0.128154574	2.216011369	1.116532449	2.086750789	0.173895899	2.091425865	1.036949313	2.077348066	2.216011369	1.820189274	0	2.780672667	1.004263382	2.092665615	2.200378967	1.892277228	1.351531292
14	V-SBJV-PASS-PRS - ätes	1.968261487	1.957366177	2.86215064	0.959318827	1.906811731	1.72382757	2.833254382	0.952223273	1.877956481	1.957325747	0.959318827	1	2.780672667	0	1.814569536	2.129796305	1.94512772	4.153809749	2.231508165
15	V-SBJV-PASS-PST - åtes	3.069635244	3.064424443	1.117479867	2.07615894	0.112582781	2.843202274	1.150165798	2.052980132	0.145222327	3.064485538	2.07615894	2.814779725	1.004263382	1.814569536	0	2.986736144	2.078524125	2.834358732	2.226705091
16	V-V.CVB-ACT - ätit	1.236291913	1.27011041	2.092665615	1.43628612	2.986736144	1.413643533	2.107255521	1.258171483	2.980104216	1.269534333	1.43628612	1.960962145	2.092665615	2.129796305	2.986736144	0	1.001421127	3.582178218	1.349312028
17	V-V.CVB-PASS - ätits	2.210800568	2.206537186	2.200378967	1.252601703	2.078524125	2.251539555	2.213169114	1.209555345	2.0884579	2.205784732	1.252601703	2.901942207	2.200378967	1.94512772	2.078524125	1.001421127	0	3.575958353	2.288184438
18	V-V.PTCP-PRS - ätande	3.456418384	3.196039604	2.00950495	3.19734974	2.844297208	3.486732673	2.011089109	3.454803597	2.833885471	3.196195006	3.19734974	3.229306931	1.892277228	4.153809749	2.834358732	3.582178218	3.575958353	0	2.467584369
19	V-V.PTCP-PST - äten	1.408969805	1.461606747	1.46560142	1.61431316	2.323246878	1.545051043	1.485130937	1.426512968	2.323727185	1.461128387	1.61431316	2.058144696	1.351531292	2.231508165	2.226705091	1.349312028	2.288184438	2.467584369	0

Figure 7: Average edit distances for the Swedish verb paradigm. Values range from 0 to 4.153. Darker red shades indicate closer relationships between feature sets, while darker ball-blue shades represent greater differences.



	Features	V-IMP-ACT - ät	V-IND-PL-ACT-PRS - äta	V-IND-PL-ACT-PST - åto	V-IND-PL-PASS-PRS - ätas	V-IND-PL-PASS-PST - åtos	V-IND-SG-ACT-PRS - äter	V-IND-SG-ACT-PST - åt	V-IND-SG-PASS-PRS - äts/ätes	V-IND-SG-PASS-PST - åts	V-NFIN-ACT - äta	V-NFIN-PASS - ätas	V-SBJV-ACT-PRS - äte	V-SBJV-ACT-PST - åte	V-SBJV-PASS-PRS - ätes	V-SBJV-PASS-PST - åtes	V-V.CVB-ACT - ätit	V-V.CVB-PASS - ätits	V-V.PTCP-PRS - ätande	V-V.PTCP-PST - äten
1	V-IMP-ACT - ät	1	6	52	7	54	6	47	5	48	6	7	6	51	6	53	37	38	66	80
2	V-IND-PL-ACT-PRS - äta	6	1	55	1	51	5	61	7	55	1	1	7	54	6	50	36	34	48	88
3	V-IND-PL-ACT-PST - åto	54	54	1	49	4	56	11	57	13	54	49	57	2	52	5	33	34	116	70
4	V-IND-PL-PASS-PRS - ätas	7	1	51	1	51	6	55	7	55	1	1	7	50	6	50	35	34	33	79
5	V-IND-PL-PASS-PST - åtos	55	50	4	49	1	54	13	54	10	50	49	52	5	52	2	34	31	100	67
6	V-IND-SG-ACT-PRS - äter	6	6	63	6	59	1	65	8	61	6	6	7	53	6	52	39	38	65	93
7	V-IND-SG-ACT-PST - åt	47	55	11	50	13	57	1	50	4	55	50	58	11	53	13	40	41	124	77
8	V-IND-SG-PASS-PRS - äts/ätes	5	8	55	7	53	7	49	1	47	8	7	6	54	5	52	36	36	51	76
9	V-IND-SG-PASS-PST - åts	48	51	13	50	10	56	4	47	1	51	50	53	13	53	10	38	38	103	72
10	V-NFIN-ACT - äta	6	1	55	1	51	5	61	7	55	1	1	7	54	6	50	36	34	48	88
11	V-NFIN-PASS - ätas	7	1	51	1	51	6	55	7	55	1	1	7	50	6	50	35	34	33	79
12	V-SBJV-ACT-PRS - äte	6	7	64	6	57	6	66	5	58	7	6	1	54	1	50	35	32	62	88
13	V-SBJV-ACT-PST - åte	53	53	2	48	5	55	11	56	13	53	48	56	1	51	4	33	34	124	73
14	V-SBJV-PASS-PRS - ätes	7	7	56	6	57	6	57	5	58	7	6	1	49	1	50	34	33	47	79
15	V-SBJV-PASS-PST - åtes	54	49	5	48	2	53	13	53	10	49	48	51	4	51	1	34	31	104	68
16	V-V.CVB-ACT - ätit	38	40	33	36	34	40	39	38	39	40	36	40	33	35	34	1	3	100	60
17	V-V.CVB-PASS - ätits	38	37	36	36	32	40	41	36	37	37	36	39	35	36	32	3	1	88	54
18	V-V.PTCP-PRS - ätande	66	48	137	33	115	68	142	57	119	48	33	64	128	48	108	106	91	1	39
19	V-V.PTCP-PST - äten	81	90	80	79	74	91	85	78	79	90	79	87	73	76	68	56	49	34	1

Figure 8: Edit Script scores for the Swedish verb paradigm. Values range from 1 to 142. Darker purple shades indicate fewer unique character sets (closer relationships), while darker air-force-blue shades reflect greater variation.



	Features	V-IMP-ACT - ät	V-IND-PL-ACT-PRS - äta	V-IND-PL-ACT-PST - åto	V-IND-PL-PASS-PRS - ätas	V-IND-PL-PASS-PST - åtos	V-IND-SG-ACT-PRS - äter	V-IND-SG-ACT-PST - åt	V-IND-SG-PASS-PRS - äts/ätes	V-IND-SG-PASS-PST - åts	V-NFIN-ACT - äta	V-NFIN-PASS - ätas	V-SBJV-ACT-PRS - äte	V-SBJV-ACT-PST - åte	V-SBJV-PASS-PRS - ätes	V-SBJV-PASS-PST - åtes	V-V.CVB-ACT - ätit	V-V.CVB-PASS - ätits	V-V.PTCP-PRS - ätande	V-V.PTCP-PST - äten
1	V-IMP-ACT - ät	0.88	0.87	0.82	0.86	0.81	0.87	0.82	0.87	0.82	0.86	0.87	0.85	0.82	0.85	0.82	0.82	0.82	0.85	0.81
2	V-IND-PL-ACT-PRS - äta	0.73	0.83	0.7	0.82	0.69	0.73	0.7	0.71	0.69	0.83	0.82	0.81	0.7	0.82	0.69	0.69	0.68	0.81	0.69
3	V-IND-PL-ACT-PST - åto	0.75	0.74	0.84	0.74	0.83	0.75	0.83	0.74	0.82	0.75	0.75	0.75	0.84	0.73	0.83	0.8	0.8	0.77	0.8
4	V-IND-PL-PASS-PRS - ätas	0.72	0.82	0.69	0.83	0.68	0.71	0.69	0.7	0.69	0.82	0.83	0.81	0.69	0.81	0.69	0.7	0.7	0.82	0.7
5	V-IND-PL-PASS-PST - åtos	0.73	0.73	0.78	0.72	0.8	0.73	0.79	0.73	0.79	0.72	0.73	0.71	0.79	0.71	0.8	0.78	0.77	0.74	0.78
6	V-IND-SG-ACT-PRS - äter	0.82	0.81	0.77	0.81	0.75	0.83	0.78	0.79	0.76	0.81	0.81	0.82	0.76	0.81	0.75	0.77	0.77	0.8	0.75
7	V-IND-SG-ACT-PST - åt	0.76	0.75	0.81	0.74	0.79	0.75	0.82	0.75	0.8	0.75	0.75	0.75	0.81	0.74	0.8	0.77	0.77	0.75	0.79
8	V-IND-SG-PASS-PRS - äts/ätes	0.8	0.8	0.76	0.8	0.76	0.79	0.76	0.8	0.75	0.81	0.8	0.78	0.74	0.77	0.75	0.75	0.75	0.77	0.73
9	V-IND-SG-PASS-PST - åts	0.72	0.7	0.78	0.71	0.78	0.71	0.79	0.7	0.79	0.7	0.71	0.71	0.79	0.71	0.78	0.75	0.76	0.72	0.75
10	V-NFIN-ACT - äta	0.72	0.84	0.7	0.82	0.69	0.72	0.7	0.71	0.7	0.84	0.82	0.82	0.7	0.81	0.7	0.7	0.7	0.81	0.7
11	V-NFIN-PASS - ätas	0.69	0.79	0.66	0.78	0.65	0.68	0.66	0.66	0.66	0.79	0.8	0.76	0.67	0.77	0.65	0.69	0.68	0.76	0.68
12	V-SBJV-ACT-PRS - äte	0.71	0.8	0.68	0.79	0.68	0.71	0.68	0.69	0.68	0.8	0.8	0.81	0.69	0.8	0.68	0.7	0.69	0.78	0.67
13	V-SBJV-ACT-PST - åte	0.77	0.77	0.83	0.76	0.83	0.76	0.82	0.75	0.81	0.77	0.76	0.74	0.84	0.74	0.83	0.8	0.79	0.76	0.81
14	V-SBJV-PASS-PRS - ätes	0.69	0.77	0.68	0.77	0.68	0.69	0.67	0.68	0.67	0.77	0.77	0.77	0.67	0.77	0.66	0.67	0.67	0.75	0.68
15	V-SBJV-PASS-PST - åtes	0.67	0.66	0.74	0.66	0.73	0.65	0.74	0.67	0.74	0.68	0.67	0.66	0.75	0.66	0.75	0.69	0.69	0.66	0.71
16	V-V.CVB-ACT - ätit	0.76	0.76	0.8	0.76	0.79	0.77	0.8	0.76	0.79	0.76	0.76	0.76	0.8	0.75	0.79	0.82	0.81	0.77	0.8
17	V-V.CVB-PASS - ätits	0.71	0.73	0.74	0.72	0.75	0.71	0.75	0.73	0.74	0.72	0.73	0.7	0.73	0.71	0.75	0.78	0.77	0.71	0.73
18	V-V.PTCP-PRS - ätande	0.65	0.74	0.64	0.74	0.63	0.65	0.64	0.62	0.63	0.75	0.74	0.73	0.65	0.72	0.63	0.64	0.63	0.8	0.68
19	V-V.PTCP-PST - äten	0.71	0.7	0.74	0.69	0.73	0.69	0.75	0.7	0.73	0.7	0.69	0.7	0.74	0.69	0.74	0.74	0.73	0.75	0.8

Figure 9: Reinflection Accuracy scores for the Swedish verb paradigm. Values range from 0.62 to 0.88. Darker teal shades indicate higher accuracy, while darker pink shades reflect lower performance.

Accelerating Large Language Model Pretraining via LFR Pedagogy: Learn, Focus, and Review

Neha Prakriya, Jui-Nan Yen, Cho-Jui Hsieh, Jason Cong
University of California, Los Angeles
{nehaprakriya, juinanyen, chohsieh, cong}@cs.ucla.edu

Abstract

We introduce an effective and scalable data selection technique to accelerate the pretraining of large language models (LLMs). Given the variation in quality and informativeness of web-scale corpora, we present the Learn-Focus-Review (LFR) paradigm-a dynamic training approach that adapts to the model's learning progress. Inspired by human learning techniques like spaced repetition, LFR tracks the model's learning performance across data instances and prioritizes revisiting challenging and diverse regions of the dataset that are more prone to being forgotten, enabling better retention and more efficient learning. Through experiments spanning over 2200 GPU hours, we show that LFR significantly enhances data efficiency in pretraining while improving downstream performance across commonsense reasoning, question answering, problem-solving, language modeling, and translation tasks. LFR consistently achieves lower perplexity and higher accuracy using just 5%–19% of the training tokens as models trained on the full dataset. Notably, LFR matches the performance of industry-standard Pythia models with up to 2× the parameter count while requiring only 3.2% of the training tokens. Unlike prior work on data selection, LFR models are Chinchilla-optimal demonstrating the effectiveness of our training methodology.

1 Introduction

LLMs have achieved remarkable success in understanding and generating human language. This success is driven by the ever-increasing model parameter sizes which require web-scale training datasets like SlimPajama (Soboleva et al., 2023), CommonCrawl (Penedo et al., 2023; Raffel et al., 2023), Pile (Gao et al., 2020), and OpenWebText (Radford et al., 2019; ope), leading to unsustainable training costs. Between 2016 and 2023, model training costs have skyrocketed by a factor of 750× every two years (Gholami et al., 2024), while GPU memory has scaled at a much slower pace of 2× every two years. For example, pretraining the GPT-4 model (OpenAI et al., 2024) was estimated to have cost around $100M USD over a period of 3-4 months using 25k NVIDIA A100 GPUs (gpt).

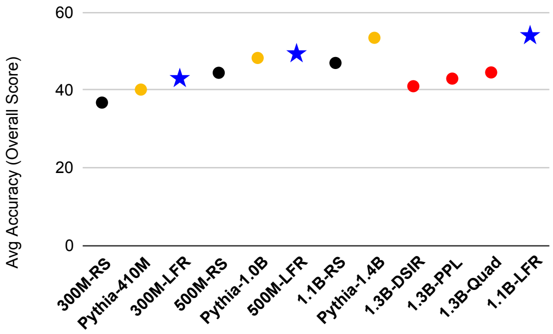


Figure 1: Average accuracy norm across commonsense reasoning, problem-solving, world knowledge, and reading comprehension tasks. Across model sizes (300M–1.1B), LFR (stars) outperforms full-dataset training (RS in black circles) by 6%, Pythia (yellow circles) by 1.5%, and Quad (Zhang et al., 2024) (red circle) by 9%, using only 3–6% and 65% of the training tokens of Pythia and Quad, respectively. Notably, Pythia and Quad have larger parameter counts. See Section 5 for details.

As such, a key challenge for unlocking the next generation of language models is to significantly reduce training costs while retaining or improving downstream task performance.

Data quality and selection play a key role in the development of cost-effective and high-performance models (Hoffmann et al., 2022; Brown et al., 2020; Tirumala et al., 2023; Abbas et al., 2023; lla, 2024). In fact, DeepSeek-V3 technical report (DeepSeek-AI et al., 2025) and the Llama 3.1 Technical Report (lla, 2024) highlight the importance of data quality through curated data mixes and sophisticated data preprocessing pipelines to minimize redundancy and maximize

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 268–290
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

corpus diversity.

Recent work on data selection for pretraining has achieved great strides in reducing the overall training time. Methods like D4 (Tirumala et al., 2023), SemDeDup (Abbas et al., 2023), MiniPile (Kaddour, 2023; min), DSIR (Xie et al., 2023), and perplexity-based filtering (Marion et al., 2023; Chen et al., 2024; Muennighoff et al., 2023) rely on similarity metrics, clustering, or perplexity to filter data. However, data importance evolves throughout training and depends on model architecture, making static filtering inherently limited in effectiveness. While (Zhang et al., 2024) employ a dynamic data selection approach using the multi-armed bandit technique, they select 30B tokens from the SlimPajama dataset to train a 1.3B parameter model. However, according to the Chinchilla scaling laws (Hoffmann et al., 2022), this token count exceeds the optimal range for models of this size, suggesting that their selected subsets may contain redundant or lower-quality data. Other studies propose leveraging state-of-the-art (SOTA) pretrained LLMs like GPT-4 (Wettig et al., 2024) or proxy models, as seen in MATES (Yu et al., 2024) and RHO-1 (Lin et al., 2024), to assess data quality for a target model. However, these approaches rely on existing separately trained models, which may introduce a mismatch between the data needed for optimal convergence and the data selected.

We address the high training cost of LLMs and the shortcomings of existing data selection methods by drawing inspiration from spaced repetition (Smolen et al., 2016a; spa). This scientifically proven technique enhances retention by strategically presenting information at optimal intervals, ensuring that the most relevant data is introduced at the right time for efficient learning. Building on this foundation, we propose the *Learn-Focus-Review* (LFR) training paradigm. Figure 1 displays the overall efficacy of LFR. Our work offers the following contributions:

1. Profile LLM pretraining to observe multiple descent behavior in 25-78% of the training tokens from web-scale corpuses, which are forgotten multiple times during training.
2. Develop a Learn-Focus-Review (LFR) training pipeline that dynamically gauges the LLM's learning pace, focusing on complex data blocks while regularly reviewing all data blocks to prevent forgetting.
3. Conduct over 2200 GPU hours of training experiments using the AMD MI250, AMD MI210, and AMD MI100 GPUs. We pretrain Llama and GPT models of varying sizes from scratch on the SlimPajama (627B) and OpenWebText (9B) datasets and evaluate them on several downstream tasks from the commonsense reasoning, question-answering, problem solving, language modeling, and translation domains.
4. LFR results in significantly lower perplexity and higher accuracy compared to baseline models trained on the full dataset, achieving these improvements by training on just 5-19% of the training tokens used by the baseline. All our models are Chinchilla-optimal.
5. LFR outperforms the performance on 70% of tasks of the Pythia models with up to 2× the parameter count while requiring only 3-6% of the training tokens.
6. LFR outperforms prior state-of-the-art data selection work by 9-13% in downstream task accuracy while using only 65% of the training tokens.
7. Observe that LLMs first learn conversational and anecdotal data, before being able to retain factual, instructional, and coding language information in long-term memory.

In the following sections, we examine prior works on efficient LLM pretraining before diving deeper into our proposed training strategies and design decisions.

2 Related Work

Prior works on efficient pretraining of LLMs using data selection have primarily focused on using distance metrics and clustering techniques. Tirumala et al. (2023) proposes D4, which deduplicates and selects cluster centers in the embedding space generated by pretrained models. SemDeDup (Abbas et al., 2023) prunes semantic duplicates using pretrained models. It can successfully prune 50% of the training data with minimal performance loss. MiniPile (Kaddour, 2023; min) uses the pretrained E5-Large (Wang et al., 2024) model to embed documents in the Pile dataset and clusters them to select a smaller pretraining corpus of $\sim$6GB. DSIR (Xie et al., 2023) proposes selecting subsets from large unlabeled datasets through importance resampling to match the distribution of the target dataset. However, considering the high cost of training, it is unsustainable to sample a new subset and pretrain the LLM from scratch for every new downstream task.

More recently, perplexity-based and influence function-based filtering techniques have been proposed (Marion et al., 2023; Lin et al., 2024; Muennighoff et al., 2023; Chen et al., 2024; Wettig et al., 2024; Yu et al., 2024), which use proxy models to obtain perplexity/influence scores for different data points and assess sample importance. However, these methods require an additional pretrained model, increasing computational overhead. Moreover, if the proxy model has a different architecture from the target model, its assessment of data importance may not accurately transfer, leading to suboptimal data selection and inefficiencies in training.

The Chinchilla scaling laws (Hoffmann et al., 2022) derive an optimal model size–to–training tokens ratio for fixed compute budgets, finding that parameters and data should scale roughly 1:1. We observe that several of the prior works discussed in this Section do not incorporate Chinchilla scaling laws (Hoffmann et al., 2022) into their data selection strategies, leading to suboptimal filtering of web-scale corpora and potential overtraining. For example, Zhang et al. (2024) present Quad, a data selection method which calculates influence scores to measure a data point's impact on model performance. They select 30B tokens from the SlimPajama dataset (627B) for their 1.3B model and continual pretraining of the 7B model. This indicates that the models have been overtrained or trained on redundant tokens.

3 Problem Formulation and Profiling

3.1 LLM Pretraining Objective

Given an LLM model parameterized by weights θ and a web-scale dataset D, we first tokenize all documents in the dataset and obtain context-length-sized sequences of tokens, called data blocks, s_i such that the training corpus becomes $D = \{s_1, s_2, s_3, ...s_n\}$. For the SlimPajama and OpenWebText datasets used in this paper, the context length is 1024 tokens, with a total of 627B and 9B tokens, respectively. Given one such sequence of tokens or data block, $s_i = \{x_1, x_2, ...x_n\}$, the training objective is to autoregressively predict the next M tokens:

$$p_\theta(y \mid x) = \prod_{i=1}^{M} p_\theta(y_i \mid y_{1:i-1}, x). \quad (1)$$

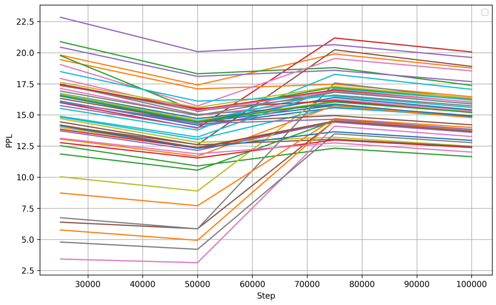


Figure 2: PPL trajectories of data samples from the SlimPajama dataset as processed by the Llama-300M model, focusing on a subset of 50 samples for clarity. Notably, 78.5% of the samples exhibit this behavior, characterized by multiple descent patterns rather than a steady decline. This indicates that the model frequently forgets and relearns data during training, highlighting inefficiencies in traditional training dynamics

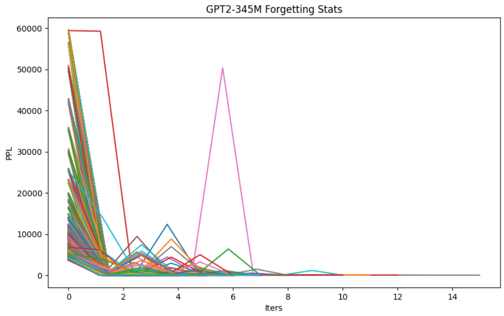


Figure 3: PPLs of data samples being forgotton by the GPT2-345M model on the OpenWebText dataset. This multi-descent behavior is exhibited by 20% of the data.

3.2 Observations from Training on Randomly Sampled Data

In order to better understand the drawbacks of this traditional training technique, we profile the pretraining process for the Llama and GPT models. The training hyperparameters and model configurations are provided in the Appendix A. Similarly to Marion et al. (2023), we use *perplexity* as a metric to monitor the training progress. Given a token sequence $s_i = \{x_1, x_2, ..., x_n\}$ from the dataset D, perplexity is computed as:

$$PPL(s_i) = \exp\left(\frac{1}{|s_i|}\sum_{x_j \in s_i} NLL(x_j)\right), \quad (2)$$

where $NLL(x_j)$ is the negative log likelihood of token x_j computed as follows:

$$NLL(x_j) = -\log P(x_j \mid x_{<j}; \theta). \qquad (3)$$

Using this metric, models exhibiting lower perplexities are considered better since this indicates a high probability of selecting text closest to the raw dataset.

The observed PPL values associated with each data block are classified as one of the following:

1. *Learned*: recorded perplexities monotonically decrease.
2. *Unlearned*: recorded perplexities monotonically increase.
3. *Forgotten*: recorded perplexities first increase and then decrease. Such an upward and downward trend may repeat any number of times during training.

Based on this classification, we observe that 78.5% of the data blocks are forgotten at least once in the Llama model (Figure 2), compared to 25% in the GPT model (Figure 3). We hypothesize that more data blocks are frequently forgotten in the Llama model due to the higher complexity and challenge posed by the SlimPajama dataset, as opposed to the OpenWebText dataset. It is important to note that the SlimPajama dataset is an aggregation of seven datasets, including sources such as GitHub, Wikipedia, and CommonCrawl. In fact, of the data blocks that are forgotten, 82% are forgotten multiple times during training, i.e., they display *multiple descent behavior* (Figure 3). Xia et al. (2022) reported a double-descent behavior for the OPT models (Zhang et al., 2022), and our above experiment further demonstrates that the "forgetting" can happen multiple times in LLM pretraining.

4 LFR Training Methodology

Based on our profiling observations in Section 3.2 we propose to replace traditional autoregressive language modeling methods with Spaced Repetition (Tabibian et al., 2019). Spaced Repetition is an evidence-based learning method proven to improve information retention and learning pace in humans (Smolen et al., 2016b). In this technique, challenging pieces of information are reviewed more often, at regular intervals, and easier pieces of information are presented to the learner less often. Our algorithm is detailed in Algorithm 1. We pretrain our models with a combination of the following three steps:

1. **Learn**: Initially, we present the model with the entire dataset and train on randomly selected data blocks for p_1 steps, as normally seen in the traditional approach (line 1 in Alg. 1). p_1 can be configured by the user based on the available compute budget, model, and dataset. In single-epoch training (lines 3-7 in Alg. 1), we measure the perplexities (PPLs) of all data samples in the training set and cluster the data embeddings (inputs to the model's last layer). For multi-epoch training (lines 8-11 in Alg 1), we record the perplexities for all data blocks during the p_1 steps. Depending on the training style (single or multi-epoch), we either pass the clustered embeddings and PPL values or the PPL values observed during training to the next step. The following two phases can be repeated up to $reps$ times, depending on the available compute budget.
2. **Focus**: We provide two variations of the Focus stage based on the number of training epochs.
   (a) Single-epoch training: We discard s_1% of the clusters (line 6 in Alg 1). Within the retained clusters, we perform weighted sampling from sub-clusters, prioritizing regions of the retained clusters which the model finds most challenging (line 7 in Alg. 1). Sub-clusters with higher PPL are assigned greater sampling weights, enabling a hierarchical focus on the most critical regions. For instance, during Llama training, GitHub code emerged as the most challenging cluster. Within this cluster, the Focus stage further emphasizes sampling from C++ code, which proved more difficult for the model, over Python code. In this phase of training, we restrict the weighted sampling of data points to this reduced subset for p_2 steps. s_1 and p_2 are user-controlled hyperparameters.
   (b) Multi-epoch training: We discard s_1% of the data blocks (line 10 in Alg. 1) with the lowest PPL values. In doing so, we provide a mechanism for shifting the model's focus towards learning

data blocks that were determined to be difficult.

3. **Review**: Next, we reintroduce all of the removed data blocks and train the model by randomly sampling from the entire corpus for p_3 steps (line 13 in Alg. 1). This ensures that we allow the model to review and revisit data blocks which it may have forgotten.

Algorithm 1 LFR Training Methodology

```
Require: Training dataset $D$, model $M$ with initial parameters $\theta_0$, hyperparameters $p_1$, $s_1$, $p_2$, $p_3$, $reps$, and $epochs$.
Ensure: Minimization of Equation 3.
 1: $PPLs, \theta_{p_1} \leftarrow$ Learn($\theta_0, D, p_1$)
 2: for $r = 1, 2, \ldots, reps$ do
 3:     if $epochs == 1$ then
 4:         $D_k \leftarrow Cluster(D)$
 5:         $Sort(PPLs, D_k)$
 6:         $S_{sub} \leftarrow (1 - s_1) \times D_k$
 7:         $S_1 \leftarrow sample(S_{sub}, PPLs)$
 8:     else
 9:         $Sort(PPLs, D)$
10:         $S_1 \leftarrow (1 - s_1) \times D$
11:     end if
12:     $\theta_{p_2} \leftarrow$ Focus($\theta_{p_1}, S_1, p_2$)
13:     $PPLs, \theta_{p_3} \leftarrow$ Review($\theta_{p_2}, D, p_3$)
14: end for
    $Return\ \theta$
```

Our training strategy is simple, intuitive and human-like. It gives the model an opportunity to learn from all of the data, prioritize and relearn forgotten data points, and review data samples from harder regions of the dataset more frequently than they would have been using random sampling. While the static clustering-based techniques (Tirumala et al., 2023; Abbas et al., 2023; Kaddour, 2023) presented in Section 2 allow for accelerated training, they are not designed to suit the multi-descent training dynamics observed in Section 4 and require pretrained model embeddings to calculate distance metrics. Furthermore, prior methods including perplexity-based pruning methods (Marion et al., 2023) are static. Sections 5.4 and the Appendix characterize the data blocks found easy and hard by the LLM, and demonstrate why static, clustering-based data selection methods achieve poor downstream task performance. Lastly, our approach does not require any pretrained models to obtain embeddings. Our focused training strategy allows the model to absorb harder information (data blocks with higher perplexity) faster, by presenting them more number of times.

5 Evaluation

In this section, we present a comprehensive evaluation of LFR. We pretrain the Llama models of sizes 300M, 500M, and 1.1B and the GPT models (Radford et al., 2019) of various sizes between 124M and 1.5B parameters. We use the SlimPajama (Soboleva et al., 2023) (627B) and OpenWebText dataset (ope) (9B) and train from scratch using 4 AMD MI100, 4 AMD MI210 GPUs, and 8 AMD MI250 GPUs. Our pretraining experiments utilize a fully sharded data parallel (FSDP) approach. All model configurations and training hyperparameters of our experiments are detailed in the Appendix A.

Our models and all baselines are evaluated across a diverse set of downstream tasks spanning multiple domains: (1) Commonsense reasoning (HellaSwag, Winogrande, PIQA), (2) General knowledge (ARC_C, ARC_E, MMLU, Natural Questions), (3) Reading comprehension (OpenbookQA, BoolQ), (4) Language modeling (WikiText-2, WikiText-103, LAMBADA, 1BW), and (5) Translation (WMT-14). Performance results and comparisons to prior state-of-the-art methods are detailed in Sections 5.3.

Section 5.4 analyzes the impact of the Focus and Review stages and the data LFR prioritizes in SlimPajama. The Appendix provides examples, details on retained/dropped data across models, evidence that LLMs learn instructions and code after facts and anecdotes, and a sensitivity study on LFR hyperparameters.

5.1 LFR Configuration

We pretrain the Llama models for 100k steps, using 9.8B tokens for the 300M and 500M models and 19.6B tokens for the 1.1B model, following the Chinchilla scaling law (Hoffmann et al., 2022) to ensure optimal data utilization. First, we Learn for 20k steps ($p_1 = 20k$). Next, we cluster the data and discard 57.2% of the clusters, retaining only the 3 most challenging clusters out of 7 based on their PPL values ($s_1 = 50$). We chose this configuration based on our limited pretraining budget and profiling in Section 3.2, which showed that 78.5% and 25% of data samples are forgotten at least once during training for the Llama and GPT models, respectively. We then apply the Focus

Model	Tokens	Arc_C	Arc_E	Boolq	HS	OBQA	Piqa	WG	Avg
300M-RS	50B	17.29	39.06	33.17	32.3	28.83	58.36	48.54	36.79
Pythia-410M	300B	20.1	**44**	40	**35.82**	29.59	61.8	49.7	40.14
300M-LFR	9.8B	**23.61**	39.52	**54.86**	35.44	**30.56**	**63.21**	**53.88**	**43.01**
500M-RS	50B	25.1	43.7	53.7	36.5	32.6	65.1	52.2	44.47
Pythia-1.0B	300B	27.05	48.99	**60.83**	47.16	**31.4**	**69.21**	53.43	48.29
500M-LFR	9.8B	**28.11**	**52.89**	58.72	**50.65**	31.1	68.66	**55.72**	**49.4**
1.1B-RS	50B	27.31	50.27	60.58	38.11	31.11	66.67	54.99	47
Pythia-1.4B	300B	30.1	61.7	62.11	**55.18**	30.2	72	**63.1**	53.48
DSIR	30B	20.14	49.28	61.41	30.89	16.2	61.17	47.99	41.01
PPL	30B	20.82	45.41	58.35	35.92	18.8	66.89	54.62	42.97
1.3B-Quad	30B	20.99	52.27	62.14	34.41	20.00	70.04	52.09	44.56
1.1B-LFR	19.6B	**29.18**	**63.47**	**62.23**	54.27	**34.89**	**73.29**	61.12	**54.06**

Table 1: Zero-shot performance (acc_norm for all except Winogrande and Boolq which use acc) on downstream tasks evaluated using LLM Evaluation Harness (Gao et al., 2024). RS refers to the random sampling baseline, HS refers to HellaSwag, and WG refers to Winogrande. The model with the highest performance (measured by acc_norm) is highlighted in bold. Notably, LFR models are trained using only 3.2-6% of the tokens required to train Pythia models of comparable size, yet they achieve higher accuracy in 70% of cases. Additionally, LFR models consistently outperform the random sampling baseline by a large margin, despite being trained on 19.6% of the pretraining tokens.

stage for 60k steps ($p_2 = 60k$), prioritizing the retained high-PPL clusters. It takes <10min to cluster which can be hidden by the high training latency. We provide a detailed analysis on the hierarchical clustering and the data points found easy and difficult in Section 5. Lastly, we Review the entire dataset for the last 20k steps ($p_3 = 20k$). In the case of the GPT models, we Learn for 1 epoch ($p_1 = 1$), Focus on 50% of the data for 1 epoch ($s_1 = 50, p_2 = 1$), Review the entire dataset for another epoch ($p_3 = 1$), and Focus on 30% of the data for 5 epochs ($reps = 2, s_2 = 70, p_4 = 1$). This configuration is chosen based on the findings in Section 3.2. Figure 3 reveals that forgotten samples are typically forgotten multiple times, requiring an average of 4 presentations to be learned. For GPT, we use the first three phases to identify these samples and allocate 5 epochs focusing on 30% of them in the final phase to ensure long-term retention.

These configurations are tunable based on the available pretraining budget and the optimal tokens estimated through the Chinchilla scaling laws. Furthermore, we test LFR's sensitivity to hyperparameters p_1, s_1, p_2, p_3, and $reps$ in the Appendix A.

5.2 Baselines

We evaluate the models pretrained using LFR with a comprehensive set of prior works and industry-standard checkpoints. They include:

1. Industry-standard models: We compare the Llama models trained through LFR with Pythia models (Biderman et al., 2023) of up to 2× the size obtained from EleutherAI's Huggingface[1]. These models have been trained on 300B tokens while the LFR models were trained on 9.8B-19.6B tokens (3.2-6% of the tokens). We compare the GPT models pretrained through LFR for 40k iterations with the same GPT architectures pretrained by OpenAI [2] for 800k iterations. We use the same batch size as these models (Refer to the Appendix for details) by adjusting the gradient accumulation steps and the per-device batch size.

2. Random Sampling: while the previous baselines ensures that we compare with industry-standard models, we also train and compare LFR against the same models pretrained using random sampling with 5.10× and 20× more tokens than LFR for the Llama and GPT models respectively. This baseline allows us to test LFR against the same models trained through traditional autoregressive techniques.

3. Prior works: We compare our training methodology with the models trained through the

---

[1] https://huggingface.co/models?other=pythia

[2] https://huggingface.co/openai-community

current state-of-the-art data selection methods such as Quad (Zhang et al., 2024), static-PPL based filtering (Marion et al., 2023), DSIR (Xie et al., 2023), and MiniPile (Kaddour, 2023) in Section 5.3.

5.3 Performance on Downstream Tasks

We evaluate Llama models trained with LFR on commonsense reasoning, general knowledge QA, and reading comprehension, comparing accuracy norms with baselines in Table 1. LFR outperforms random sampling (RS) by 6% while using 2.4×–5× fewer training tokens and improves accuracy over Pythia by 1.5% despite using only 3.2–6% of the tokens. Compared to prior SOTA data selection, LFR achieves greater dataset pruning while improving downstream performance. Notably, their models are over-trained per Chinchilla laws, highlighting suboptimal data selection.

We test the GPT models on language modeling tasks and compare with the OpenAI baseline in Table 2 by measuring the PPL. Note that our models are trained on 5% of the training tokens as compared with the OpenAI models, further validating that data quality is more important than quantity. We find that the PPL reduction obtained by LFR increase as the dataset size increases (from WikiText-2 to 1BW). Also, smaller models show a larger PPL reduction by using LFR than larger models. On average, using our approach, perplexity was reduced by 4.92, 3.26, 2.17, and 1.40 for the GPT 124M, 345M, 774M, and 1.5B models, respectively.

We also test the LFR-trained models on standard benchmarks from the translation, question-answering, world knowledge, and problem solving domains in Table 3. LFR models trained with 20× fewer training iterations achieves better performance than models trained using random sampling. Details of each of the datasets is provided in the Appendix A.

5.4 Ablation Study

In this section, our goal is to understand the impacts of the Focus and Review stages of LFR and exploring more aggressive data selection strategies by varying the hyperparameters p_1, s_1, p_2, p_3, and $reps$.

5.4.1 Impact of Focus

Consider training the Llama 300M parameter model on the SlimPajama dataset, which comprises of seven sub-datasets sourced from CommonCrawl, Github, C4, Books, Wikipedia, StackExchange, and ArXiv. During the Focus stage, LFR employs weighted sampling from the three most challenging clusters while discarding clusters with the lowest perplexity (PPL). Additionally, within the retained clusters, LFR performs hierarchical sampling by prioritizing regions with higher PPL, further refining the data selection process. LFR classifies the Github, StackExchange, and ArXiv clusters as more challenging at 20k iterations, than the other four data sources.

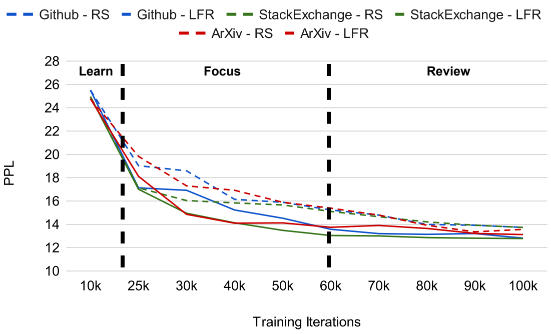


Figure 4: PPL values are tracked at different training iterations for the clusters identified as challenging and prioritized during the Focus stage of LFR. The dotted line represents the PPL values for the same clusters when trained with random sampling (RS). Notably, LFR facilitates accelerated learning of these challenging data points between 20k and 60k iterations (the Focus stage), whereas random sampling consistently results in higher PPL values throughout.

Figure 4 illustrates the training dynamics of challenging data points. LFR (solid line) accelerates learning of these harder examples compared to random sampling (dotted line), ensuring complex information is learned earlier, which drives the performance gains in Table 1. In the Review stage, discarded clusters (CommonCrawl, C4, Books, Wikipedia) are reintroduced, bringing LFR and random sampling closer together. However, LFR retains the benefits of the Focus stage by performing marginally better on the challenging sections.

5.4.2 Impact of Review

Next, we analyze the impact of the Review phase on data points deemed simple and discarded during Focus. Unlike prior data selection methods, LFR reintroduces these samples, preventing catastrophic forgetting. Figure 5 highlights the importance of Review by plotting PPL values for easy data points under LFR (solid line) and random sampling (dot-

Model	WikiText-2	WikiText-103	LAMBADA	1BW
124M-OpenAI (800k iters)	22.1	31.58	18	39.18
124M-RS (40k iters)	23.32	23.42	17.71	39.49
124M-LFR (40k iters)	19.81	22.49	16.61	32.27
345M-OpenAI (800k iters)	19.82	22.05	14.26	29.95
345M-RS (40k iters)	21.11	21.8	14.84	30.66
345M-LFR (40k iters)	16.31	17.48	13.7	25.52
774M-OpenAI (800k iters)	15.93	18.53	13.74	26.52
774M-RS (40k iters)	16.71	18.89	14.10	28.56
774M-LFR (40k iters)	15.11	14.58	12.51	23.83
1.5B-OpenAI (800k iters)	13.80	16.59	12.15	23.87
1.5B-LFR (40k iters)	13.10	14.37	11.23	22.09

Table 2: PPL results for language modeling datasets across model sizes. Here, N-OpenAI refers to the OpenAI baseline (trained for 800k iterations), N-RS refers to the random sampling baseline (trained for 40k iterations), and N-LFR refers to our proposed training pedagogy (trained for 40k iterations), where N is the number of model parameters.

Model	Iters	WMT (BLEU)	NQ (Acc)	MMLU				
				STEM (Acc)	HM (Acc)	SS (Acc)	Other (Acc)	Avg. (Acc)
1.5B OpenAI	800k	11.5	4.1	24.5	24.8	**24.0**	**27.8**	25.3
1.5B LFR	40k	**11.8**	**4.61**	**26.1**	**27.2**	23.8	25.1	**25.5**

Table 3: LFR-trained GPT models evaluated on translation (WMT-14 (wmt)), question-answering (Natural Questions (Kwiatkowski et al., 2019)), and world knowledge and problem solving (MMLU (Hendrycks et al., 2021) domains using the BLEU scores and accuracy metrics. Note that NQ refers to Natural Questions, HM refers to Humanities, SS refers to Social Sciences, Other refers to business, health, and other miscellaneous topics, and Avg. refers to the average accuracy across all 57 subjects in MMLU. We compare our 1.5B parameter model with those trained by OpenAI for 20× more training iterations. The model with the superior performance is highlighted in bold.

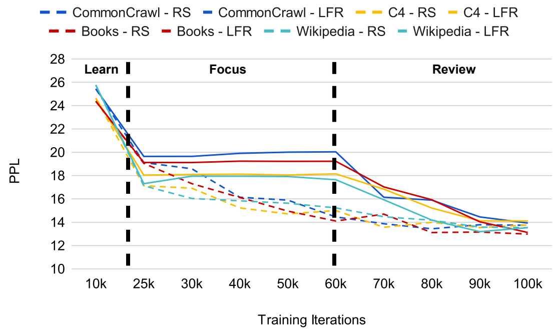


Figure 5: PPL values are tracked at different training iterations for the clusters identified as easy, discarded during the Focus stage, and reintroduced during the Review phase. The dotted line represents the PPL values for the same clusters when trained with random sampling (RS). Notably, we demonstrate that models forget the data points discarded during training, unless reintroduced to the training corpus as in the case of LFR.

ted line). During Focus, when the model prioritizes challenging clusters like GitHub, StackExchange, and ArXiv (Figure 4), it forgets discarded data (solid line rises above dotted). The Review phase restores these points, ensuring better model performance and giving LFR a distinct edge over other methods (Section 5.3). See the Appendix for raw examples of easy and difficult samples identified by LFR.

5.5 Overall Learning Schedule

LFR reveals that models follow a structured learning trajectory: first mastering conversational and anecdotal data (CommonCrawl, C4, books), then retaining factual knowledge (Wikipedia), and finally learning code, QA, and scientific content (ArXiv). By recognizing this progression automatically as shown in Sections 5.4.1 and 5.4.2, LFR optimizes training by dynamically guiding the model at its own learning pace, ensuring efficient and targeted learning.

6 Conclusion

We introduced LFR (Learn-Focus-Review), a novel data selection paradigm that accelerates LLM pretraining while significantly reducing training costs. Through 2200 GPU hours of experiments, LFR achieved lower perplexity and higher accuracy while using up to 20× fewer training iterations than traditional methods. Our findings show that LLMs follow a natural learning progression—first acquiring conversational data, then factual knowledge, and finally mastering code and scientific concepts. By dynamically guiding learning, LFR provides a scalable, cost-effective alternative to existing pretraining strategies. We hope this work inspires further research into more adaptive and efficient training paradigms.

7 Acknowledgments

This work is partially supported by the PRISM Center under the JUMP 2.0 Program, CDSC Industry Partners (`https://cdsc.ucla.edu/partners/`), AMD HACC Program, and NSF 2048280, 2331966, 2325121, 2244760. The authors would like to thank AMD for their generous support provided through the AMD AI & HPC Fund Research Award.

8 Limitations and Ethical Considerations

LFR presents the following directions for future work:

1. LFR is evaluated on models up to 1.5B parameters using open-source corpus like the SlimPajama dataset, constrained by our compute resources. With the clear success on models of such scale, we hope to inspire researchers to validate such focused learning approaches for different model families, and domains.

2. The sensitivity study in Section 5.4 and the Appendix reveals that the hyperparameters selected in Section 4 have a large impact on the performance of the trained model. Due to our limited compute budget, we are unable to present more comprehensive hyperparameter tuning experiments than those presented in Section 5.4.

References

a. Arc Challenge Dataset. `https://huggingface.co/datasets/allenai/ai2_arc`.

b. Arc Easy Dataset. `https://huggingface.co/datasets/allenai/ai2_arc`.

a. BookCorpus Dataset. `https://huggingface.co/datasets/bookcorpus/bookcorpus`.

b. BoolQ Dataset. `https://huggingface.co/datasets/google/boolq`.

GPT-4 Cost Estimation. `https://en.wikipedia.org/wiki/GPT-4#:~:text=Sam%20Altman%20stated%20that%20the,was%20more%20than%20%24100%20million.`

HellaSwag Dataset. `https://huggingface.co/datasets/DatologyAI/hellaswag`.

MiniPile Dataset. `https://huggingface.co/datasets/JeanKaddour/minipile`.

OpenBookQA Dataset. `https://huggingface.co/datasets/allenai/openbookqa`.

OpenWebText Dataset. `https://huggingface.co/datasets/Skylion007/openwebtext`.

PIQA Dataset. `https://huggingface.co/datasets/ybisk/piqa`.

Spaced Repetition: Wikipedia. `https://en.wikipedia.org/wiki/Spaced_repetition`.

WikiText Dataset. `https://huggingface.co/datasets/Salesforce/`.

WinoGrande Dataset. `https://huggingface.co/datasets/allenai/winogrande`.

WMT-14 Hugging Face Dataset. `https://huggingface.co/datasets/wmt/wmt14`.

2024. Meta Llama 3. `https://ai.meta.com/blog/meta-llama-3/`.

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. 2023. Semdedup: Data-efficient learning at web-scale through semantic deduplication. *Preprint*, arXiv:2303.09540.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: a suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. *Preprint*, arXiv:1312.3005.

Jie Chen, Zhipeng Chen, Jiapeng Wang, Kun Zhou, Yutao Zhu, Jinhao Jiang, Yingqian Min, Wayne Xin Zhao, Zhicheng Dou, Jiaxin Mao, Yankai Lin, Ruihua Song, Jun Xu, Xu Chen, Rui Yan, Zhewei Wei, Di Hu, Wenbing Huang, and Ji-Rong Wen. 2024. Towards effective and efficient continual pre-training of large language models. *Preprint*, arXiv:2407.18743.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *Preprint*, arXiv:2101.00027.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2024. A framework for few-shot language model evaluation.

Amir Gholami, Zhewei Yao, Sehoon Kim, Coleman Hooper, Michael W. Mahoney, and Kurt Keutzer. 2024. AI and Memory Wall. *Preprint*, arXiv:2403.14123.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models. *Preprint*, arXiv:2203.15556.

Jean Kaddour. 2023. The MiniPile Challenge for Data-Efficient Language Models. *Preprint*, arXiv:2304.08442.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, yelong shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Not all tokens are what you need for pretraining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Max Marion, Ahmet Üstün, Luiza Pozzobon, Alex Wang, Marzieh Fadaee, and Sara Hooker. 2023. When Less is More: Investigating Data Pruning for Pretraining LLMs at Scale. *Preprint*, arXiv:2309.04564.

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. 2023. Scaling data-constrained language models. *ArXiv*, abs/2305.16264.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. *Preprint*, arXiv:2303.08774.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only. *Preprint*, arXiv:2306.01116.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Paul Smolen, Yili Zhang, and John Byrne. 2016a. The right time to learn: mechanisms and optimization of spaced learning. *Nature Reviews Neuroscience*, 17.

Paul Smolen, Yili Zhang, and John H. Byrne. 2016b. The right time to learn: mechanisms and optimization of spaced learning. *Nature Reviews Neuroscience*, 17(2):77–88.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2019. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences*, 116(10):3988–3993.

Kushal Tirumala, Daniel Simig, Armen Aghajanyan, and Ari S. Morcos. 2023. D4: Improving LLM Pretraining via Document De-Duplication and Diversification. *Preprint*, arXiv:2308.12284.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Mengzhou Xia, Mikel Artetxe, Chunting Zhou, Xi Victoria Lin, Ramakanth Pasunuru, Danqi Chen, Luke Zettlemoyer, and Ves Stoyanov. 2022. Training trajectories of language models across scales. *ArXiv*, abs/2212.09803.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. 2023. Data selection for language models via importance resampling. *Preprint*, arXiv:2302.03169.

Zichun Yu, Spandan Das, and Chenyan Xiong. 2024. Mates: Model-aware data selection for efficient pretraining with data influence models. *ArXiv*, abs/2406.06046.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ye Yuan, Guoren Wang, and Conghui He. 2024. Harnessing diversity for important data selection in pretraining large language models. *ArXiv*, abs/2409.16986.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *ArXiv*, abs/2205.01068.

A Appendix

A.1 Experiment Details

Datasets The datasets used for our experiments are detailed below:

1. ARC-Challenge (arc, a): A subset of the AI2 Reasoning Challenge with 2,590 challenging multiple-choice science questions designed to test advanced reasoning and knowledge.

2. ARC-Easy (arc, b): A subset of the AI2 Reasoning Challenge with 5,117 relatively easier multiple-choice science questions focusing on basic reasoning and recall.

3. BoolQ (boo, b): A dataset of 16,000+ boolean (yes/no) questions paired with passages, requiring models to infer answers from supporting evidence.

4. HellaSwag (hel): A dataset with 70,000+ multiple-choice questions focused on commonsense reasoning and contextual understanding, particularly in describing scenarios.

5. OpenBookQA (Ope): A multiple-choice question-answering dataset with 5,957 questions requiring knowledge retrieval from a science "open book" and commonsense reasoning.

6. PIQA (Piq): A physical commonsense reasoning dataset with 20,000+ binary-choice questions about everyday situations and physical interactions.

7. Winogrande (win): A dataset with 44,000+ sentence pairs designed to test commonsense reasoning through pronoun disambiguation challenges.

8. WikiText (wik): the WikiText language modeling dataset consists of 100M tokens extracted from Wikipedia articles with high rating. It features two different variants, namely, WikiText-2 and WikiText-103 which differ in the number of tokens and vocabulary size. WikiText-2 consists of 2M tokens and a vocabulary size of 33k whereas WikiText-103 is larger with 103M tokens and a vocabulary size of 267k.

9. LAMBADA (Paperno et al., 2016): the LAMBADA dataset is extracted from the BookCorpus dataset (boo, a) and contains 10k passages. This dataset is useful for testing the ability of an LLM to capture long-range dependencies in text. The objective of this model is to predict the final word in a set of sentences, where humans need at least 50 tokens of context to accurately anticipate the word.

10. One Billion Word Benchmark (Chelba et al., 2014) (1BW): this dataset contains one billion words extracted from the WMT 2011 News Crawl data and is used to measure progress in statistical language modeling.

11. WMT-14 French-English Translation (Artetxe et al., 2018): This dataset contains 36 million training sentence pairs for english to french translation. The test set, which is used for evaluation purposes, consists of 3,003 sentence pairs.

12. Natural Questions (Kwiatkowski et al., 2019): This dataset contains question-answer pairs from Google Search and Wikipedia-based annotations. The training, validation, and test sets consist of 307,372, 7,830, and 7,842 examples.

Models: Tables 4 and 5 describes the different model configurations and pretraining hyperparameters used in LFR for the Llama models.

300M	500M	1.1B	
Layers	12	11	22
#Heads	16	32	32
n_embd	1024	2048	2048

Table 4: Number of layers, attention heads, and the embedding dimensions in the Llama models used for pretraining.

Tables 6 and 7 describes the different model configurations and pretraining hyperparameters used in LFR for the GPT-2 models.

Pretraining: Table 7 shows the hyperparameters for pretraining the GPT-2 124M-1.5B parameter models.

Note that OpenAI pretrained the GPT-2 models using a batch size of 512. Due to insufficient GPU memory, we adjust the number of gradient accumulation steps to achieve the same effective batch size of 512.

Finetuning: We use all the same hyperparameters as pretraining, except for the following:

1. Learning rate: 3.00E-5

Parameter	Value
Context Length	1024
Embedding Dimension	(768, 1024, 2048)
Total Iterations	100,000
Effective Batch Size	768
Block Size	4096
Weight Decay	1.00E-1
Adam β_1	0.90
Adam β_2	0.95
Warmup Iterations	8000
Minimum Learning Rate	4.00E-5
Maximum Learning Rate	4.00E-04
Learning Rate Schedule	Cosine Decay
Learning Rate Decay Iterations	100,000
GPUs	(4x AMD MI210, 4x AMD MI210, 8x AMD MI250)

Table 5: Pretraining hyperparameters for the Llama 300M-1.1B parameter models. Parameters with multiple values (e.g. Embedding dimensions, batch size, gradient accumulation steps, and GPUs) specified in brackets are for the 300M, 500M, and 1.1B parameter models respectively.

2. Learning rate schedule: Constant
3. Total iterations: 50

A.2 Limitations and Ethical Considerations

LFR presents the following directions for future work:

1. LFR is evaluated on models up to 1.5B parameters using web-scale datasets such as SlimPajama, constrained by our compute resources. With the clear success on models of such scale, we hope to inspire researchers to validate such focused learning approaches for different model families, and domains.
2. The sensitivity study in this Appendix reveals that the hyperparameters selected in the evaluation section have a large impact on the performance of the trained model. Due to our limited compute budget, we are unable to present more comprehensive hyperparameter tuning experiments than those presented later in this Appendix.

	124M	355M	774M	1.5B
Layers	12	24	36	48
#Heads	12	16	20	25
n_embd	768	1024	1280	1600

Table 6: Number of layers, attention heads, and the embedding dimensions in the GPT-2 models used for pretraining.

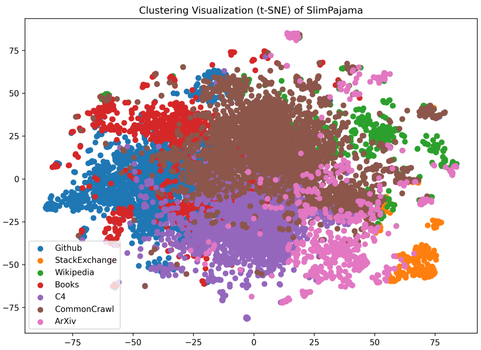


Figure 6: Clustering the data embeddings from the SlimPajama dataset using the Llama-300M model at the 50k training step.

A.3 Llama Pretraining - Data Importance

In this section, we study the data points identified as easy and challenging by LFR when pretraining with the SlimPajama dataset. Listing A.3 provides an example of a code snippet from Github classified as easy by LFR, and discarded in the Focus stage of the Llama model training. Listing A.3 provides an example of a data sample retained from the Github cluster. Note that this code is more complex, presents an opportunity to the model to improve its coding capabilities as opposed to the variable declarations in Listing A.3.

Listing 1: Code snippet classified as easy by LFR, primarily consisting of variable declarations. As seen from the code, it contributes minimally to enhancing the model's coding capabilities.

```
package frclibj;

import edu.wpi.first.wpilibj.
    Timer;

public class TrcDbgTrace
{
    public static final String
        ESC_PREFIX       = "\u001b
        [";
```

Parameter	Value
Context Length	1024
Embedding Dimension	(768, 1024, 1280, 1600)
Total Iterations	40000
Effective Batch Size	512
Batch Size	(16, 16, 8, 4)
Gradient Accumulation Steps	(32, 32, 64, 128)
Block Size	1024
Weight Decay	1.00E-01
Adam β_1	0.9
Adam β_2	0.95
Warmup Iterations	2000
Minimum Learning Rate	6.00E-05
Maximum Learning Rate	6.00E-04
Learning Rate Schedule	Linear
Learning Rate Decay Iterations	40000
GPUs	(4xMI100, 4xMI210, 4xMI210, 4xMI210)

Table 7: Pretraining hyperparameters for the GPT-2 124M-1.5B parameter models. Parameters with multiple values (e.g. Embedding dimensions, batch size, gradient accumulation steps, and GPUs) specified in brackets are for the 124M, 345M, 774M, and 1.5B parameter models respectively.

```
    public static final String
        ESC_SUFFIX          = "m";
    public static final String
        ESC_SEP             = ";";

    public static final String
        SGR_RESET           = "0";
    public static final String
        SGR_BRIGHT          = "1";
    public static final String
        SGR_DIM             = "2";
    public static final String
        SGR_ITALIC          = "3";
    public static final String
        SGR_UNDERLINE       = "4";
    public static final String
        SGR_BLINKSLOW       = "5";
    public static final String
        SGR_BLINKFAST       = "6";
    public static final String
        SGR_REVERSE         = "7";
    public static final String
        SGR_HIDDEN          = "8";
    public static final String
        SGR_CROSSEDOUT      = "9";

    public static final String
        ESC_NORMAL          =
        ESC_PREFIX;
}
```

Listing 2: Code snippet classified as challenging by LFR. This code consists of a function which executes an Oracle query and returns a scalar value. As seen from the code, it contributes significantly to enhancing the model's coding capabilities as compared with Listing A.3.

```
/// <summary>
/// Executes an Oracle query that
    returns a single scalar value
    as the result.
/// </summary>
/// <param name="commandText">The
    Oracle query to execute</
   param>
/// <param name="parameters">
   Optional parameters to pass to
    the query</param>
/// <returns>The result of the
   query as an object</returns>
public object QueryValue(string
   commandText, IEnumerable
   parameters)
{
    object result;

    if (String.IsNullOrEmpty(
        commandText))
    {
        throw new
            ArgumentException("
            Command text cannot be
             null or empty.");
    }

    try
    {
        ensureConnectionOpen();
        var command =
            createCommand(
            commandText,
            parameters);
        result = command.
            ExecuteScalar();
    }
    finally
    {
        ensureConnectionClosed();
    }

    return result;
}
```

Similarly, we also provide examples of question-answer pairs from StackExchange which were discarded and retained in the Focus stage of the Llama pretraining in Listings A.3 and A.3 respectively.

Listing 3: Question-answer pair from StackExchange classified as easy by LFR. The question revolves around a process in PayPal which does not contribute as much to the answering capability or world knowledge of the model.

```
Q: PayPal IPN $_POST['txn_id']
    not set. I'm using the PayPal
    sandbox to do a subscribe
    button, and then when I get
    the IPN response for a
    subscription or a subscription
     cancellation $_POST['txn_id']
     is never set.
So I don't know how to identify
    transactions to only accept
    unique ones.
Thanks!
EDIT: for example, all the info
    that I have in POST for a
    subscr_cancel are:
amount1, amount3, address_status,
     subscr_date, payer_id,
    address_street, mc_amount1,
    mc_amount3, charset,
    address_zip, first_name,
    reattempt,
    address_country_code,
    address_name, notify_version,
    subscr_id, custom,
    payer_status, business,
    address_country, address_city,
     verify_sign, payer_email,
    btn_id, last_name,
    address_state, receiver_email,
     recurring, txn_type,
    item_name, mc_currency,
    residence_country, test_ipn,
    period1, period3,
    correlation_id.

A: According to Table 2. Summary
    of subscription variables:
For subscription variables, the
    transaction ID (txn_id) is
    only available for USD Payment
     and Multi-Currency Payment
    transaction types (txn_type).

As expected, PayPal will not send
     the txn_id to your IPN for
    the transaction type,
    subscr_cancel, and will only
    send txn_id if the transaction
     type is subscr_payment.

For further explanation on which
    variables are sent to your IPN
     URL based on your transaction
    , please check out IPN and PDT
     Variables.

Have you checked $_REQUEST['
    txn_id'] as this may be sent
    to your server via GET.
```

Listing 4: Question-answer pair from StackExchange classified as challenging by LFR. The question revolves around solving an ODE which contributes more to the learning of the model than Listing A.3.

```
Q: Passing additional iteration-
    dependent inputs to ode45
I'm trying to solve a
    differential equation using
    the ode45 function. Consider
    the following code,
[t1,X2] = ode45(@(t,x)fun(t,x,C1,
    C2,C3,C4),t0,X01);

where parameters C1, C2, C3, and
    C4 are column vectors, which
    should be available to the
    function that ode45 is
    referring to (fun.m).
I want the values to change after
     every iteration, so for
    example, at the beginning the
    entry of C1 I want is C1(1),
    in the next iteration it's C1
    (2), etc.
How can I implement that?

A: You may have noticed that the
    official docs are not too
    helpful in this scenario (as
    they pretty much force you to
    use global variables - which
    is doable, but discouraged).
 Instead, I'll show you how this
    can be done with classes and
    function handles. Consider the
     following:

classdef SimpleQueue < handle
  %SIMPLEQUEUE A simple FIFO data
      structure.

   properties (Access = private)
     data
     position
   end

   methods (Access = public)
     function obj = SimpleQueue(
        inputData)
       %SIMPLEQUEUE Construct an
          instance of this class
       obj.data = inputData;
       rewind(obj);
     end % constructor

     function out = pop(obj,
        howMany)
       %POP return the next
          howMany elements.
       if nargin < 2
         howMany = 1; % default
            amount of values to
            return
       end
       finalPosition = obj.
          position + howMany;
       if finalPosition > numel(
          obj.data)
         error('Too many elements
            requested!');
       end
       out = obj.data(obj.position
          + 1 : obj.position +
          howMany);
       obj.position =
          finalPosition;
     end % pop

     function [] = rewind(obj)
       %REWIND restarts the
          element tracking
       % Subsequent calls to pop()
          shall return elements
          from the beginning.
       obj.position = 0;
     end % rewind
   end % methods
end % classdef

How to use this? Simple:
C1q = SimpleQueue(C1);
C2q = SimpleQueue(C2);
C3q = SimpleQueue(C3);
C4q = SimpleQueue(C4);

[t1,X2] = ode45(@(t,x)fun(t,x,
    @C1q.pop,@C2q.pop,
@C3q.pop,@C4q.pop),t0,X01);

As you can see, inside fun we use
    C1q() instead of C1.
```

A.4 Sensitivity Study

In this section, our goal is to understand the effects of more aggressive focus, revision, and learning strategies than the training strategy presented in the paper. Here, we vary the values of hyperparameters p_1, s_1, p_2, p_3, and *reps* and study the effects on the downstream task perplexity. Note that the GPT-2 models used a four phase training process. Specifically, we aim to answer the following two questions using the GPT-2 models:

1. What is the impact of not reintroducing the discarded data samples?
2. What is the impact of the degree of pruning in Phases 2 and 4?

To answer the first question, we pretrain a 124M parameter GPT-2 model without the reintroduction of data blocks in Phase 3, and use the reduced subset from Phase 2 for the rest of the training. Then, we finetune for downstream language modeling tasks similarly and compared the perplexities using LFR in Table 8. This training strategy which removes Phase 3, is labeled as `no-reintro`. Next, to answer the second question, we pretrain a 124M parameter GPT-2 model using LFR but increase the degree of pruning in Phase 2 from 50% to 70%, i.e., reduce the training subset to 30% of the original size. This aggressive training strategy is labeled as `aggr-2`.

We observe that both aggressive training strategies do not work as well as the original method. However, we continue to explore more automated ways of deciding the training schedule for different model families as part of our future work.

Model	WikiText-2	WikiText-103	LAMBADA	1BW
`no-reintro`	23.24	25.76	17.27	36.02
`aggr-2`	23.91	27.00	21.11	38.62
LFR	19.81	22.49	16.61	32.27

Table 8: Downstream task perplexities with more aggressive training strategies.

A.5 Analysis on Dropped and Retained Data Blocks - GPT-2

In this section, our goal is to characterize the data points retained and dropped during pretraining by LFR in Phases 2 and 4 across the training time and model size. Specifically, we aim to answer the following questions:

1. What types of data blocks are learned earlier in the training process compared to those learned later?
2. Are similar data blocks considered learned and dropped in Phases 2 and 4?
3. Are the dropped data blocks similar across model sizes?
4. Are the data blocks dropped similar to those retained at any given training phase?

To answer the first question, we printed out the texts dropped and retained at different training phases. Tables 12 and 14 show text blocks dropped in Phases 2 and 4 by the 345M and 124M parameter models, while Tables 13 and 15 show data blocks retained. By reading through the texts, we notice that the model first learned conversations and personal anecdotes, before being able to retain factual information. We provide a more detailed analysis of the learning process in Section A.6.

In order to answer questions 2-4, we recorded only the IDs of dropped data blocks during Phases 2 and 4 for both the GPT-2 124M and GPT-2 345M models, totaling 4 lists of dropped IDs. We then load the recorded data blocks and embed them into a higher dimensional space using the GPT-2 tokenizer. Considering that there is a total of 8.7M data blocks (9B tokens divided into blocks of 1024 tokens), we cluster the embeddings using k-means clustering with $k = 270$ to reduce the analysis space and complexity. Finally, for each model, we compute the cosine similarity for all combinations of the embeddings of dropped data blocks across training phases and visualize them using a heatmap. These heatmaps plot the cosine similarity values (ranging between 0 and 1) such that lighter values (closer to 1) indicate higher similarity.

Figure 7 shows the similarity of dropped data blocks across the time scale (Phase 2 shown on the X-axis and Phase 4 shown on the Y-axis) for the 124M (left) and 345M (right) parameter models. We find that there is a higher similarity in the data points dropped by the 124M parameter model in Phases 2 and 4 than in the case of the 345M parameter model (mean, variance, and standard deviation are provided in Table 9). This behavior signals that the lower capacity of the 124M parameter model inhibits its learning process in Phase 3, such that it finds similar points confusing in Phases 2 and 4. In contrast, the 345M parameter model learns the data blocks it found confusing in Phase 2 by focusing on them, and moves on to learning new complex blocks by Phase 4.

We conduct a similar study in order to characterize the similarity in data blocks across model sizes. Figure 8 plots the cosine similarity heatmap for the data blocks dropped by the 124M parameter model (X-axis) and those dropped by the 345M parameter model (Y-axis) in Phase 2. The mean, variance, and standard deviations of the cosine similarity are 0.38, 0.15, and 0.023, respectively. This indicates that the data blocks found easy and dropped in Phase 2 by both models display a moderate level of similarity, but also differ significantly.

Finally, we observe the cosine similarity of data blocks dropped and retained during phase 4 for the 124M (left) and 345M (right) parameter models in Figure 9. The mean, standard deviation, and variance are detailed in Table 10. The smaller model displays greater similarity (lighter values in the heatmap) between the dropped and retained blocks than the larger model. We hypothesize that the larger model can perform reasonably well across similar data points, but struggles with very different complex blocks by the fourth training phase.

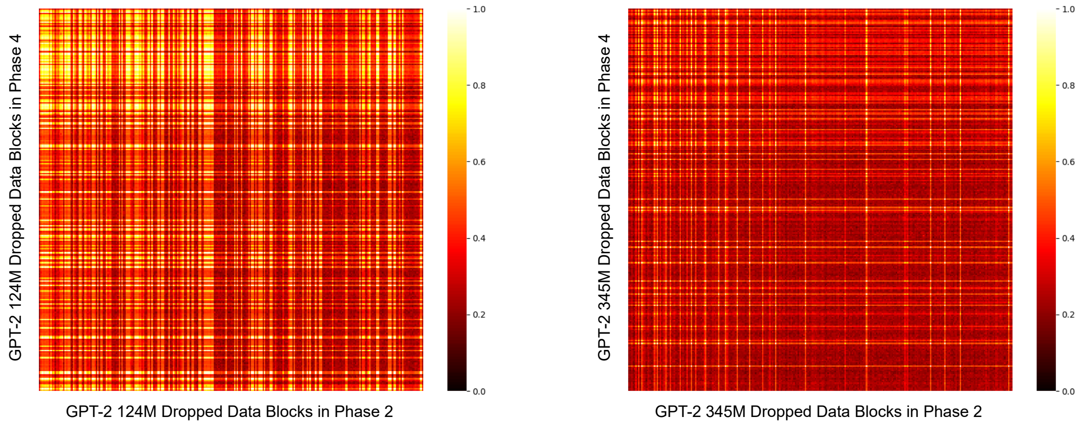


Figure 7: Cosine similarity heatmaps for dropped data blocks during phases 2 and 4 of pretraining for the GPT-2 124M (right) and 345M (left) models. The smaller model displays greater similarity in dropped data blocks over time (lighter color), indicating that it remained uncertain about similar data points than the larger model.

Model	Mean	Std	Variance
GPT-2 124M	0.45	0.20	0.04
GPT-2 345M	0.30	0.12	0.01

Table 9: Mean, standard deviation (std), and variance of cosine similarity matrices for dropped data blocks across time scale (Phase 2 and Phase 4) for the GPT-2 124M and 345M models.

Model	Mean	Std	Variance
GPT-2 124M	0.44	0.21	0.046
GPT-2 345M	0.32	0.13	0.018

Table 10: Mean, standard deviation (std), and variance of cosine similarity matrices for dropped and retained data blocks in Phase 4 of pretraining for the GPT-2 124M and 345M models.

In contrast, the smaller model does not display the same high-level of understanding (similar perplexity values) on related data blocks.

To summarize, **data block importance varies across training time, and across model sizes**. Therefore, static data selection techniques (Tirumala et al., 2023; Abbas et al., 2023; Kaddour, 2023; Xie et al., 2023) which select a fixed subset to train for the entire training duration for all model architectures do not adapt to the changing training dynamics of LLMs. Based on our analysis in Figure 8 and 7, different data blocks are found difficult by models of different capacities at different training instants, driving the need for dynamic data selection methods like LFR. We detail further analysis on the selected and discarded data blocks and demonstrate how models initially focus on learning conversational and anecdotal data, before proceeding to learn factual data in Appendix A.6.

A.6 Extended Analysis on Dropped and Retained Data Blocks for GPT-2

In this section, we expand on the ablation study in Section A.5 in order to better characterize the data

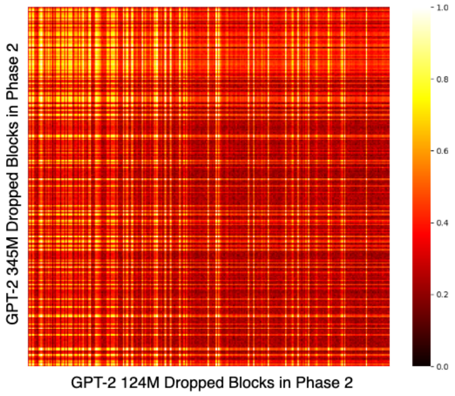


Figure 8: Cosine similarity heatmap for data blocks dropped during Phase 2 of GPT-2 124M and 345M pretraining shows moderate similarity, indicating different data points are considered easy by each model.

Model	Mean	Std	Variance
GPT-2 124M	0.42	0.19	0.04
GPT-2 345M	0.40	0.18	0.03

Table 11: Mean, standard deviation (std), and variance of cosine similarity matrices for dropped and retained data blocks in phase 2 of pretraining for the GPT-2 124M and 345M models.

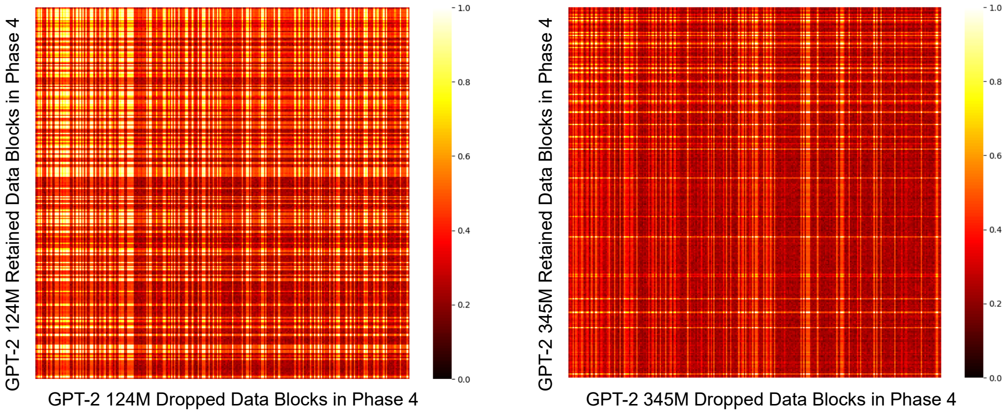


Figure 9: Cosine similarity heatmaps for dropped and retained data blocks during Phase 4 of pretraining for the GPT-2 124M (right) and 345M (left) models.

Become a fan of Slate on Facebook. Follow us on Twitter.The first time I crocheted a soccer ball was on the occasion of the 2010 World Cup. It was being held on the continent of Africa, and I thought the African Flower hexagon motif was the perfect vehicle for a crochet soccer ball celebrating the continent's first time hosting the World Cup: This time around, instead of using all 9000 of my favorite colors, I limited myself to the colors of the flags of the thirty-two countries that had made it to the final rounds of the World Cup competition, and I did my best to incorporate the designs of their flags into the thirty-two hexagons and pentagons of a soccer ball.
ML-77 Missile Launcher: Based on existing technology, the ML-77 is a rapid-fire missile launcher using seeking projectiles. Each projectile features a friend-or-foe recognition system, ensuring it will find a hostile target even if the user's aim is not completely accurate. The locking mechanism of the ML-77 allows the shooter to ignore cover and line of sight when shooting at locked on enemies, though an attack roll is still required. Locking on to an enemy requires a move action when the enemy is in line of sight and lasts for the rest of the encounter, or until a new target is locked.

Table 12: Examples of text dropped by the 345M model in phase 2 (top) and phase 4 (bottom).

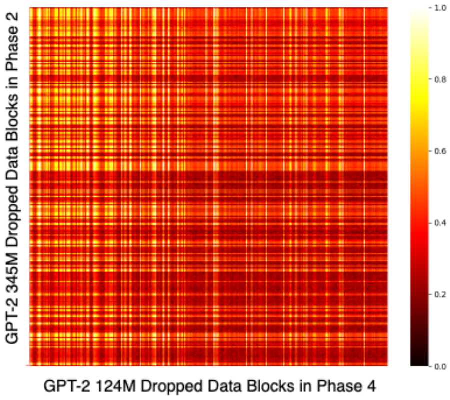


Figure 10: Cosine similarity heatmap for dropped data blocks during Phase 4 of GPT-2 124M and Phase 2 of the 345M model.

blocks considered easy / hard.

Tables 12 and 14 provides examples of text blocks dropped in Phases 2 and 4 by the 345M and 124M parameter models respectively. Similarly, Tables 13 and 15 provide examples of data blocks retained by the models in Phases 2 and 4. We printed out and went over all the text dropped and retained in both Phases, and notice that text considered easy in phase 2 was more conversational, and those considered easy in phase 4 were more factual. This might indicate that the model first learned conversations and personal anecdotes, before being able to retain factual information. These findings are further corroborated by the examples of data retained in both phases. We are working on further analysis across different model families and sizes to strengthen this understanding.

Next, we continue the analysis of the cosine similarity heatmaps evaluated across training time and model parameter scales presented in Section A.5. Here, we answer the following questions:

1. Are there similarities in the data blocks considered easy and dropped in Phase 4 of training of the 124M parameter model with those considered easy and dropped by the 345M parameter model in Phase 2?

2. Are the data blocks dropped similar to those retained at any given training phase? Note that Section A.5 presented this analysis only for Phase 4 of the 124M and 345M parameter models in Figure 9.

Figure 10 depicts the cosine similarity heatmap for the data blocks dropped by the 124M parameter model in Phase 4 (X-axis) with those dropped by the 345M parameter model in Phase 2 (Y-axis). The mean, standard deviation, and variance of the similarity are 0.43, 0.18, and 0.03 respectively. In contrast, the mean cosine similarity of data blocks dropped in Phase 2 of pretraining of both the models was 0.38 (Section A.5 and Figure 8). This indicates that the smaller model "catches up" with the knowledge accumulated by the larger model, and considers similar block easy in Phase 4 as those considered easy by the larger model in Phase 2.

Next, we plot the cosine similarity heatmap for the dropped and retained data blocks in Phase 2 for the 124M (left) and 345M (right) parameter models. The mean, variance, and standard deviations of the similarity are shown in Table 11. Observing the mean similarity value and heatmap in Table 10 and Figure 9, we find that the cosine similarity for dropped and retained data blocks is higher in Phase 2 than Phase 4 in case of the 345M parameter model. In contrast, the value remains high in both Phases for the 124M parameter model. This finding indicates that both the smaller and larger model start the training by being confused about similar data blocks. However, the larger capacity of the 345M parameter model allows it to learn the dataset well in Phases 2 and 3, and move on to more complex data blocks in Phase 4 (thus reducing the mean similarity in Phase 4). The smaller model continues remaining unsure about similar data blocks. Since we observed that the smaller model "catches up" with the training of the larger model (in Figure 10), we hypothesize that the smaller model will eventually display similar behaviour as the larger model once trained for longer iterations.

Unofficial reports claimed the car was powered by a 95kW 1.5-litre non-turbo petrol engine but Tada didn't confirm. When asked what powers the S-FR Tada revealed he was considering three choices. "When you see the S-FR concept I suppose you imagine it is a 1.5-litre car but nowadays I can choose many kind of engines," he explained. "Downsized turbo, 1.5-litre naturally aspirated and something additional as well. Now we are thinking which one is the best engine for a small sports car." Tada also admitted that the company is unlikely to turn to a partner like it did with Subaru for the 86/BRZ or the new 'big brother' sports car with BMW.
In April, MYIR released a Linux-powered MYS-6ULX single board computer, which was notable for being available in two different versions using NXP's low power, Cortex-A7 i.MX6 UltraLite (UL) or the more affordable, and almost identical i.MX6 ULL SoC. Now, MYIR has released an "MYB-6ULX Expansion Board" designed to stack onto either model. The $21.20 accessory adds a second 10100 Ethernet port to the MYS-6ULX, as well as new CAN, RS485, audio, micro-USB, RTC, and camera functions. MYB-6ULX Expansion Board with MYS-6ULX (left) and detail view (click images to enlarge). The MYB-6ULX Expansion Board has the same 70 x 55mm dimensions as the MYS-6ULX, which is available in two models: The i.MX6 UL based MYS-6ULX-IND has -40 to 85°C support instead of 0 to 70°C, and the i.MX6 ULL based MYS-6ULX-IOT features a USB-powered WiFi radio. The 4-layer expansion board runs on 5V power, and shares the industrial temperature support of the IND model.

Table 13: Examples of text retained by the 345M model in Phase 2 (top) and Phase 4 (bottom).

In the book, the mythical California is ruled by Queen Califa and populated only with female warriors who brandish gold weapons. They even harness their animals in gold because it is the only mineral on the island. The legend of Califa and her island was well known among New World explorers. In 1536 when Hernán Cortéz arrived in Baja California, he believed he had landed on the legendary island. Over three hundred years later gold was discovered in California, making the legend partially true and earning the state its nickname: The Golden State.
Segregated Witness, defined by Bitcoin Improvement Proposal 141 (BIP141), was deployed using an activation mechanism (BIP9) that requires 95 percent of all miners (by hash power) to signal support for the upgrade within the span of a two-week difficulty period. That's at least 1916 blocks within 2016 blocks, to be exact. This threshold has just been reached. While the current difficulty period will not end until tomorrow, all blocks in this difficulty period are signaling support for the upgrade so far. This now totals over 1916 of them.

Table 14: Examples of text dropped by the 124M model in Phase 2 (top) and Phase 4 (bottom).

to the GUI installer. Most notably there is no support for configuring partition layout, storage methods or package selection. Please refer to the official documentation for details. Here you can find some useful information on creating and using kickstart files which can be used to perform advanced configuring without the need for the GUI installer. The message "Insufficient memory to configure kdump!" appears during install. This is a known issue which appears on systems with less than 2 GB RAM. This can be ignored. Content for both the i386 and x86_64 architectures is split into two DVDs. We have tried to get all basic server and basic desktop installs only from DVD-1. Make sure that you setup correctly the selinux context of the public key if you transfer it to a CentOS 6 server with selinux enabled.
Once you signed up, you can either click on the Todo tab or the sign in link to gain access to the application. Note that if you are selecting sign in in the same session in which you signed up, you will automatically sign in with the same account you used for signing up. If you are signing in during a new session, you will be presented with Azure AD's credentials prompt: sign in using an account compatible with the sign up option you chose earlier (the exact same account if you used user consent, any user form the same tenant if you used admin consent). If you try to sign-in before the tenant administrator has provisioned the app in the tenant using the Sign up link above, you will see the following error.

Table 15: Examples of text retained by the 124M model in phase 2 (top) and phase 4 (bottom).

What does memory retrieval *leave on the table*? Modelling the Cost of Semi-Compositionality with MINERVA2 and sBERT

Sydelle de Souza^{1,3,4*} **Ivan Vegner**^{1,3*} **Francis Mollica**^{2‡}
Leonidas A.A. Doumas^{4‡}

^{1}School of Informatics, University of Edinburgh ^{2}University of Melbourne
^{3}Centro de Linguística da Universidade de Lisboa
^{4}School of Philosophy, Psychology & Language Sciences, University of Edinburgh

Abstract

Despite being ubiquitous in natural language, collocations (e.g., *kick+habit*) incur a unique processing cost, compared to compositional phrases (*kick+door*) and idioms (*kick+bucket*). We confirm this cost with behavioural data as well as MINERVA2, a memory model, suggesting that collocations constitute a distinct linguistic category. While the model fails to fully capture the observed human processing patterns, we find that below a specific item frequency threshold, the model's retrieval failures align with human reaction times across conditions. This suggests an alternative processing mechanism that activates when memory retrieval fails.

1 The Curious Case of Collocations

From *killing time* and *playing dead* to *running baths* and *making beds*, word combinations with semi-compositional meanings are ubiquitous in human language (Cowie, 1998). Often referred to as *collocations*, these idiosyncratic lexical elements comprise one word used in its literal sense and another in its figurative sense, constrained by an arbitrary restriction on substitution (Mel'čuk, 2003; Howarth, 1998). Thus, one can *raise questions* or *lift bans*, but neither #*lift questions* nor #*raise bans*. Collocations are syntactically well formed, but deviate from or violate the expected semantic representation (Culicover et al., 2017). To illustrate, the verb *kill* prototypically requires an animate object, so one can *kill bugs* and *kill trees*, but not **kill books*. Yet one can *kill time*, *hope*, and *dreams*. Collocations are the largest subset of formulaic language (Barfield and Gyllstad, 2009) with many being cross-linguistically attested (Yamashita, 2018). It is hardly surprising, then, that proper knowledge and use of such units provides fluency and idiomaticity to the language user (Pawley and Syder, 1983; Durrant and Schmitt, 2009). Yet, they pose an enormous hurdle to second-language learners and machines.

According to Howarth (1998), human language lies on a theoretical continuum of semantic compositionality—the degree to which the meaning of a phrase can be derived from the meaning of its constituent parts and their syntactic relations (Frege, 1892). Fully compositional combinations (e.g., *chase rabbits*, *chase thieves*, etc.) and fully non-compositional figurative idioms (e.g., *chase one's tail*)[1] lie on extreme ends of the spectrum. Semi-compositional collocations (e.g., *chase dreams*, *chase money*, etc.) lie in between. The psychological validity of this continuum has been tested with the expectation that a decrease in compositionality is directly proportional to a decrease in processing time (Gyllstad and Wolter, 2016). However, empirical evidence from both first (L1) and second (L2) language speakers shows that collocations are processed slower and less accurately than fully compositional combinations (Gyllstad and Wolter, 2016; de Souza et al., 2024), and fully opaque and non-compositional figurative idioms (e.g., *break the ice*) are processed faster and more accurately than compositional combinations (e.g, *break the cup*) (Carrol and Conklin, 2020; Tabossi et al., 2008).

These disparities are also seen in evidence from L2 acquisition[2]. Research shows that colloca-

---

*Joint first authors ‡ Joint senior authors
Correspondence to: `sydelle.desouza@ed.ac.uk`

[1] It is important to note that (Howarth, 1998) also specifies a fourth category called "pure idioms" (e.g., *blow the gaff*, *take a leak*, *shoot the breeze*). These do not possess well-specified literal meanings (see Mueller and Gibbs, 1987, for further reading) and comprise a very small subset of formulaic language occurring quite infrequently (Grant, 2005). Furthermore, most of the studies in this area focus on figurative idioms that have an additional literal reading (e.g., *kick the bucket*). Therefore, in order to constrain the scope of this paper, we limit our discussion to figurative idioms.

[2] Note that collocations as a distinct linguistic class have largely been ignored in L1 acquisition research and little is known about the mechanisms behind a child's acquisition of

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 291–311
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

tions are a major hurdle for second language (L2) learners—be they early sequential bilinguals (Nishikawa, 2019; Riches et al., 2022) or adults (Yamagata et al., 2023; Sonbul et al., 2024), even at high proficiency levels (Wolter and Gyllstad, 2013; Tsai, 2020). In contrast, idioms are learned better and used more accurately than collocations (Fioravanti et al., 2021). Cast under the broader term of *conceptual metaphor* (Lakoff and Johnson, 1980), collocations are also found to be challenging for NLP systems (Liu et al., 2022; Zayed et al., 2018; Czinczoll et al., 2022) despite the fact that the last decade has seen immense progress (see Tong et al., 2021; Wang et al., 2025).

The coalescing picture suggests that idioms are processed the fastest, followed by compositional units, and collocations the slowest. However, this processing hierarchy has not yet been directly tested in a within-participants design—a gap the present study addresses. Next, we discuss how this empirical gap is compounded by a theoretical gap.

2 Accounting for Collocation Processing

It is generally agreed upon in the language processing literature that idioms are stored and retrieved from memory holistically (Carrol and Conklin, 2014; Noveck et al., 2023). Although there are several theories concerning the processing of compositional language, there is less consensus on the matter. Being not too compositional and not too idiomatic, collocations have occupied a theoretical grey zone in mainstream psycholinguistics wherein they have been conveniently ignored in favour of a binary distinction between rules and exceptions. As a result, researchers in second language acquisition and applied psycholinguistics have drawn on the (in)famous Past Tense Debate (Seidenberg and Plaut, 2014) in morphological processing and resorted to single- versus dual-route models to explain processing at the multi-word level (Wray, 2002). We explore these models and consider their ability to account for collocational processing.

Assuming a domain-general hypothesis space, **single-route models** posit that all linguistic forms are stored in and retrieved from a single massive associative memory system[3] based on frequency of input and use (Bybee, 2012; Ambridge and Lieven, 2011). The more often a unit is encountered and/or used, the better it is entrenched in memory (Divjak, 2019; Langacker, 1987). Eventually, this leads to automatization—pure retrieval from memory[4] (Bybee, 2006) which makes processing fast and effortless. Positing such a homogenous mechanism makes for a parsimonious theoretical account of our language abilities, in particular, and our cognition in general. However, human memory is not only limited in capacity (Christiansen and Chater, 2008) but is also unstable (Kornell and Bjork, 2009). We do not store everything we encounter, nor do we remember everything we do store. More importantly, recall that behavioural evidence points to collocations incurring a processing cost versus compositional units even when frequency-matched (see de Souza et al., 2024). While memory undoubtedly plays an important role in language processing, it does not provide a satisfactory account for the processing cost of collocations which occur quite frequently (Barfield and Gyllstad, 2009).

The **dual-route model** assumes a domain-specific hypothesis space, differentiating between words and rules (Pinker, 1991). Regular word forms are thought to be computed analytically (e.g., *walk* → *walk* + *ed*, *scratch* → *scratch* + *ed*) by way of rules, while irregular word forms (e.g., *run* → *ran*, *think* → *thought*) are processed via holistic storage and retrieval from memory (Pinker, 2013).

This theoretical distinction between computation and storage is a practical trade-off between two independent cognitive processes—procedural computation and declarative memory (Pinker and Ullman, 2002). More rule-based computation means less storage. More storage means less computation. Positing such a heterogenous mechanism makes for a persuasive theoretical account of how human language can be infinitely compositional despite our limited cognitive capacities (O'Donnell et al., 2009; Galke et al., 2024). The dual-route explanation is used to account for formulaic language processing as a whole, i.e., it does not distinguish between the various subsets of multi-word units such as idioms, phrasal verbs, binomials, etc. (see Wray, 2002, 2008; Sidtis, 2020). All formulaic language is thought to be stored, while compositional language is computed on the fly. Memory retrieval is faster than analytic processing (Logan, 1997; Dasgupta and Gershman, 2021), therefore formulaic language is thought to be processed faster

---

collocations over development (see Handl and Graf, 2010).

[3] Or that all forms are processed equally as in a connectionist network (see McClelland and Rumelhart, 1985).

[4] See Logan and Etherton (1994) for a domain-general cognitive account of automatization.

than non-formulaic language (Carrol and Conklin, 2014; Vilkaite and Schmitt, 2019). This is empirically consistent across a variety of tasks only in the case of fully non-compositional units like idioms (Noveck et al., 2023). However, dual-route hypotheses make a binary distinction between compositional and formulaic language and ignores the effect of frequency on computation and retrieval. If collocations are frequent and retrieved from memory, the processing cost they incur remains unaccounted for.

3 The Present Study

We begin by addressing the empirical gap laid out in Section 1 and test whether collocations incur a processing cost relative to idioms and fully compositional phrases, as suggested by prior literature. To this end, we ask: *Do collocations take longer to process than idioms and compositional items?* We extend de Souza et al. (2024) by testing L1 English speakers on an acceptability judgement task (AJT) using stimuli from all three conditions and analyse reaction times (RTs) and accuracy. We consider three competing predictions:

Under a *single-route* account, frequency effects should dominate: idioms should be retrieved fastest, followed by collocations and then compositional items. Under a *dual-route* account, idioms and collocations—both stored, familiar units—should be processed similarly and faster than compositional phrases, which require computation. However, given prior findings, we predict that human participants will process idioms fastest, followed by compositional phrases, with collocations being the slowest—-even though collocations are often more frequent than compositional items in our dataset (see Appendix A).

Furthermore, based on the review in Section 2, it would be uncontroversial to say that memory is critical to all forms of language processing (see also Divjak, 2019; Divjak et al., 2022; Corballis, 2019). It encapsulates single-route processes and is an integral component of dual-route models. Building on this foundational role of memory, we simulate memory retrieval using a well-established frequency-based mechanistic model of memory—MINERVA2 (Hintzman, 1984), modified to incorporate two key factors known to influence collocational processing: frequency (Wolter and Gyllstad, 2013) and semantics (Gyllstad and Wolter, 2016; Fioravanti et al., 2021). We adopt a distributional semantic framework (Landauer and Dumais, 1997; Mikolov et al., 2013) and use contextualized embeddings from Sentence-BERT (Reimers and Gurevych, 2019). We modify MINERVA to simulate RTs and load its memory according to the frequency of the stimuli in the corpus. We explore successful and failed retrievals to assess their influence on the processing signatures of different item conditions under a pure memory-based model. Our central research question is:

To what extent can pure memory retrieval account for processing differences observed in L1 English speakers across idioms, collocations, and compositional phrases?

Here, we expect MINERVA2 to show differences across conditions primarily as a function of frequency, with no added processing cost for collocations. In any case, if retrieval alone were sufficient to account for human processing trends, the model should mirror the human patterns consistent with a single-route account. If not, the model will allow us to probe what aspects of human performance can be explained by memory alone, and where memory-based retrieval may fall short.

4 Collocations Incur a Processing Cost

4.1 Methodology

Stimuli de Souza et al. (2024) introduced a stimulus set consisting of 100 Verb-Noun collocations (e.g., *spill secrets*) and 100 compositional Verb-Noun combinations containing the same verb as the collocation (e.g., *spill water*). We attempted to augment this stimulus set with a matching figurative idiom (e.g., *spill the beans*) for each verb with the help of the ‘word sketch’ function in The Sketch Engine’s enTenTen21 corpus (Kilgarriff et al., 2024). However, we were only able to identify idioms for 82 verbs in the dataset resulting in a final dataset of 246 target items (1 collocation, one composition, and one idiom for each of the 82 verbs). 82 baseline items, nonsense Verb-Noun combinations (*fry knob*), were created to use as distractors in the experiment. The dataset was divided into 3 folds of 82 items wherein no two items had the same verb. As expected, there are statistically significant differences between the mean frequencies of all three constructions with idioms being the most frequent, followed by collocations and compositional items being the least frequent group (see Appendix A for more details). We account for this discrepancy by including frequency as a covariate in our statistical models.

Participants & Task A total of 186 L1 English speakers ($F = 112$; $M = 71$; $NB = 3$) were recruited using Prolific. They were remunerated £1.50 for their participation. The mean age of the sample was 38.6 years ($SD = 10.81$). They were asked to judge whether or not the word combination presented to them sounded acceptable (i.e., would they as L1 English speakers use this word combination in their everyday speech). They were asked to respond as quickly and accurately as possible, by pressing the 'y' key for yes or the 'n' key for no. During testing, each participant saw 164 items: 82 target items and 82 distractors. Items were presented in an individualized random order. A fixation cross with an inter-stimulus interval of 350 ms was presented between trials. Trials timed out at 8,000 ms if no decision was taken.

Data Pre-processing Data pre-processing was carried out using R version 4.4.1 "Race for Your Life" (R Core Team, 2024). Due to an error in data collection, data of four participants were replaced. We also remove all incorrect trials for reaction time analyses ($2,752$; see Appendix B). We then eliminated responses below 450 ms and responses over 3.5 standard deviation from the grand mean including time-outs. These outliers accounted for 1.484% of the total data ($n = 30,504$ including distractors). In terms of accuracy, all participants scored above 50%. However, we found 4 items with a mean accuracy of less than 50%. We eliminated those items along with other items that comprised the same verbs from our analyses. We do not analyse distractors ($15,252$). All reaction time (RT) analyses are conducted on this final dataset ($n = 13,369$).

4.2 Statistical Modelling

We first specified a maximal model as "justified by the design" (Barr et al., 2013). The main dependent variable was the reaction times (RTs) from the acceptability judgement task while the main predictor variable was Condition (Compositional, Collocation, Idiom; treatment coded, with idiom as the reference level). Phrasal Frequency (scaled) was included as a covariate. The maximal converging random effect structure included intercepts for Participant and Verb. The analysis model in R syntax specified using the 'lme4' (Bates et al., 2015) package is: `RT ∼ Condition + Phrasal Frequency + (1 | ID) + (1 | Verb)`.

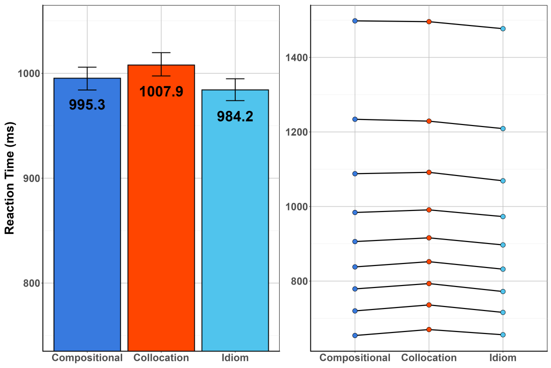


Figure 1: **Left:** mean reaction times (ms) by condition. Error bars indicate bootstrapped confidence intervals. **Right:** decile plot of reaction times by condition. Note the differences in the y-axes.

4.3 Results

Figure 1 shows the mean reaction times (RTs) by condition, as well as a breakdown by decile. Collocations have the slowest responses with a mean of 1007.87 ms ($SD = 370.84$ ms) compared to compositional items (995.32 ms, $SD = 375.76$ ms) and idioms (984.20 ms, $SD = 365.39$ ms).

Our statistical results showed a small, significant difference in RTs between compositional items and idioms ($\beta = 4.69; SE = 2.240; p = 0.037$), suggesting that compositional units were processed slower than idioms. A larger difference was found between collocations and idioms ($\beta = 13.80; SE = 1.760; p < 0.001$), replicating the processing costs predicted by the literature. Unsurprisingly, Phrasal Frequency also has a significant effect on RTs ($\beta = -18.50; SE = 1.640; p < 0.001$), corresponding to a 18.5 ms decrease in RT for every 1 standard deviation increase in phrasal frequency. In terms of accuracy, we found no significant difference between idioms and compositional items, but we do see a marginal difference ($p = 0.04$) between idioms and collocations. This is expected as all stimuli are highly frequent and should be familiar to adult L1 speakers. See Appendix B for detailed results.

5 Failures in Memory Retrieval Capture Behavioural Trends

As a first step toward elucidating the cognitive mechanisms underlying the processing trend that humans display across the compositionality continuum, we investigate the extent to which we can account for the trend with memory retrieval alone.

MINERVA is an instance-based model of

episodic memory that has been successfully applied to many cognitive phenomena from frequency judgements (Hintzman, 1988) to false memory (Arndt and Hirshman, 1998). It has also been used to model artificial grammar learning (Jamieson and Mewhort, 2009) and, recently, to metaphor recognition (Nick Reid and Jamieson, 2023).

MINERVA's core assumptions are: (i) every item encountered leaves a memory trace, represented as a distributed set of features, and (ii) similar items have similar traces. Similarities between present and past encounters drive item-specific and parallel memory retrieval. As a global memory model, it encapsulates both episodic and semantic memory which communicate with each other. On encountering a stimulus, the episodic memory sends a probe to the semantic memory to retrieve traces from past encounters. The familiarity of the probe is then calculated as the sum of the values of a similarity measure between the probe and each stored trace.

MINERVA is instantiated in a linear algebra system. The MINERVA memory $\mathbf{M}$ is an $n \times d$ matrix, each row of which contains a d-dimensional memory trace vector. When cued for retrieval with a probe $p \in \mathbb{R}^d$, MINERVA retrieves the representation of the probe iff the probe's familiarity f is greater than a threshold $K \in [0, 1)$. Familiarity is calculated by taking the cosine similarity s of the probe to all instances stored in memory, scaling s to reflect activation (weighting) of memory items a over elapsed time τ, and linearly combining instances in memory to compute a memory echo e. The familiarity score at timestep τ is the cosine similarity of the echo to the probe, following this system of equations:

$$s = \text{sim}(p, \mathbf{M}) \tag{1}$$

$$a_\tau = s^\tau \text{sign}(s) \tag{2}$$

$$e_\tau = a_\tau \mathbf{M} \tag{3}$$

$$f_\tau = \text{sim}(e_\tau, p) \tag{4}$$

Modelling AJT Responses with Taus (τ) The free parameter τ is used to accentuate differences in similarity values (Hintzman, 1988; Nick Reid and Jamieson, 2023). By raising the value of τ, higher-similarity memory traces will elicit exponentially more activation, allowing those traces to play a larger role in the overall activation profile versus pooling a potentially large number of low-similarity items.

Following Nick Reid and Jamieson (2023), we depart from prior work wherein τ is kept constant

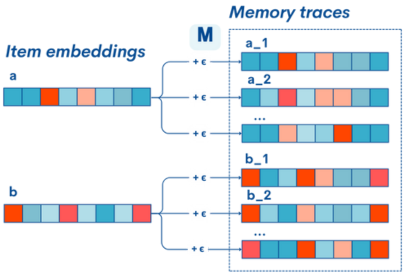


Figure 2: Illustration of how embeddings are noised and loaded into MINERVA's memory matrix M. Colors depict values within a vector. Note that the noise vectors ϵ are independently sampled for each memory trace.

for a particular experiment and model reaction times by dynamically increasing τ for a particular probe p until a desired threshold of familiarity $K \in [0, 1)$ is reached. At this point, we take the final value of τ as a proxy for the time required to recognize p from memory, i.e, a proxy for reaction time (RT). We set a time-out at $\tau = 300$ after which the next probe is presented.

In human acceptability judgements, reaction times serve as a proxy for processing difficulty. We implicitly model acceptability judgements in MINERVA as a function of whether the familiarity threshold K is reached within the allowable time window. If the familiarity score surpasses K before the time-out, i.e., successful recognition, we treat this as a "yes". Conversely, if familiarity remains below the threshold when $\tau = 300$, we treat the failure to retrieve as a "no" response.

5.1 Motivations & Assumptions

Collocational processing is known to be driven by two factors: *semantic transparency* and *frequency* (see Gyllstad and Wolter, 2016; Fioravanti et al., 2021). Our model captures semantic transparency by means of distributional semantics, i.e, vector embeddings, while frequency is captured by means of phrasal frequency in a dynamic web corpus. We demonstrate the effect of both factors in our ablations (see Section 5.4).

Semantics of Memory Traces Using distributed vector representations as memory traces for MINERVA is well-established in the literature (Chubala and Jamieson, 2013; Jamieson et al., 2018; Nick Reid and Jamieson, 2023). Given that the figurative idioms (e.g., *spill the beans*) also have a compositional reading, we need a contextualized, fine-grained vector representation to capture the semantics of each word combination. Therefore, we rely on Sentence-BERT (sBERT) which provides semantically meaningful vector embeddings for sentences (Reimers and Gurevych, 2019). To derive the vector embedding for each of the 246 target stimuli, we follow Vulić et al. (2020). First, we collect a set of 100 sentences of the word combination[5] from the enTenTen21 corpus, in which the noun occurs as the direct object of the verb. We feed each sentence to sBERT obtaining a set of contextualized word embeddings representing each word in the sentence (we perform mean pooling over sub-words). Given that the higher layers of BERT architectures are the most sensitive to lexical semantics (Reif et al., 2019), we take our embeddings from the last hidden layer of the model. From each of the 100 sentences, we extract the embeddings corresponding to the verb and the noun and average across them separately, resulting in the mean contextualized representation of the verb when paired with the noun, and of the noun when paired with the verb. Finally, we concatenate the mean embedding for the verb with the mean embedding for the noun to form the vector representation of our stimulus[6].

Memory Frequencies & Forgetting In accordance with the instance theory, MINERVA's retrieval time is inversely proportional to the number of memory traces that strongly respond to a particular probe (Nick Reid and Jamieson, 2023). Therefore, we populate MINERVA's memory matrix using $10,000$ items sampled proportionally to their phrasal frequency. Following prior work, we simulate forgetting by adding zero-centered Gaussian noise to each memory trace vector such that each dimension of each trace has an independent probability $F \in [0, 1)$ of being corrupted with noise. The more frequent a particular item, the more traces it will have in memory, averaging out the noise and making high-frequency items easier to retrieve.

5.2 Simulations

To explore the extent to which simple memory retrieval is sufficient to reproduce processing trends for each condition, we load the memory matrix as described above (see Figure 2) and test MINERVA's recognition capabilities using a noiseless vector embedding of the target stimulus as the probe. To simulate N different participants who are exposed to different samplings of items from the same environmental distributions, as well as different patterns of forgetting, we run each simulation N=300 times with different random seeds, re-sampling and re-noising the memory matrix each time. We perform a thorough hyperparameter sweep of activation threshold K and forgetting probability F. We discuss results for hyperparameter values K=0.99 and F=0.8, although our results are robust across many hyperparameter combinations (see Figure 10).

We use the same statistical model described in Section 4 to analyse the effect of semantics and frequency on retrieval (i.e., Tau).

5.3 Results

The results of our computational experiment are shown in Figure 3. As MINERVA was not presented with any baseline items and as all items were in MINERVA's memory, it should have succeeded at recognizing all items (Figure 3, left panel). Thus, we first considered only successful retrievals. Despite being provided with meaningful embeddings and frequencies, the model failed to capture human processing trends. Collocations were retrieved faster than idioms ($\beta = -0.41; SE = 0.004; p < 0.001$) while compositional items were retrieved slower than idioms ($\beta = 0.62; SE = 0.004; p < 0.001$). See Appendix C for more details. Given the surprising results, we analyzed the model's failures to retrieve, i.e., timeouts, on every item (see Figure 3, right panel). MINERVA timed out on 50% of the retrievals for collocations, followed by compositional items (38.6%), with idioms timing out the least (33.8%). A mixed-effects logistic regression confirmed all differences between conditions to be statistically significant[7] (see Table 6).

---

[5] Distractor items were not included in the simulations as they are nonsense combinations, have no context sentences and would have very low frequency in MINERVA's memory.

[6] We use concatenation instead of mean pooling as our stimuli are all Verb + Direct Object and concatenation preserves word order and therefore, syntactic role information. However, see Appendix E.

[7] To rule out the possibility that these results are a quirk of the MINERVA architecture, we also ran memory retrieval simulations using the same memory matrix on the Modern Hopfield Network (Ramsauer et al., 2021, MHN). The MHN is a generalization of the classical Hopfield network (Hopfield, 1982) adapted to work with continuous states, and is formally connected to the QKV attention mechanism in Transformers. We find that the MHN displays the same characteristic pattern of failures in at least one configuration of hyperparameters (see Appendix D). Although our MHN results are a proof-of-concept, given the robustness of the MINERVA findings with respect to hyperparameters (see Figure 10) they suggest that elevated failure rates on collocations may be a property of associative memory retrieval writ large.

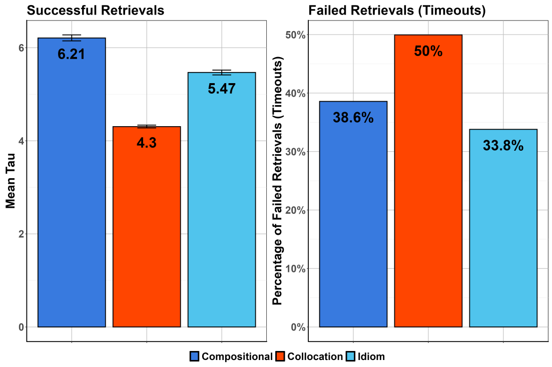


Figure 3: **Left:** mean Tau (τ) by condition for successful retrievals in MINERVA. The y-axis represents mean Tau, the model's output which acts as a proxy for reaction times. Error bars indicate bootstrapped confidence intervals. **Right:** percentage of failed retrievals, i.e., timeouts, per condition. Note that while the pattern of Taus on *successful retrievals* is different from the pattern of human RTs, the pattern of *timeouts* per condition matches the pattern of human RTs.

Unlike the pattern of Taus on successful retrievals, the pattern of retrieval failures in MINERVA appears to capture the trend in human RTs across the three conditions.

Additionally, we found that MINERVA always succeeds at retrieving items above a high frequency threshold (Figure 4, black line). We find a similar frequency boundary in humans (Figure 4, green line), which lies very close to the MINERVA threshold. On items above this threshold[8], participants did not show a significant difference in RT by condition, while still showing a significant effect of frequency.

5.4 Ablations

Semantics-only In the semantics-only ablation, the model was loaded with all instances being equally frequent. Thus, the only distinguishing factor between the memory traces were their semantics. The results are shown in Figure 5. We visually observe that the trends for collocations match those of the main experiment—they time out the most, but are retrieved the fastest on successful retrievals. However, unlike in the main experiment, we observe that compositional items are retrieved slightly faster and time out less frequently than idioms. Investigating the cause of this discrepancy is an interesting avenue for future work.

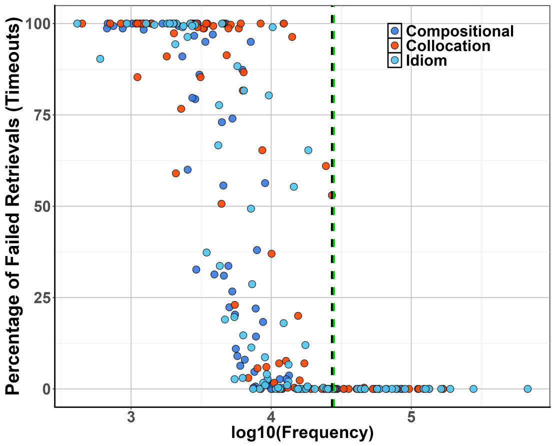


Figure 4: Percentage of failed retrievals (i.e., timeouts) in MINERVA per stimulus item, as a function of the frequency of the item. The x-axis is displayed in log scale. The black line indicates the frequency threshold ($f = 27123$) above which MINERVA times out less than 1% of the time. The green line ($f = 28000$) indicates the frequency threshold above which condition stops being a significant predictor of human RTs.

Frequency-only In the frequency-only ablation, the model was loaded with embeddings comprised of Gaussian noise[9]. However, each noise-item was sampled according to correct frequency information. The results are shown in Figure 6. For successful retrievals, we visually observed that idioms and collocations were retrieved equally quickly, whereas compositional items were retrieved slower. This pattern also persists in the timeouts. Given that frequency drives MINERVA's retrieval mechanism, this pattern of Taus and timeouts is not surprising. Idioms—the most frequent subset—are retrieved most easily, followed by collocations, and finally compositional units which are the least frequent.

The results of these ablations suggest that it is the semantics of the item traces that drive the unique processing cost for collocations in MINERVA. Additionally, as shown with the trends for idioms vs compositional items, frequency dynamics mitigate the effects of semantic dynamics, producing the overall behaviour observed in the main experiment.

6 Discussion & Analysis

Our behavioural results confirm the processing cost for collocations surmised from the literature. This effect occurs despite collocations and compositional items being very close in frequency (with the balance in favour of collocations), and the par-

---

[8] 16 compositional, 18 collocations, 17 idioms

[9] We calibrate the noise to the mean and standard deviation of the embeddings in the main condition.

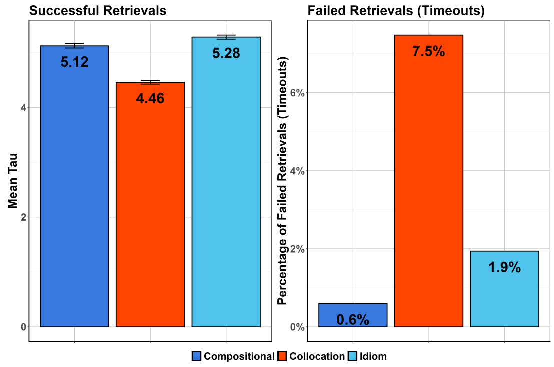


Figure 5: **Left:** mean Tau (τ) by condition for successful retrievals in Ablation 1, wherein frequency information was eliminated. The y-axis represents mean Tau, the model's output which acts as a proxy for reaction times. Error bars indicate bootstrapped confidence intervals. **Right:** percentage of failed retrievals, i.e., timeouts, per condition in Ablation 1. Note that just as in the main MINERVA experiment, collocations time out much more frequently than the other conditions.

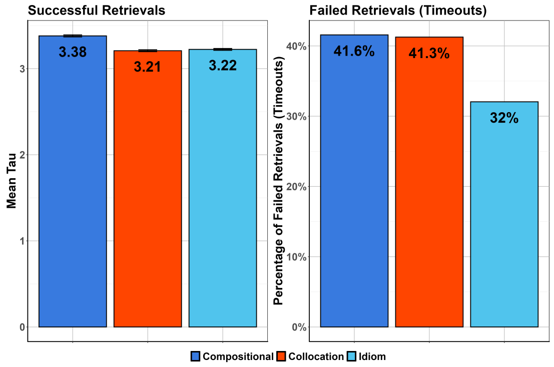


Figure 6: **Left:** mean Tau (τ) by condition for successful retrievals in Ablation 2, wherein semantic information was eliminated while leaving the correct item frequency distribution. The y-axis represents mean Tau, the model's output which acts as a proxy for reaction times. Error bars indicate bootstrapped confidence intervals. **Right:** percentage of failed retrievals, i.e., timeouts, per condition. Note that the trends in timeouts follow the frequency distribution across the conditions.

ticipants as adult L1 English speakers being highly familiar with the items. The result is mirrored in our computational findings. These stark differences in processing patterns for collocations compared to idioms and compositional items suggest that they must be treated as a separate class of linguistic items, and not be cast under the broad umbrella of formulaic language.

To recap our simulation results, MINERVA's successful retrievals failed to reproduce the processing trend observed in humans and also exhibited many more incorrect responses, i.e., unsuccessful memory retrievals, than humans. However, these retrieval failures do appear to capture the key asymmetries in human processing. Again, this is especially noticeable for collocations on which MINERVA timed out the most. We also found that above a certain frequency threshold, MINERVA matches human patterns. This suggests that simple memory retrieval, as implemented in a frequency based model of memory, is only sufficient to explain human processing trends for highly frequent items. Below this threshold, retrieval starts to fail. Given that MINERVA does not have any processing mechanism beyond memory retrieval, it simply times out on these items. We conjecture that at this point humans invoke other processing mechanisms to facilitate interpreting of the stimulus, incurring a cost in reaction time.

The fact that collocations incur a higher processing cost despite being more frequent than compositional items shows that single-route accounts provide an incomplete picture. They further demonstrate that dual-route accounts with a binary distinction between formulaic versus compositional language are also insufficient to account for the processing of this large and frequent subset of language. This underscores the need for a model which can account for a more fine-grained representation of semantic compositionality. One such plausible mechanism is analogical reasoning (Eddington, 2000; Ambridge, 2020). Like single-route models, this domain-general approach posits that all linguistic units are processed by a single mechanism (Skousen, 1990). However, in addition to memory retrieval, it posits on-the-fly analogy without resorting to any rule-based mechanisms. On receiving an input, a memory search is undertaken to find analogous exemplars previously experienced. The input is then evaluated based on the degree of similarity in order to find the most frequent category within the found set of most similar exemplars (Gentner and Namy, 2006).

Memory retrieval is the first step in analogical processing (Gentner and Colhoun, 2010). Thus, processing a sufficiently frequent item via analogy will simply resort to memory retrieval. Such a mechanism would be invariant to the semantic compositionality of the item in question, as we have seen in humans. Below this threshold, however, proper analogical machinery comes into play.

In compositional items, both the verb and the noun play a prototypical role. Thus, even though the language user may not recall this exact verb-noun pairing from memory, it is relatively easy to map the verb and noun to similar instances of the same, due to the high semantic overlap between compositional uses of the verb and the noun. In collocations, however, the verb is not used in its prototypical sense. Resolving the meaning of the verb requires a much "farther" mapping, which may involve increased search over possible abstractions of the verb or extensive structure-mapping. Engaging such machinery inevitably incurs a processing cost with respect to compositional items (Gentner and Namy, 2006), as reflected in RTs. Finally, idioms, which cannot be processed analytically, must be retrieved holistically irrespective of frequency.

Moreover, there is a body of evidence for the role of analogy in metaphor comprehension (see Morsanyi et al., 2022, for a review), child language acquisition (see Raynal et al., 2024), and processing of novel verb metaphors (King and Gentner, 2022), which are, in essence, unconventionalized collocations. We posit that that an analogical account of language processing may provide a more complete explanation of these findings, and that further work should explore this proposal.

The retrieval failures for idioms may stem from a limitation of our dataset—the fact that we only consider figurative idioms which have a compositional reading. We were unable to ascertain the relative frequency of idiomatic versus literal readings in the context sentences of every idiom in our stimuli that we use to generate embeddings. It is also unknown to what precise extent sBERT can accurately represent idiomatic meanings, or whether our human participants interpreted idiomatic stimuli in a figurative sense. Combined, these factors suggest that the semantics of our set of idioms are somewhat akin to our set of compositional items, and some of the processing trends which pertain to compositional items are inadvertently present in the trend of responses to idioms. In line with the holistic retrieval hypothesis, we surmise that idioms for which the literal reading is much less frequent than the idiomatic one (e.g., *kick the bucket*) will tend to be processed faster and with fewer timeouts than more ambiguous ones (e.g., *hold the key*). Future work will attempt to investigate this prediction and further augment our understanding of idiomatic processing by including pure idioms, i.e., those without a literal reading, in the dataset, and employing other behavioural tasks which involve presentation of items within context (e.g., self-paced reading).

One intriguing implication of our computational experiment may be of interest to the NLP community. Switching Equation 2 for $a_\tau = \text{softmax}(\tau s)$, MINERVA's retrieval mechanism becomes identical to query-key-value attention in Transformers (Vaswani et al., 2017). Here, the probe plays the role of the query and the memory items the role of keys and value. Increasing Tau can be considered as a mechanism for dynamically weighting the output toward the keys which bear most similarity to the query, similar to increasing the softmax scale parameter over time. Under this formulation, MINERVA is also a variant of the Modern Hopfield Network with no learnable parameters (Ramsauer et al., 2021). This wealth of connections suggests that our findings may apply more broadly to all attention-based methods. Given the prevalence of collocations in language, if neural embeddings of semi-compositional language are particularly prone to failures in attention-based retrieval, this may significantly impair language understanding and generation in Transformer-based models. Future work will attempt to mechanistically diagnose the underlying reasons for the increased failure rates in collocations and ascertain whether these issues impact the performance of general-purpose Transformer-based language models.

On a more applied level, large language models (LLMs) as writing assistants have gained popularity (Boisson et al., 2024). This has drawn attention to how these models handle figurative language like metaphors, of which collocations are a prominent subset. While LLMs can produce metaphors, users often note shortcomings such as clichéd phrasing or a lack of creativity in metaphor generation (Chakrabarty et al., 2024), perhaps reflecting an underlying lack of capability in interpreting these linguistic units. These observations further underscore the importance of understanding collocational processing not only in human cognition, but also in NLP systems.

Overall, we show that semi-compositional units are a bigger "pain in the neck" (Sag et al., 2002) than other subsets of the semantic compositionality continuum: too complex for rote retrieval, yet too idiosyncratic for rule-based computation. As it stands, memory retrieval does *leave something on the table*, underscoring the need for theories that capture the graded nature of meaning and structure in language.

7 Limitations

- Our approach relies on contextual embeddings to capture semantic information. However, these embeddings do not always differentiate clearly between compositional and idiomatic readings. Given that our idiomatic stimuli also have a productive reading, the same embedding may be used for both literal and figurative interpretations. Similarly, we cannot ensure that our task is eliciting an idiomatic reading in humans as human listeners disambiguate based on context.

- The current dataset was not built from scratch with frequency-matching criteria for idioms. Frequency is a well-established predictor of language processing and an ideal dataset would equate or carefully control the frequency distributions of idioms relative to other word types.

- Our study exclusively examined verb–noun (VN) collocations. While these are a critical class of multiword expressions, little is known about other collocational structures (e.g., adjective–noun, phrasal verbs, etc.) which are also prevalent in natural language and may be processed differently. Extending our investigation to these additional types will be important for assessing the generalizability of our findings across the broader spectrum of semi-compositional linguistic units.

- MINERVA provides a parsimonious framework for modelling memory retrieval, yet it inherently simplifies many aspects of human cognitive processing. The model does not integrate attentional mechanisms or dynamic contextual cues beyond the static embeddings provided, and it does not account for developmental changes in memory and language processing. These simplifications may limit the model's ability to capture the full complexity of human language processing, particularly in cases where retrieval failures (time-outs) interact with other cognitive processes. Our simulations relied on specific hyperparameter settings (e.g., activation threshold K=0.99 and forgetting probability F=0.8) that were chosen based on qualitative assessments. Although results were robust across a range of parameter values, the possibility remains that different parametrizations could yield different patterns.

Ethics

The study received ethics approval in accordance with the University of Edinburgh's School of Philosophy, Psychology and Language Sciences Research Ethics Process (RT number: 339-2122/4).

Data and Code

Our data and code will be made available at the following URL: `https://github.com/psydelle/minerva-release`

Acknowledgements

This work was primarily supported by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics and School of Philosophy, Psychology & Language Sciences.

SdS is also grateful for the support of the Fundação para a Ciência e a Tecnologia, under project UID/00214: Centro de Linguística da Universidade de Lisboa, of Center of Linguistics of the University of Lisbon, from School of Arts and Humanities of the University of Lisbon.

We are also grateful to Sivan Milton for checking our stimuli set, and Mattia Opper and Dr Edoardo Ponti for advice on adapting sBERT embeddings for our experiment.

References

Ben Ambridge. 2020. Against stored abstractions: A radical exemplar model of language acquisition. 40(5-6):509–559.

Ben Ambridge and E. Lieven. 2011. *Child language acquisition: Contrasting theoretical approaches*. Cambridge University Press, Cambridge.

Jason Arndt and Elliot Hirshman. 1998. True and False Recognition in MINERVA2: Explanations from a Global Matching Perspective. *Journal of Memory and Language*, 39(3):371–391.

Andy Barfield and Henrik Gyllstad, editors. 2009. *Researching Collocations in Another Language*. Palgrave Macmillan UK, London.

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. Publisher: Elsevier Inc.

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.

Joanne Boisson, Asahi Ushio, Hsuvas Borkakoty, Kiamehr Rezaee, Dimosthenis Antypas, Zara Siddique, Nina White, and Jose Camacho-Collados. 2024. How Are Metaphors Processed by Language Models? The Case of Analogies. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 365–387, Miami, FL, USA. Association for Computational Linguistics.

Joan Bybee. 2006. From usage to grammar: The mind's response to repetition. *Language*, 82(4).

Joan Bybee. 2012. A usage-based perspective on language. In *Language, Usage and Cognition*, pages 1–13. Cambridge University Press, Cambridge.

Gareth Carrol and Kathy Conklin. 2014. Getting your wires crossed: Evidence for fast processing of L1 idioms in an L2. *Bilingualism: Language and Cognition*, 17(4):784–797. Publisher: Cambridge University Press.

Gareth Carrol and Kathy Conklin. 2020. Is All Formulaic Language Created Equal? Unpacking the Processing Advantage for Different Types of Formulaic Sequences. *Language and Speech*, 63(1):95–122.

Tuhin Chakrabarty, Vishakh Padmakumar, Faeze Brahman, and Smaranda Muresan. 2024. Creativity Support in the Age of Large Language Models: An Empirical Study Involving Emerging Writers. *arXiv preprint*. ArXiv:2309.12570 [cs].

Morten H. Christiansen and Nick Chater. 2008. Language as shaped by the brain. *Behavioral and Brain Sciences*, 31(5):489–509.

Chrissy M. Chubala and Randall K. Jamieson. 2013. Recoding and representation in artificial grammar learning. *Behavior Research Methods*, 45(2):470–479.

Michael C. Corballis. 2019. Language, Memory, and Mental Time Travel: An Evolutionary Perspective. *Frontiers in Human Neuroscience*, 13. Publisher: Frontiers.

Anthony Paul Cowie. 1998. *Phraseology : theory, analysis, and applications*. Clarendon Press, Oxford. Series Title: Oxford studies in lexicography and lexicology.

Peter W. Culicover, Ray Jackendoff, and Jenny Audring. 2017. Multiword Constructions in the Grammar. *Topics in Cognitive Science*, 9(3):552–568.

Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2022. Scientific and Creative Analogies in Pretrained Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2094–2100, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ishita Dasgupta and Samuel J. Gershman. 2021. Memory as a Computational Resource. *Trends in Cognitive Sciences*, 25(3):240–251.

Sydelle de Souza, Francis Mollica, and Jennifer Culbertson. 2024. What can L1 speakers tell us about killing hope? A Novel Behavioral Measure for Identifying Collocations. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 46(0).

Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. 2017. On a Model of Associative Memory with Huge Storage Capacity. *Journal of Statistical Physics*, 168(2):288–299.

Dagmar Divjak. 2019. *Frequency in Language: Memory, Attention and Learning*. Cambridge University Press, Cambridge.

Dagmar Divjak, Petar Milin, Srdan Medimorec, and Maciej Borowski. 2022. Behavioral Signatures of Memory Resources for Language: Looking beyond the Lexicon/Grammar Divide. *Cognitive Science*, 46(11):e13206.

Philip Durrant and Norbert Schmitt. 2009. To what extent do native and non-native writers make use of collocations? *IRAL - International Review of Applied Linguistics in Language Teaching*, 47(2):157–177.

David Eddington. 2000. Analogy and the dual-route model of morphology. *Lingua*, 110(4):281–298.

Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. BOHB: robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1436–1445. PMLR.

Irene Fioravanti, Marco Silvio Giuseppe Senaldi, Alessandro Lenci, and Anna Siyanova-Chanturia. 2021. Lexical fixedness and compositionality in L1 speakers' and L2 learners' intuitions about word combinations: Evidence from Italian. *Second Language Research*, 37(2):291–322.

Gottlob Frege. 1892. *Über Sinn und Bedeutung*, 1. auflage edition. Zeitschrift für Philosophie und philosophische Kritik, Neue Folge. Pfeffer, Leipzig.

Lukas Galke, Yoav Ram, and Limor Raviv. 2024. Deep neural networks and humans both benefit from compositional language structure. *Nature Communications*, 15(1):10816. Publisher: Nature Publishing Group.

Dedre Gentner and Julie Colhoun. 2010. Analogical Processes in Human Thinking and Learning. In Britt Glatzeder, Vinod Goel, and Albrecht Müller, editors, *Towards a Theory of Thinking: Building Blocks for a Conceptual Framework*, pages 35–48. Springer, Berlin, Heidelberg.

Dedre Gentner and Laura L. Namy. 2006. Analogical Processes in Language Learning. *Current Directions in Psychological Science*, 15(6):297–301. Publisher: SAGE Publications Inc.

Lynn E. Grant. 2005. Frequency of 'core idioms' in the British National Corpus (BNC). *International Journal of Corpus Linguistics*, 10(4):429–451. Publisher: John Benjamins.

Henrik Gyllstad and Brent Wolter. 2016. Collocational Processing in Light of the Phraseological Continuum Model: Does Semantic Transparency Matter? *Language Learning*, 66(2):296–323.

Susanne Handl and Eva-Maria Graf. 2010. Collocation, anchoring, and the mental lexicon – an ontogenetic perspective. In Hans-Jörg Schmid and Susanne Handl, editors, *Cognitive Foundations of Linguistic Usage Patterns: Empirical Studies*, pages 119–148. De Gruyter Mouton.

Douglas L. Hintzman. 1984. MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers*, 16(2):96–101.

Douglas L. Hintzman. 1988. Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review*, 95(4):528.

J J Hopfield. 1982. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558.

Peter Howarth. 1998. Phraseology and second language proficiency. *Applied Linguistics*, 19(1):24–44.

Randall K. Jamieson, Johnathan E. Avery, Brendan T. Johns, and Michael N. Jones. 2018. An Instance Theory of Semantic Memory. *Computational Brain & Behavior*, 1(2):119–136.

Randall K. Jamieson and D. J.K. Mewhort. 2009. Applying an exemplar model to the artificial-grammar task: Inferring grammaticality from similarity. *Quarterly Journal of Experimental Psychology*, 62(3):550–575.

Adam Kilgarriff, Pavel Rychly, Pavel Smrz, and David Tugwell. 2024. The Sketch Engine.

Daniel King and Dedre Gentner. 2022. Verb Metaphoric Extension Under Semantic Strain. *Cognitive Science*, 46(5):e13141. Publisher: John Wiley & Sons, Ltd.

Nate Kornell and Robert A. Bjork. 2009. A stability bias in human memory: Overestimating remembering and underestimating learning. *Journal of Experimental Psychology: General*, 138(4):449–468.

Dmitry Krotov and John J. Hopfield. 2016. Dense associative memory for pattern recognition. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1180–1188, Red Hook, NY, USA. Curran Associates Inc.

George Lakoff and Mark Johnson. 1980. *Metaphors We Live By*. University of Chicago Press, Chicago.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

Ronald W Langacker. 1987. *Foundations of cognitive grammar*. Stanford University Press, Stanford, Calif.

Emmy Liu, Chenxuan Cui, Kenneth Zheng, and Graham Neubig. 2022. Testing the Ability of Language Models to Interpret Figurative Language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4437–4452, Seattle, United States. Association for Computational Linguistics.

Gordon D. Logan. 1997. Automaticity and Reading: Perspectives from the Instance Theory of Automatization. *Reading & Writing Quarterly*, 13(2):123–146.

Gordon D Logan and Joseph L Etherton. 1994. What Is Learned During Automatization? The Role of Attention in Constructing an Instance. *Journal of Experimental Psychology: Learning Memory, and Cognition*, 20(5):1022–1050.

James L McClelland and David E Rumelhart. 1985. On learning the past tense of English verbs. In J. L. McClelland and D. E. and the PDP Research Group Rumelhart, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, volume 2. Bradford Books/MIT Press, Cambridge, MA.

Igor Mel'čuk. 2003. Collocations: définition, rôle et utilité. *Travaux et recherches en linguistique appliquée. Série E, Lexicologie et lexicographie.*, (1):23–31. Num Pages: 9 Place: Amsterdam Publisher: Editions 'De Werelt'.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Kinga Morsanyi, Jayne Hamilton, Dušan Stamenković, and Keith J. Holyoak. 2022. Linking metaphor comprehension with analogical reasoning: Evidence from typical development and autism spectrum disorder. *British Journal of Psychology*, 113(2):479–495.

Rachel A. G. Mueller and Raymond W. Gibbs. 1987. Processing idioms with multiple meanings. *Journal of Psycholinguistic Research*, 16(1):63–81.

J. Nick Reid and Randall K. Jamieson. 2023. True and false recognition in MINERVA 2: Extension to sentences and metaphors. *Journal of Memory and Language*, 129:104397. Publisher: Elsevier Inc.

Tomomi Nishikawa. 2019. Non-nativelike outcome of naturalistic child L2 acquisition of Japanese: The case of noun–verb collocations. *International Review of Applied Linguistics in Language Teaching*, (Lenneberg 1967).

Ira A. Noveck, Nicholas Griffen, and Diana Mazzarella. 2023. Taking stock of an idiom's background assumptions: an alternative relevance theoretic account. *Frontiers in Psychology*, 14.

Timothy J O'Donnell, Noah D Goodman, and Joshua B Tenenbaum. 2009. Fragment Grammars : Exploring Computation and Reuse in Language Fragment Grammars. *Computer Science and Artificial Intelligence Laboratory Technical Report*.

Andrew Pawley and Frances Hodgets Syder. 1983. Two puzzles for linguistic theory: Nativelike selection and nativelike fluency. *Language and Communication*, pages 191–226. ISBN: 9781317869634.

Steven Pinker. 1991. Rules of Language. In *Science*, volume 253, pages 530–535. American Association for the Advancement of Science. Issue: 5019.

Steven Pinker. 2013. *Learnability and Cognition, New Edition: The Acquisition of Argument Structure*. MIT Press, Cambridge, MA.

Steven Pinker and Michael Ullman. 2002. The past and future of the past tense. *Trends in Cognitive Sciences*, 6(11):456–463.

R Core Team. 2024. R: a language and environment for statistical computing. manual, R Foundation for Statistical Computing, Vienna, Austria.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David P. Kreil, Michael K. Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. 2021. Hopfield Networks is All You Need. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Lucas Raynal, Evelyne Clément, Louise Goyet, Pia Rämä, and Emmanuel Sander. 2024. Neural correlates of unconventional verb extensions reveal preschoolers' analogical abilities. *Journal of Experimental Child Psychology*, 246:105984.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv preprint*. ArXiv:1908.10084 [cs].

Nick Riches, Carolyn Letts, Hadeel Awad, Rachel Ramsey, and Ewa Dąbrowska. 2022. Collocational knowledge in children: a comparison of English-speaking monolingual children, and children acquiring English as an Additional Language. *Journal of Child Language*, 49(5):1008–1023.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2276:1–15. ISBN: 3540432191.

Mark S. Seidenberg and David C. Plaut. 2014. Quasiregularity and Its Discontents: The Legacy of the Past Tense Debate. *Cognitive Science*, 38(6):1190–1228.

Diana Sidtis. 2020. Familiar Phrases in Language Competence. In Alexander Haselow and Gunther Kaltenböck, editors, *Grammar and Cognition : Dualistic Models of Language Structure and Language Processing*, pages 38–67. John Benjamins Publishing Company, Amsterdam, Philadelphia.

Royal Skousen. 1990. *Analogical Modeling of Language*. Springer Netherlands, Dordrecht.

Suhad Sonbul, Dina Abdel Salam El-Dakhs, and Rezan Alharbi. 2024. Rendering natural collocations in a translation task: The effect of direction, congruency, semantic transparency, and proficiency. *International Journal of Applied Linguistics*, 34(1):117–133.

Patrizia Tabossi, Rachele Fanari, and Kinou Wolf. 2008. Processing idiomatic expressions: Effects of semantic compositionality. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 34(2):313–327.

Xiaoyu Tong, Ekaterina Shutova, and Martha Lewis. 2021. Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4673–4686, Online. Association for Computational Linguistics.

Mei-Hsing Tsai. 2020. Teaching L2 collocations through concept-based instruction: The effect of L2 proficiency and congruency. *International Journal of Applied Linguistics*, 30(3):553–575.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009. ArXiv: 1706.03762.

Laura Vilkaite and Norbert Schmitt. 2019. Reading collocations in an L2: Do collocation processing benefits extend to non-adjacent collocations? *Applied Linguistics*, 40(2):329–354.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Dian Wang, Yang Li, Suge Wang, Xin Chen, Jian Liao, Deyu Li, and Xiaoli Li. 2025. CKEMI: Concept knowledge enhanced metaphor identification framework. *Information Processing & Management*, 62(1):103946.

Brent Wolter and Henrik Gyllstad. 2013. Frequency of Input and L2 Collocational Processing: A Comparison of Congruent and Incongruent Collocations. *Studies in Second Language Acquisition*, 35:451–482.

Alison Wray. 2002. *Formulaic Language and the Lexicon*, volume 80. Cambridge University Press, Cambridge. Publication Title: Language ISSN: 1535-0665.

Alison Wray. 2008. *Formulaic language: pushing the boundaries*. Oxford University Press, Oxford. Series Title: Oxford Applied Linguistics.

Satoshi Yamagata, Tatsuya Nakata, and James Rogers. 2023. Effects of distributed practice on the acquisition of verb-noun collocations. *Studies in Second Language Acquisition*, 45(2):291–317.

Junko Yamashita. 2018. Possibility of semantic involvement in the L1-L2 congruency effect in the processing of L2 collocations. *Journal of Second Language Studies*, 1(1):60–78.

Omnia Zayed, John Philip McCrae, and Paul Buitelaar. 2018. Phrase-Level Metaphor Identification Using Distributed Representations of Word Meaning. In *Proceedings of the Workshop on Figurative Language Processing*, pages 81–90, New Orleans, Louisiana. Association for Computational Linguistics.

A Dataset Statistics

Table 1: Descriptive statistics of phrasal frequency by condition

Condition	Mean	SD	N
Compositional	19374.47	30671.53	78
Collocation	21528.21	30971.42	78
Idiom	36784.68	87468.40	78

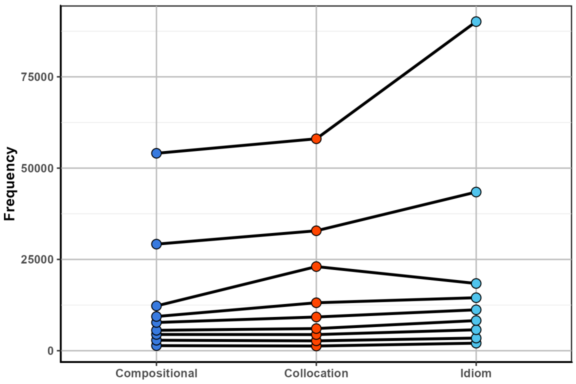


Figure 7: Item frequencies across conditions, by decile

B Human Data

Table 2: Descriptive statistics of human reaction times (ms) by condition

Condition	Mean	SD	N
Idiom	984.20	365.39	4462
Compositional	995.32	375.76	4423
Collocation	1007.87	370.84	4484

Table 3: Descriptive statistics of human accuracy by condition

Condition	Mean	SD	N
Idiom	0.93	0.25	4785
Compositional	0.92	0.27	4791
Collocation	0.94	0.24	4772

Table 4: Number of incorrect trials by condition

Condition	n
Compositional	400
Collocation	464
Idiom	433
Baseline	1455

C GLMM Results for Main Simulation

Table 5: Generalized mixed-effects regression results for human AJT reaction times (left), and Tau, a proxy for reaction times, simulated in MINERVA (right). MINERVA is run with $K = 0.99, F = 0.8$. Only correct responses and succesfull retrievals are analysed.

	Dependent variable:	
	RT	Tau
	Human	MINERVA
Compositional	4.690** (2.240)	0.624*** (0.004)
Collocation	13.800*** (1.760)	-0.410^{***} (0.004)
Frequency	-18.500^{***} (1.640)	-0.541^{***} (0.004)
Constant	1,047.0*** (2.140)	5.900*** (0.004)
N	13,369	43,708
Note:	*p<0.1; **p<0.05; ***p<0.01	

D Modern Hopfield Network Experiments

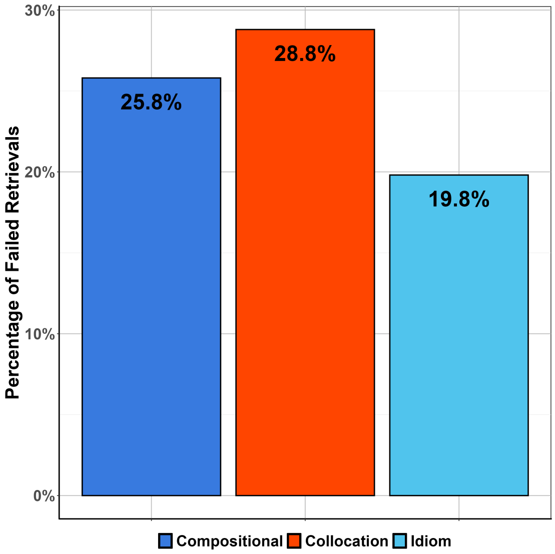


Figure 8: Percentage of failed retrievals in the Modern Hopfield Network. Collocations fail the most, followed by compositional items, and idioms fail the least. This matches the trends for the failures by condition in MINERVA, and the trend of reaction times in humans.

We suggest that our computational results pertain to the algorithmic implications of memory retrieval and are not based on a spurious quirk of the MINERVA architecture. To reinforce this claim, we present a proof-of-concept simulation of the same behaviour with the Modern Hopfield Network (Ramsauer et al., 2021, MHN).

The classical Hopfield network (Hopfield, 1982) is a model of associative memory based on binary states, designed for pattern completion and associative memory retrieval. MHN generalizes the Hopfield network and the recent iterations thereof (Krotov and Hopfield, 2016; Demircigil et al., 2017) with continuous states, and introduces a corresponding update rule which is closely connected to the query-key-value attention operation in Transformers.

The MHN can be configured in a number of different ways in order to simulate different functions, such as pattern matching, sequence pooling, and attention. In this simulation, we focus on the configuration denoted as `HopfieldLayer` in Ramsauer et al. (2021). Specifically, the model learns a static memory matrix of size $M \in \mathbb{R}^{n \times c}$, where n is the number of memory items and c is the dimensionality of the memory's latent space. It also learns projection matrices $A \in \mathbb{R}^{d \times c}, B \in \mathbb{R}^{c \times d}$ which

Table 6: Logistic regression results for retrieval failures. Reference Level is Idiom. **Left column:** MINERVA is run with $K = 0.99, F = 0.8$. **Right column:** Modern Hopfield Network for parameters presented in Appendix D. Both models converged but with singular fits. This was due to the `(1|ID)` random intercept (where ID is the random seed for the model run) accounting for nearly 0 variance.

	Dependent variable:	
	Retrieval Failures	
	MINERVA	MHN
Compositional	0.68^{***} (0.63, 0.73)	0.817^{***} (0.71, 0.942)
Collocation	4.074^{***} (3.78, 4.39)	1.192^{**} (1.022, 1.39)
Frequency	$1.61\text{e}{-08}^{***}$ (8.68e−09, 2.98e−08)	$1.75\text{e}{-25}^{***}$ (1.22e−26, 2.53e−24)
Constant	$1{,}047.0^{***}$ (1.02e−03, 3.58e−03)	5.900^{***} (1.03e−10, 8.25e−10)
N	70200	23400
Note:		*p<0.1; **p<0.05; ***p<0.01

project inputs into and out of the latent space, respectively. Matching between inputs and memory items is done via query-key attention in the latent space, so a lower value for c forces the MHN to compress more strongly.

The experimental setup is similar to that for MINERVA, with the difference being the learnable nature of the model's memory. As in Section 5.1, for each stimulus item $x \in \mathbb{R}^d$ we construct a number of noisy versions $\{x'_1, \ldots x'_i\}$ of its vector embedding, where i is proportional to the item's corpus frequency. Analogously to the MINERVA memory matrix, our total training set for the MHN comprises $10,000$ embeddings, with more frequent items being more represented. Given one such noisy embedding x', the MHN was trained to output the un-noisy source embedding x. Specifically, it must maximize s = cosine_similarity$(\hat{x}, x)$, where $\hat{x}$ model output. We conjecture that to succeed on the task, the model must optimize its limited memory to map multiple noisy versions of the same item to the item's canonical representation.

As in the MINERVA experiments, we judge whether the model's retrieval was successful based on the cosine similarity s. If s is above threshold K, we say the retrieval is a success. Otherwise, it is a failure. Unlike in MINERVA, there is currently no analogue to RTs in MHN.

We ran a Bayesian hyperparameter sweep with Hyperband early stopping (Falkner et al., 2018) to find configurations in which the pattern of failures across the three conditions which match human trends. The results of one such configuration run over 100 seeds is presented in Figure 8. As in the MINERVA experiments, collocations fail the most, followed by compositional items, with idioms failing the least. All differences between conditions are statistically significant (see Table 6) when analyzed with the same model as the MINERVA failures. This matches the trends for the failures by condition in MINERVA, and the trend of reaction times in humans.

Please note that our parameter sweep was not exhaustive and we cannot make any claims with respect to hyperparameter robustness of the MHN simulations. However, the fact that the MHN can be shown to display the same behaviour as MINERVA in at least one case suggests that the elevated retrieval failure rate of collocations may be a property of associative memory retrieval writ large, rather than a quirk of the MINERVA architecture.

The model hyperparameters are as follows: $F =$

$0.2, K = 0.971, c = 625, n = 2205$. Scaling parameter β was set to $\frac{1}{\sqrt{c}} = 0.0016$. The model was trained for 300 epochs with batch size 16 on one A100 GPU.

E Averaging vs Concatenating sBERT Embeddings

In this ablation, we investigate the impact which concatenating verb and noun embeddings has on our modelling results. Instead of concatenating verb and noun embeddings, we perform mean-pooling across them, the same as we do for sub-word tokens. As shown in Figure 9, the trends exhibited by the model in the $K = 0.99, F = 0.8$ hyperparameter configuration are largely the same as those reported in the main text.

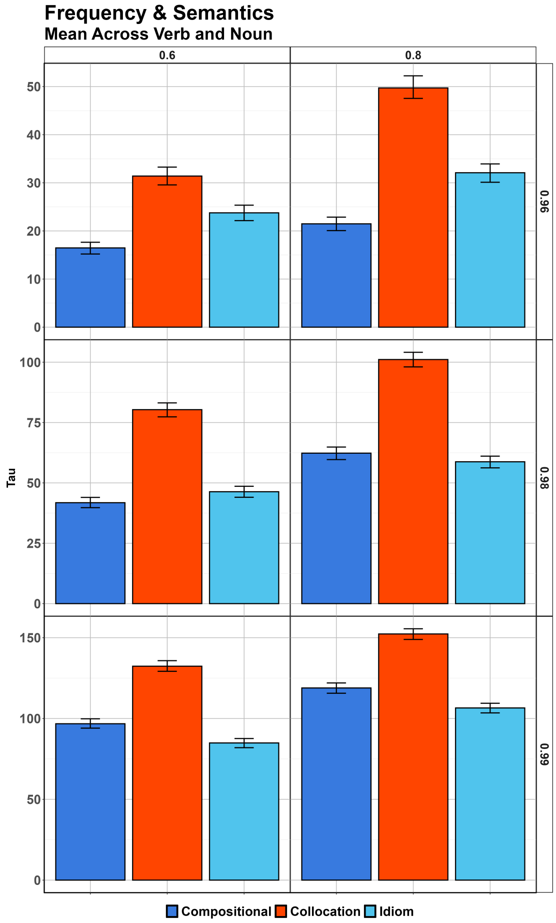


Figure 9: Reduced hyperparameter sweep showing the effects of mean-pooling the verb and noun embeddings before loading them into MINERVA, instead of concatenating them. Note that the hyperparemeter combination reported in the main text is $K = 0.99, F = 0.8$.

F Hyperparameter Sweeps for Simulation Experiments

See next page.

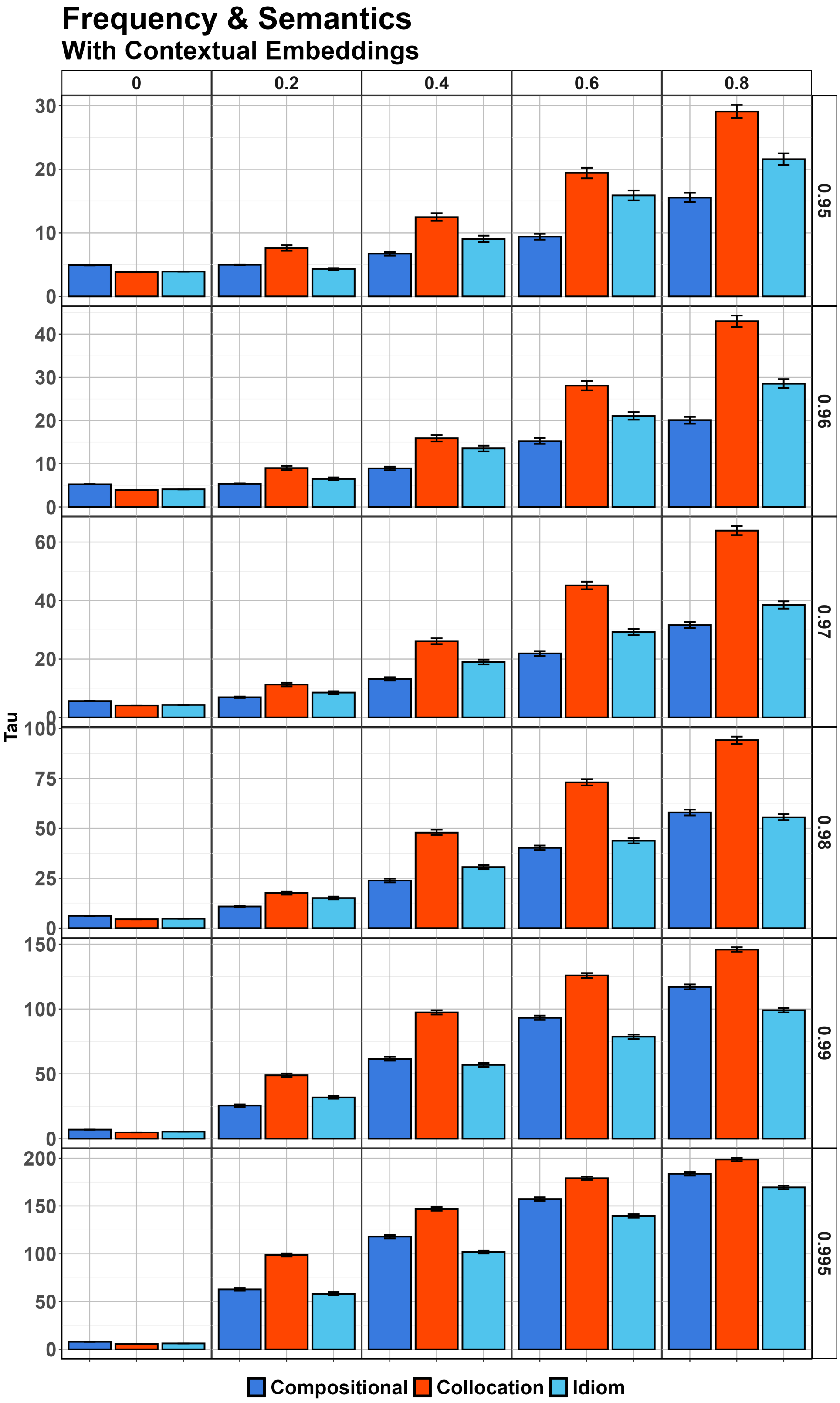


Figure 10: Results of the hyperparameter sweep for all values of activation threshold K and forgetting probability F for our main experiment. Error bars indicate bootstrapped confidence intervals. Note the difference in scales on the y-axis.

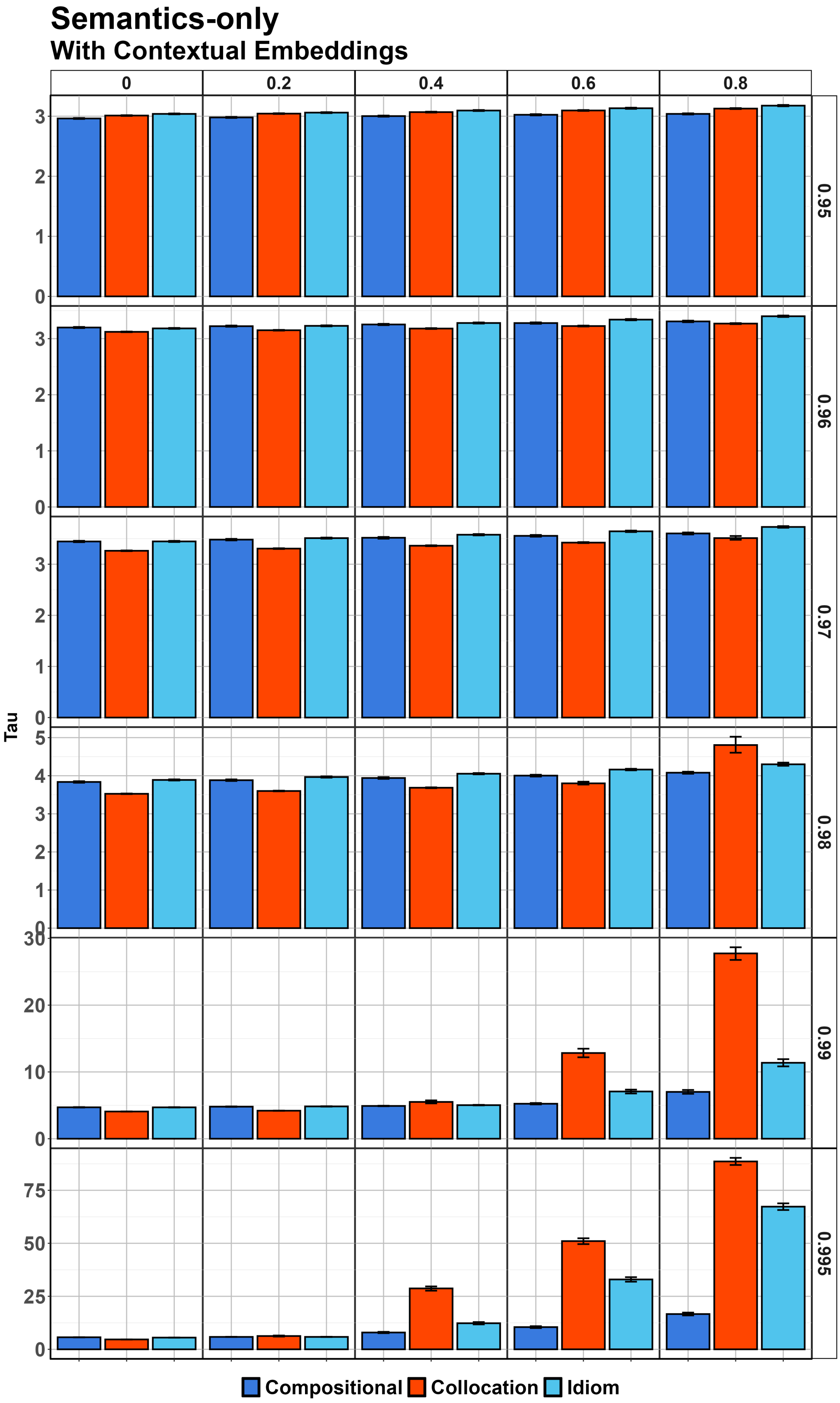


Figure 11: Results of the hyperparameter sweep for all values of activation threshold K and forgetting probability F for Simulation 2: Semantics-only wherein the matrix was loaded with all items having equal frequency. Error bars indicate bootstrapped confidence intervals. Note the difference in scales on the y-axis.

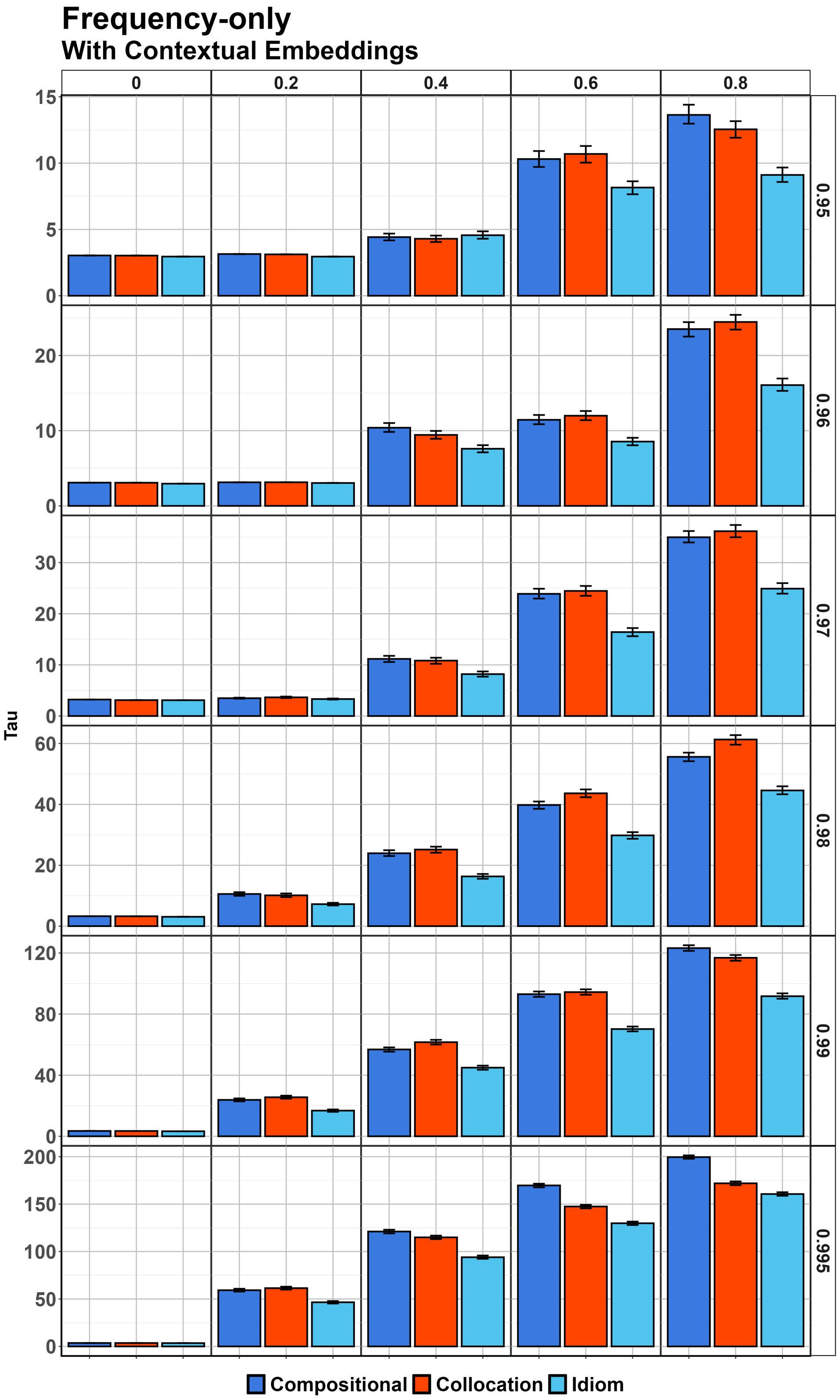


Figure 12: Results of the hyperparameter sweep for all values of activation threshold K and forgetting probability F for Simulation 2: Semantics-only wherein the matrix was loaded with noised embeddings but with the correct frequency. Error bars indicate bootstrapped confidence intervals. Note the difference in scales on the y-axis.

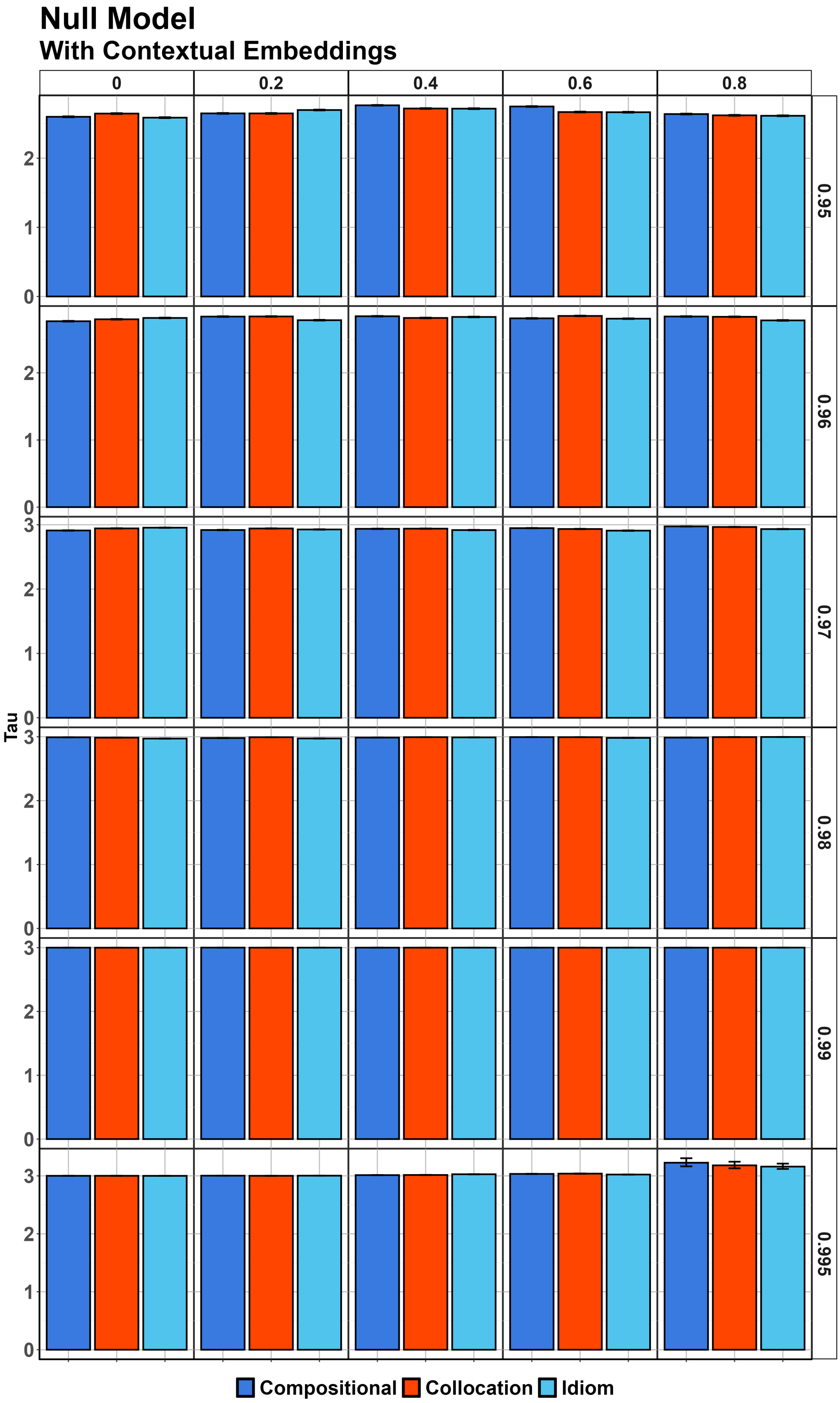


Figure 13: Results of the hyperparameter sweep for all values of activation threshold K and forgetting probability F for the Null Model wherein all the items in the matrix were loaded with noised embeddings and equal frequency. Error bars indicate bootstrapped confidence intervals. Note the difference in scales on the y-axis.

Polarity inversion operators in PLM

Kletz David[1,2], Amsili Pascal[1], Candito Marie[2]
[1]Université Sorbonne Nouvelle & Lattice (CNRS/ENS-PSL/USN),
[2]Université Paris Cité & LLF (CNRS/UPC),
`david.kletz@sorbonne-nouvelle.fr, marie.candito@u-paris.fr, pascal.amsili@ens.fr`

Abstract

From a linguistic perspective, negation is a unique and inherently compositional operator. In this study, we investigate whether the `bert-large-cased` Pretrained Language Model (PLM) properly encodes this compositional aspect of negation when embedding a token that falls within the scope of negation. To explore this, we train two external Multi-Layer Perceptrons to modify contextual embeddings in a controlled manner. The goal is to reverse the polarity information encoded in the embedding while preserving all other token-related information. The first MLP, called the Negator, transforms a negative polarity into a positive one, while the second, the Affirmator, performs the reverse transformation. We then conduct a series of evaluations to assess the effectiveness of these operators. Our results indicate that while the Negator/Affirmator is functional, it only partially simulates the negation operator. Specifically, applying it recursively does not allow us to recover the original polarity, suggesting an incomplete representation of negation within the PLM's embeddings. In addition, a downstream evaluation on the Negated LAMA dataset reveals that the modifications introduced by the Negator/Affirmator lead to a slight improvement in the model's ability to account for negation in its predictions. However, applying the Negator/Affirmator recursively results in degraded representations, further reinforcing the idea that negation is not fully compositional within PLM embeddings.

1 Introduction

In this work, we aim to investigate how well Pretrained Language Models (PLMs) handle compositionality, by focusing on the possibility of defining a "negation operator."

From a logical and linguistic perspective, negation provides a typical example of semantic compositionalism: its effect is systematic and independent of the specific meaning of the clause to which it applies: negation simply reverses the truth value of a statement.

To put it differently, the meaning of a negation word (such as *not*) in a sentence does not depend on the particular verb used in the sentence, nor on the original polarity (i.e., whether the sentence was initially affirmative or negative). Instead, it follows a general rule: it systematically flips the sentence's polarity.

PLMs, however, do not construct the meaning of a sentence by recursively decomposing it into meaningful constituents. Instead, they generate contextual embeddings, so that the representation of a word depends on the surrounding words in the sentence. Given this, our goal is to identify a transformation (a function) that acts as a "negation operator" on embeddings. In other words, we want to find a way to manipulate the numerical representation of a word such that, after applying this transformation, we obtain an embedding that closely resembles what the model would have produced if the same word had occurred in a sentence with the opposite polarity.

For example, given an affirmative sentence like (1-a), we want to define an operation such that, when applied to the embedding that a PLM associates with the main verb *buy* in an affirmative context (noted Vp), it yields an embedding (noted $Vp-$) that is as close as possible to the embedding that the same PLM would assign to the token *buy* in a negative context (1-b) (noted Vn).

(1) a. Sam will buy a new car.
b. Sam will not buy a new car.

In the remainder of this paper, we will refer to a pair of sentences such as (1) as a **minimal pair** (keeping implicit the fact that the difference between the two sentences necessarily involves negation).

Our reasoning is as follows: if a PLM contains

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 312–322
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

a compositional negation operator, then the difference between the embeddings Vp and Vn should be learnable, regardless of the lexical properties of the verb and the polarity of its context.

We therefore try in this work to learn a polarity inversion function that can operate on verb embeddings and that is sufficiently general to work on verbs whose lemmas were not seen at training, and on verbs occurring in affirmative or negative contexts.

We show that it is indeed possible to learn an operator (a *Negator*) that produces from Vp embeddings new embeddings $Vp-$ close enough to their corresponding Vn, and such that it generalizes correctly to lemmas not seen during training. This suggests that it is possible to locate in the embeddings distinct encodings for lexical representation and polarity. It is also possible to learn an *Affirmator* that produces an appropriate embedding $Vn+$ even for lemmas not seen at training (section 3). However, it turns out that these two operators cannot be used one for another: a Negator (trained only with embeddings occurring in affirmative contexts) does not succeed at "inverting" the embedding of a verb occurring in a positive context (similarly for the Affirmator). This indicates that they do not generalize to a true polarity inversion operation independent of the direction of the inversion, which is contrary to the classical logical and linguistic interpretation of negation (they are not involutions, i.e., they are not their own inverse: $\text{Negator}(\text{Negator}(x)) \neq x$ and $\text{Affirmator}(\text{Affirmator}(x)) \neq x$). On the contrary, we show (in section 4) that they are indeed reciprocal functions of each other: $\text{Affirmator}(\text{Negator}(x)) \approx x$.

It is worth studying further the properties of these two operators, even though they don't behave exactly as is expected from a logical perspective. Since they are not involutions, we study in section 5 the effect of their multiple application, and in particular a possible (non-linguistic) effect of "polarity reinforcement", usable to improve the processing of negation by a PLM.

Finally, in the last section of this paper (§ 6), we study the impact of the integration of our Negator into the processing pipeline of the negated LAMA task.

Our experiments show that the integration of the Negator leads to a slight improvement in the model's predictions. This suggests that modifying embeddings with the Negator allows the language modeling head to differentiate a little bit better between positive and negative embeddings, enabling it to adjust its predictions more accurately. However, the operator is applied several times ("recursively"), the predictions of the model become very unnatural, which is another way to show that our operators do not restrain their action to the strict encoding of polarity in the embeddings.

2 Related works

Negation in PLMs The presence of contextual polarity information in contextual embeddings generated by PLMs has been investigated by Celikkanat et al. (2020), who specifically looked for "traces" of negation. By analyzing contextual embeddings produced by a PLM, they showed that it is possible to predict whether the main verb of a sentence is negated or not. Building on this, Kletz et al. (2023b) showed that the encoding of such information is itself dependent on the syntactic position of the token used as input, in particular whether it falls or not within the scope of a negation.

Beyond encoding, the ability of models to consider negation in their predictions within a Masked Language Model (MLM) setup has also been explored. Kassner and Schütze (2020) and Ettinger (2020) examined how negating the main verb of a clause affects its truth value. Specifically, they investigated the capability of masked language models to adjust their predictions for a masked position when confronted with factual world knowledge (Li et al., 2016).

Kassner and Schütze (2020) constructed the negated LAMA dataset by negating sentences from the original LAMA dataset (Petroni et al., 2019). They then analyzed the behavior of masked PLMs when processing negated cloze-style sentences. Their findings revealed a similarity between model predictions in affirmative and negative contexts, leading them to conclude that "PLMs do not distinguish positive and negative sentences."

Similarly, Ettinger (2020) used sentences originally designed by Fischler et al. (1983) to observe how human expectations about sentence continuation shift when negation is introduced. The lack of corresponding adjustments in PLM predictions led her to a similar conclusion that PLMs exhibit insensitivity to negation.

However, other approaches (Gubelmann and Handschuh (2022) and Kletz et al. (2023a)), decided to avoid factual statements. They constructed examples with two sentences, where a particular word was either highly plausible (in positive cases) or semantically ruled out (in negative cases) at a masked position in the second sentence, given the context provided by the first. The fact that larger PLMs adjusted their predictions based on sentence polarity led these authors to a different conclusion that certain PLMs are indeed capable of considering negation.

Hosseini et al. (2021) proposed improving the predictions of `bert-base-cased` in negative contexts by fine-tuning it into a new model called `BERTNOT`. They created a dataset of 40,000 concatenated sentence pairs, each consisting of a premise (sourced from Wikipedia) and a hypothesis where a noun dependent on the main verb was selected and masked. Half of these pairs retained the hypothesis unchanged, while the other half contained a negated version of the premise, created by negating the main verb. The fine-tuning process involved two key objectives: one function aimed to prevent the model from predicting the selected token in sequences where the second sentence was negated, while another function ensured that the masked token distribution remained unchanged for the other 20,000 sequences. `BERTNOT` was subsequently evaluated using NLI datasets and Negated LAMA. The evaluation results indicated that `BERTNOT` made far fewer factually incorrect predictions than `bert-base-cased`.

Compositionality in PLMs In general, the evaluation of compositionality in language models focuses on compositional behaviors (McCurdy et al., 2024) and the ability of PLMs to generalize. Research in this area typically tests models through external tasks, where successful resolution implies the ability to generalize compositionally—either lexically (as in COGS (Kim and Linzen, 2020) and SCAN (Lake and Baroni, 2018)) or structurally (as in SLOG (Li et al., 2023)).

Kim and Linzen (2020) reported disappointing performance from tested models on generalization sets, concluding that these models struggle with both lexical and structural compositional generalization. However, more recent studies have shown that using models with pretraining strategies focused on meta-learning (Conklin et al., 2021) or employing newer transformer-based architectures (Sun et al., 2023; Tay et al., 2021; Raffel et al., 2020) significantly improves compositional generalization, surpassing the capabilities of smaller transformer models.

3 Inverting polarity : training a Negator and Affirmator

In this section, we learn mathematical functions (MLPs) to modify contextualized embeddings so as to mimic the difference between embeddings originating from the two clauses of a minimal pair. We will talk of the "**polarity of an embedding**" for short. Hence for instance, in *I wish war didn't exist*, the polarity of the embedding of *exist* or *war* is negative, whereas the polarity of the embedding of *I* or *wish* is positive.

More precisely, we consider embeddings of target verbs. The basic principle for our Negator function (resp. Affirmator) is to take as input the contextualized embedding of an affirmative verb, noted Vp (Vn for a negative verb) and output the corresponding embedding as if the verb was in a negative (resp. affirmative) context ($Vp-$, resp. $Vn+$)[1].

The Negator (resp. Affirmator) consists in a MLP trained on (Vp,Vn) pairs (resp. (Vn,Vp) pairs). The evaluation consists in comparing $Vp-$ to the original Vn, and $Vn+$ to the original Vp. For short we will talk of the **original** embeddings (Vp or Vn) and their corresponding **reversed** embeddings ($Vn+$ and $Vp-$).

Data We took as a starting point a set of 20,000 minimal pairs provided by Hosseini et al. (2021), formed with 20,000 sentences from Wikipedia, where the direct object of a target verb has been masked, along with a version where the target verb is negated.

We have deduplicated the 20,000 pairs, and removed pairs containing either zero or more than one masked position (resulting from errors in the masking process), and those where the target verb is tokenized into several subwords when encoded by the PLM we test (namely `bert-large-cased`). This brought the dataset down to ~15,000 pairs. For our purposes, we restored the masked object, and identified the target verb[2], left unmasked.

We then split this data into 11,708 training pairs and 2,927 test pairs, each set corresponding to disjoint sets of target verb lemmas.

---

[1] We did try to obtain the Affirmator by defining the reciprocal function of the Negator. However, the learned parameter square matrices turned out to be non-invertible (details in Appendix A).

[2] To this end, we parsed the sentences using stanza (Qi et al., 2020), and took the closest verbal ancestor node of the direct object, in the dependency tree.

Architecture and training The MLPs for Negator and Affirmator have the same architecture: 4 hidden layers of same size as input contextualized embeddings, namely 1024, with *LeakyRELU* activation (with a negative slope of 10^{-2}) for the first 3 layers and ELU for the last layer ($\alpha = 1$).

We train the Negator on the training (Vp,Vn) pairs using the MSE loss, and simply switched to (Vn,Vp) pairs to train the Affirmator[3].

Evaluation metrics We measure the quality of the Negator (resp. Affirmator) using two direct metrics and two indirect metrics, each comparing the original embeddings to their corresponding reversed embeddings (hence comparing $Vp-$ to Vn, and $Vn+$ to the original Vp). The two direct metrics are simply **cosine** similarity and mean square error (**MSE**). The two indirect metrics compare the probability distributions output by the language modeling head of the PLM, when fed with an original embedding vs. when fed with the corresponding reversed one. More precisely, if we note Pn the distribution obtained with the original Vn embedding, and $Pp-$ that obtained from $Vp-$, we use the **KL-divergence** $D_{KL}(Pp - || Pn)$ averaged over each evaluated pair, and the proportion of evaluated pairs for which the top-1 prediction is the same in Pn and $Pp-$ (and accordingly for the Affirmator case), hereafter **same-top-1**. Among these four metrics, higher cosines and same-top-1 will mean better quality, while it is the opposite for MSE and KL divergences.

Moreover, while same-top-1 can be interpreted in isolation, for the other three metrics, we need reference values for comparison. To this effect, we compute cosine, MSE and KL-divergence for sets of various pairs of embeddings, obtained by encoding sentence pairs from our dataset, with `bert-large-cased`. These pairs of embeddings either concern the same token from a pair of sentences varying in polarity (Vp, Vn), or different tokens from the same sentence, or two tokens in two different sentences but corresponding to the same word form, and finally two embeddings from two random tokens taken from two random sentences from the affirmative sentences of our dataset.

The reference values are provided in Table 1. The first row concerns Vp and Vn pairs, and provide the reference values for embeddings differing only in polarity, and we will refer to these values to evaluate our Negator and Affirmator. As all the metrics show, all other tested pairs of embeddings show a much higher divergence. Note that two distinct tokens of the same sentence (second row) have much closer embeddings than the embeddings of the same word in two different sentences (third row).

sent.	pol.	token	MSE	Cosine	KL-div
=	≠	=	0.02	0.96	0.05
=	=	≠	0.30	0.50	8.17
≠	?	=	0.46	0.23	9.21
≠	+	?	0.57	0.14	20.74

Table 1: Calibration of metrics: reference values for MSE, cosine and KL-divergence metrics, when using various kinds of pairs of embeddings. The pairs are either embeddings from the same sentence (when ignoring polarity) (first column), from sentences with equal, different, irrelevant (?) or positive polarity (pol. column), and from the same word or not (token column). The first row compares (Vp, Vn).

3.1 Results

	MSE	cosine	KL-div	same-top-1
Vp vs. $Vn+$	0.12	0.80	0.66	83.9
Vn vs. $Vp-$	0.13	0.79	0.80	81.5

Table 2: Evaluation of the Affirmator (first row) and Negator (second row) on the test set: comparison metrics for pairs of original vs reversed embeddings.

We provide the evaluation results of the Negator and Affirmator, computed on the test set, in Table 2. The same-top-1 results are above 80%. Interpreting the three other metrics requires to compare them to the reference values in Table 1. The same trend is observed for MSE, cosine and KL-divergence: although the results comparing original and reversed embeddings are less good than when comparing the original (Vp, Vn) pair (first row of Table 1), they are a lot better than when comparing other kinds of pairs of embeddings (last 3 rows of Table 1). These observations tend to show that our trained polarity inversion operations lead to embeddings that are (i) close to the corresponding original embedding ($Vn+$ close to Vp, $Vp-$ close to Vn); and (ii) close enough to appropriately feed the original language modeling head, resulting in a probability distribution over the vocabulary that is close to the original one.

---

[3] We use the Adam optimizer. We tuned the learning rate (10^3) and the number of epochs (4) using cross-validation on the training set.

Generalization across verbs Since the training and test set contain disjoint sets of verb lemmas, the previous observations tend to show a good generalization to verbs unseen during training. To further check this generalization, we would also like to verify that the averaging applied in the metrics does not hide a disparity in performance, and in particular that errors are not concentrated on a specific set of verbs. To this end we calculate the same-top-1 proportion per lemma, and count the number of verbal lemmas for which the proportion is very low (top-1 accuracy below 20%), indicating a total failure of inverting the polarity of embeddings for these verbs. We restrict ourselves to lemmas with at least 5 occurrences in the test set.

The results are provided in table 3. We observe only 3 lemmas with a same-top-1 proportion of less than 20% for Affirmator, and none for Negator. We can thus conclude that there are practically no lemmas for which polarity re-encoding systematically fails.

This further confirms that it is indeed possible to learn a polarity transformation of a verbal embedding, *independently of the corresponding verb*, a first step towards a compositional polarity inversion operator (cf. section 1).

Model	# tested lemmas	Cases w/ rate <20%
Affirmator	277	3
Negator	271	0

Table 3: Total number of unique lemmas tested, and number with same-top-1 proportion below 20%.

Generalization across polarities The second necessary condition was that the learnt polarity inversion operations should generalize across polarity. In our case, it means firstly that the Negator and Affirmator should actually correspond to the same (or a close) mathematical function, performing a polarity inversion independently of the polarity of its input. Secondly, given the logical interpretation of negation, both the Negator and Affirmator should be an involution, namely their own reciprocal function, hence Negator(Negator(V)) should be close to V. We report on this investigation in section 5.

4 Evaluation via a polarity probe

In order to further assess the effectiveness of the Negator/Affirmator, we employ a MLP probe trained to predict the polarity of verbal embeddings. Importantly, the probe is trained exclusively on Vp and Vn, without exposure to reversed embeddings ($Vp-$ and $Vn+$).

Training of the probe The trained probe is an MLP consisting of a hidden layer of the same size as the input (1024), with sigmoid activation. It is trained for 5 epochs with a learning rate of 0.3.

As training data we reuse the dataset used to train our Affirmators/Negators: we keep at random one sentence from each pair, which yields $\simeq$14,000 sentences balanced with respect of their polarity. We split them into 11708/2927 for training and testing, keeping a balanced polarity in each set.

Evaluation on original embeddings The accuracy of the probe on the test set is provided in the "Original" columns of Table 4.

Embedding								
Original			Reversed			Reinforced		
inp.	exp.	acc.	inp.	exp.	acc.	inp.	exp.	acc.
Vn	n	95.9	$Vn+$	p	99.9	$Vn-$	n	99.9
Vp	p	96.6	$Vp-$	n	99.9	$Vp+$	p	99.8

Table 4: Accuracies of the polarity-predicting probe, on the verbal embeddings of the test set, using either the **Original** embeddings (Vn or Vp), the **Reversed** ones ($Vn+$ or $Vp-$), and the **Reinforced** ones ($Vp+$ or $Vn-$). Columns inp.: type of input embedding; Columns exp.: expected polarity label; Columns acc.: probe accuracy

We observe that the probe has a very high accuracy to predict the polarity of original embeddings (first three columns, above 95%), although not perfect.

Evaluation on reversed embeddings We now check how the probe behave when fed with reversed embeddings. Results of applying the probe on these are provided in the "Reversed" columns of Table 4. We observe almost perfect accuracy for both the Negator and the Affirmator. This constitutes a further evaluation of the quality of the Negator/Affirmator, since they allow to better predict the polarity of an embedding.

5 Polarity inversion or reinforcement?

In this section, we examine the effects of applying the Negator to a verbal embedding originating from a negated verb (which, following our notation, results in $Vn-$). Similarly, we analyze $Vp+$ cases, where the Affirmator is applied to a verbal embedding originally not negated.

In this case, if the Negator/Affirmator is the same transformation, applying a real inversion of polarity independently of the polarity of their argument, then $Vn-$ should be close to Vp and have positive polarity. $Vp+$ should be close to Vn and have negative polarity.

If on the contrary the Negator/Affirmator are distinct, each "moving" the polarity of their argument on a "polarity scale" in opposite directions, then we anticipate a reinforcement of the encoding of the polarity (and we will use the term **reinforced** embedding for $Vn-$ and $Vp+$ types of embeddings).

The results are provided in the "Reinforced" columns of Table 4, the accuracy being calculated when expecting a reinforcement rather than an inversion. We can see that the accuracies are almost perfect for both the Negator and Affirmator. So as the name "reinforced" hinted, we observe a reinforcement of the polarity instead of an inversion independent of input polarity.

Note though that the accuracy on the Reinforced cases is similar to that of the Reversed cases. So while it shows that the Negator/Affirmator does strengthen the polarity encoding, it is surprising it cannot surpass the reversed cases.

Error analysis We further study the counts of well-classified/misclassified cases, and whether the polarity inversion or reinforcement introduces new errors. Table 4 provides the exact counts of correct/incorrect polarity prediction by the probe, when fed by original, reversed and reinforced embeddings. After polarity inversion, we count 179 corrected errors and only 3 introduced errors (resp. 177 and 3 after reinforcement).

The very low number of new errors introduced by the Negator/Affirmator further assesses their ability to inverse/reinforce polarity encoding in embeddings, without altering it.

Orig.	Count	After modif.	Rev.	Reinf.
✓	re			
		↪✓	4589	4589
		↪ x	3	3
x	180			
		↪ ✓	179	177
		↪ x	1	3

Table 5: Counts of correct/incorrect labels after applying the polarity probe on original, reversed and reinforced embeddings.

6 Using the Negator to enhance `bert-large-cased`'s predictions

We now propose to use the Negator for a different objective: rather than studying the possibility of learning a compositional negation operator, we investigate whether the negator can help to improve the negation "understanding" of a `bert-large-cased` model, in a downstream task. We choose the negatedLAMA task, which Kassner and Schütze (2020) designed to assess the ability of `bert` to adapt its language modeling predictions to the presence of negation (cf. section 2).

6.1 The negated LAMA data and task

The negated LAMA dataset (Kassner and Schütze, 2020) is a negated version of LAMA (Petroni et al., 2019), itself developed to assess the factual knowledge stored in PLMs. It consists of factual statements derived from various encyclopedic sources[4], in which a token is masked (e.g. *dog* (2)), hereafter the **original affirmative token**).

The negated LAMA dataset is constructed by associating each affirmative factual statement (p) from LAMA with their negated counterpart (n).

(2) **Op** (Original): A beagle is a type of **dog**.
Mp A beagle is a type of [MASK].
Mn A beagle is not a type of [MASK].

The original affirmative token should be the top-1 prediction for the affirmative sentences, but this token becomes factually wrong in the negative counterparts, hence these pairs provide a way to assess a model's sensitivity to polarity changes.

Since the negated LAMA data is not explicitly available, we reconstructed the dataset, and the details of this process can be found in Appendix B. Consequently, although we made every effort to ensure accuracy, the version of the dataset we use differs from the ones employed by Kassner and Schütze (2020) and Hosseini et al. (2021).

To measure performances of the model, we use the **stability rate** of Kassner and Schütze (2020), which measures the percentage of identical top-1 predictions for (Mp,Mn) pairs. The lower the stability rate is, the more the model is sensitive to negation. Note that this measure does not take into consideration the cases where the top-1 prediction for Mp is not identical to the original affirmative

---

[4] Google-RE (Google, 2013), T-REX (Elsahar et al., 2018), ConceptNet (Speer and Havasi, 2012), and SQuAD (Rajpurkar et al., 2016).

token: if, for example in (2), a model has *mammal* as its top-1 prediction, what matters for the stability rate is whether this token is still the top-1 prediction in the negative case Mn. Hence, we also introduce a metric to quantify the average rate of factually incorrect predictions, referred to as the ***fipa rate***[5]. The *fipa* rate measures the proportion of top-1 predictions for negated sentences that still match the original affirmative token. A lower *fipa* rate suggests that the model is better at generating factually correct predictions under negation[6].

6.2 Setup

We propose to integrate the Negator into the language modeling prediction pipeline of a `bert-large-cased` model, by applying the Negator to a token's representation at last layer, before feeding the language modeling head (see Figure 1).

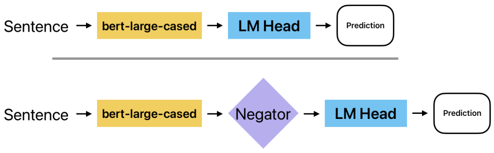


Figure 1: Inserting the Negator in the language modeling task.

Training We retrain a Negator using the encodings of the original masked tokens of Hosseini et al. (2021)'s dataset, not being unmasked (using our notation scheme, we consider Mp, $Mp-$, Mn, $Mn-$, under the same conditions as described in Section 3).

Application to negated LAMA The use of the Negator allows us to make two comparisons: between original, reversed and reinforced embeddings (Mn, Mp-, Mn-).

Furthermore, inspired by Ravfogel et al. (2021), we apply the Negator recursively multiple times, progressively even more reinforcing the encoding of negation polarity (e.g., $Mp \rightarrow Mp- \rightarrow Mp-- \rightarrow Mp---$ etc.). We will call these **super-reinforced embeddings**, and note $Mnk-$ the result of applying k times the Negator to Mn.

6.3 Quantitative analysis

The results for the *fipa* rate and stability rate are presented in Table 6.

The prediction shifts of the PLM are highly dependent on the dataset subset used, with no subset enabling the PLM to achieve a stability rate below 30%.

The insertion of the Negator into the processing pipeline under a polarity inversion configuration (line 2) alters the model's predictions. However, the stability and *fipa* rates do not show any improvement—often remaining similar or even worse—compared to directly negating the input sentence. The insertion of the Negator into the processing pipeline under a polarity reinforcement configuration (line 3) is the first combination to yield an improvement, reducing *fipa* rate by 5% to 20% and increasing the stability rate from 14% to 26%.

The use of super-reinforced embeddings leads to the most significant improvements in evaluation quality, both in polarity reinforcement and inversion configurations. The *fipa* rate decreases, ranging from 65% to 79%, and the stability rate improves between 71% and 84%.

Compared to other models, the model incorporating super-reinforced embeddings surpasses the performance of bert-large-cased, as tested by Kassner and Schütze (2020). Additionally, it outperforms BERTNOT (Hosseini et al., 2021), [7] achieving lower *fipa* rates than those reported by Hosseini et al. (2021). Furthermore, a comparison between line 1 and line 4 reveals that the reduction in fipa rate is even more significant than the improvement achieved by BERTNOT.

6.4 Qualitative analysis

To ensure that the representations are not degraded by the application of the Negator and that only the encoding of polarity is affected, we now conduct a complementary qualitative analysis.

We analyze the top-1 predictions of our architecture. For comparison, we revisit the four examples highlighted in Hosseini et al. (2021) and the eight examples from Kassner and Schütze (2020). The results are presented in Table 7.

Semantic and syntactic constraints are preserved in the $Mn-$ configuration.[8] However, the model's

---

[5] This metric may correspond to the average top-1 error rate used by Hosseini et al. (2021). However, since they do not explicitly define it, we cannot confirm this equivalence.

[6] Note each above cited work use only one of these two metrics, which clearly gives an incomplete evaluation.

[7] For this comparison, we refer to Table 12 in the Appendix of their paper, as the results presented in the main text—while higher—were obtained using a BERT-base-cased model.

[8] For instance, in the sentence "Charles Nodier did not die

	***fipa* rate**				**stability rate**			
subset	SQUAD	conceptnet	Google-re	T-rex	SQUAD	conceptnet	Google-re	T-rex
Mn	11.2	2.7	22.2	57.7	43.4	31.6	60.3	90.0
Mp-	15.5	3.3	22.5	58.3	59.5	59.0	61.1	84.8
Mn-	8.9	2.4	20.2	54.5	32.6	23.5	44.1	77.4
Mn5-	**2.3**	**0.8**	**6.4**	**19.8**	**9.5**	**7.3**	**11.6**	**25.3**
Mp5-	5.3	1.4	11.9	23.1	18.4	14.6	25.3	31.4

Table 6: Percentage of cases where the top-1 prediction when feeding the LM head with embedding in column 1 is (left) identical to the expected factual answer for the Vp case, and (right) identical to the top-1 prediction for the Vp case; each broken down for each LAMA subset.

Paper	Sentence	**Representation received by the LM head**		
		Mn	Mn-	Mn5-
H	iOS is not developed by [MASK].	Apple (0.22)	Apple (0.19)	it (0.05)
H	The majority of the amazon forest is not in [MASK].	cultivation (0.43)	cultivation (0.13)	forest (0.04)
H	Charles Nodier did not die in [MASK].	battle (0.29)	battle (0.14)	prison (0.13)
H	Mac OS is not developed by [MASK].	Apple (0.73)	Apple (0.64)	Apple (0.19)
K&S	Marcel Oopa did not die in the city of [MASK].	Paris (0.09)	Paris (0.08)	residence (0.04)
K&S	Anatoly Alexine was not born in the city of [MASK].	Moscow (0.31)	Moscow (0.28)	town (0.05)
K&S	Platonism is not named after [MASK].	Plato (0.78)	Plato (0.35)	himself (0.48)
K&S	Lexus is not owned by [MASK].	Toyota (0.18)	Google (0.07)	it (0.03)
K&S	Birds cannot [MASK].	fly (0.76)	fly (0.33)	property (0.01)
K&S	A beagle is not a type of [MASK].	dog (0.83)	dog (0.72)	person (0.53)
K&S	Quran is not a [MASK] text.	religious (0.32)	religious (0.23)	valid (0.13)
K&S	Isaac's chains are not made out of [MASK].	iron (0.22)	iron(0.16)	stone (0.08)

Table 7: Qualitative analysis of predictions on embeddings modified by Negator. Column 'Paper': 'H' refers to sentences from Hosseini et al. (2021) 'K&S' refers to sentences from Kassner and Schütze (2020). Each cell indicates the prediction. The associated probability is given in parentheses.

predictions frequently remain unchanged from the original, which are often factually incorrect.

Incorporating super-reinforced embeddings does lead to modifications in predictions. However, with the super-reinforced embeddings, the generated sentences often appear unnatural, ultimately compromising the quality of the predictions.

These observations suggest that this method cannot serve to enhance negation interpretation of `bert-large-cased`.

7 Conclusions

In this paper, we explored the compositionality of negation within PLMs by investigating whether a transformation, which we call the "Negator/Affirmator," could reverse the polarity of a verb's embedding. Our results show that it is possible to learn such a function and that it can generalize to unseen lemmas. However, a complementary study reveals that the simple application of the Negator is not sufficient to significantly improve the predictions of `bert-large-cased` in the presence of negation, while multiple applications of the Negator improve the treatment of negation at the expense of a degradation of the embeddings. Even though a negation operation seems therefore learnable, its use for improving the predictions of a PLM still remains problematic.

Looking ahead, it would be interesting to extend this work by isolating operations that encode other compositional operators. This could help determine whether the handling of negation by PLMs is specific or if it is part of a broader pattern in the treatment of compositional operations.

References

Hande Celikkanat, Sami Virpioja, Jörg Tiedemann, and Marianna Apidianaki. 2020. Controlling the Imprint of Passivization and Negation in Contextualized Representations. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 136–148, Online. Association for Computational Linguistics.

Henry Conklin, Bailin Wang, Kenny Smith, and Ivan Titov. 2021. Meta-learning to compositionally generalize. In *Proceedings of the 59th Annual Meet-*

---

in [MASK].", the masked position is syntactically constrained to be filled by a noun, noun phrase, or temporal expression, while semantically, it must refer to a place or time of death.

ing of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 3322–3335, Online. Association for Computational Linguistics.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. 2018. T-REx: A large scale alignment of natural language with knowledge base triples. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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.

Ira Fischler, Paul A Bloom, Donald G Childers, Salim E Roucos, and Nathan W Perry Jr. 1983. Brain potentials related to stages of sentence verification. *Psychophysiology*, 20(4):400–409.

Google. 2013. Google relation extraction corpus. `https://code.google.com/archive/p/relation-extraction-corpus/`. Corpus of judged relation triples.

Reto Gubelmann and Siegfried Handschuh. 2022. Context matters: A pragmatic study of PLMs' negation understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4602–4621, Dublin, Ireland. Association for Computational Linguistics.

Arian Hosseini, Siva Reddy, Dzmitry Bahdanau, R Devon Hjelm, Alessandro Sordoni, and Aaron Courville. 2021. Understanding by understanding not: Modeling negation in language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1301–1312, Online. Association for Computational Linguistics.

Nora Kassner and Hinrich Schütze. 2020. Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

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

David Kletz, Pascal Amsili, and Marie Candito. 2023a. The self-contained negation test set. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 212–221, Singapore. Association for Computational Linguistics.

David Kletz, Marie Candito, and Pascal Amsili. 2023b. Probing structural constraints of negation in pretrained language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 541–554, Tórshavn, Faroe Islands. University of Tartu Library.

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

Bingzhi Li, Lucia Donatelli, Alexander Koller, Tal Linzen, Yuekun Yao, and Najoung Kim. 2023. SLOG: A structural generalization benchmark for semantic parsing. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3213–3232, Singapore. Association for Computational Linguistics.

Xiang Li, Aynaz Taheri, Lifu Tu, and Kevin Gimpel. 2016. Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1445–1455, Berlin, Germany. Association for Computational Linguistics.

Kate McCurdy, Paul Soulos, Paul Smolensky, Roland Fernandez, and Jianfeng Gao. 2024. Toward compositional behavior in neural models: A survey of current views. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9323–9339, Miami, Florida, USA. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Robyn Speer and Catherine Havasi. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3679–3686, Istanbul, Turkey. European Language Resources Association (ELRA).

Kaiser Sun, Adina Williams, and Dieuwke Hupkes. 2023. A replication study of compositional generalization works on semantic parsing. In *ML Reproducibility Challenge 2022*.

Yi Tay, Mostafa Dehghani, Jai Prakash Gupta, Vamsi Aribandi, Dara Bahri, Zhen Qin, and Donald Metzler. 2021. Are pretrained convolutions better than pretrained transformers? 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 4349–4359, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Limitations

Our study focuses solely on the `bert-large-cased` model, meaning our findings cannot be directly generalized to other Pretrained Language Models (PLMs). Future work should extend this analysis to a broader range of architectures to assess whether our observations hold across different models.

Additionally, both our dataset and model were in English, limiting our conclusions to this linguistic context. Since negation varies across languages in both syntax and semantics, evaluating models trained on other languages would be necessary to determine the broader applicability of our approach.

A Attemps to define the reciprocal of the Negator

Instead of learning both a Negator and Affirmator, we also tried to learn a Negator, and then define its reciprocal, to serve as Affirmator. This supposes to define the reciprocal of activation functions and of linear combinations.

To this end, we used bijective activation functions, whose reciprocal functions are:

$$LeakyRELU^{-1}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \frac{x}{\alpha}, & \text{otherwise} \end{cases}$$

$$ELU^{-1}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ log(\frac{x}{\alpha+1}), & \text{otherwise} \end{cases}$$

The reciprocal of the linear combination with parameters W and b, requires W to be invertible (which is why we chose square parameter matrices), and is written as:

$$x = W^{-1}(y - b) \tag{1}$$

Unfortunately, we empirically observed across various runs that the resulting Negator contained at least one non-invertible matrix (namely with a rank lower than the shape of the matrix).

We also tried to use Moore-Penrose pseudoinverse parameter matrices[9]. In such a case, the definition of the reciprocal is as below (with W^{PI} the pseudoinverse of W):

$$x = W^{PI}(y - b) \tag{2}$$

So for a given linear layer, if $y = Wx + b$ then we compute a x' such that $x' = W^{PI}(y - b) \simeq x$, namely there exists a matrix M such that $x' = x + M$.

---

[9] Using pytorch, `https://pytorch.org/docs/stable/generated/torch.linalg.pinv.html`, Paszke et al. (2019))

This empirically failed in the sense that when applying the reciprocal functions in sequence, we noted the M matrices kept growing exponentially. The approximation made using pseudoinverses led to growing errors.

We conclude to the impossibility of inversing the Negator to obtain an Affirmator.

B Reconstruction and Preprocessing of Negated LAMA

The original LAMA dataset is available both in the repository of Petroni et al. (2019) [10] and on the Hugging Face (Wolf et al., 2020) platform [11]. However, with the exception of the SQUAD subset, the number of entries differs between these two sources for every subset. A comparison of the dataset sizes from these two sources can be found in Table 8, under the columns "LAMA" (repository from the original paper) and "LAMA HF" (Hugging Face platform).

The inputs of negated LAMA are either explicitly provided, or through the introduction of a negation pattern.

Upon examining the data, we found that not all entries could be used to reconstruct the negated LAMA dataset. We applied filtering criteria to exclude entries with the following issues:

- Presence of multiple masked tokens
- Absence of a corresponding negated sentence
- Lack of alignment with a recognizable negation pattern

These inconsistencies accounted for nearly two-thirds of the data in the Google-RE and T-REx subsets, and we were unable to fully resolve all of them.

The final sizes of the subsets used to evaluate our models are listed in the "Retained Examples" column of Table 8.

Consequently, we use a version of negated LAMA that is different from the one used by Kassner and Schütze (2020) and Hosseini et al. (2021).

Dataset	**LAMA Subsets**		
	LAMA	LAMA HF	#retained examples
SQUAD	305	305	301
conceptnet	2996	29774	8296
Google-re	5527	6106	2926
T-rex	34039	1304391	16991

Table 8: Subset sizes of LAMA from different sources. **Col. “LAMA”**: Number of entries in Petroni et al. (2019) repository. **Col. “LAMA HF”**: Number of entries in the Hugging Face version. **Col. “Retained Examples”**: Final number of entries used in our negated LAMA version.

---

[10] `https://dl.fbaipublicfiles.com/LAMA/negated_data.tar.gz`

[11] `https://huggingface.co/datasets/facebook/lama`

Dynamic Epistemic Friction in Dialogue

Timothy Obiso[1] **Kenneth Lai**[1] **Abhijnan Nath**[2]
Nikhil Krishnaswamy[2] **James Pustejovsky**[1]
[1]Brandeis University, Waltham, MA USA
[2]Colorado State University, Fort Collins, CO USA
{timothyobiso, klai12, jamesp}@brandeis.edu
{abhijnan.nath, nkrishna}@colostate.edu

Abstract

Recent developments in aligning Large Language Models (LLMs) with human preferences have significantly enhanced their utility in human-AI collaborative scenarios. However, such approaches often neglect the critical role of "epistemic friction," or the inherent resistance encountered when updating beliefs in response to new, conflicting, or ambiguous information. In this paper, we define *dynamic epistemic friction* as the resistance to epistemic integration, characterized by the misalignment between an agent's current belief state and new propositions supported by external evidence. We position this within the framework of Dynamic Epistemic Logic (Van Benthem and Pacuit, 2011), where friction emerges as nontrivial belief-revision during the interaction. We then present analyses from a situated collaborative task that demonstrate how this model of epistemic friction can effectively predict belief updates in dialogues, and we subsequently discuss how the model of belief alignment as a measure of epistemic resistance or friction can naturally be made more sophisticated to accommodate the complexities of real-world dialogue scenarios.

1 Introduction

In cooperative, well-grounded conversations, the exchange of information often appears straightforward. Participants typically assume that updates to one another's beliefs will be smooth and consistent with mutual common ground. A listener hears a speaker's assertion and, assuming trust and shared context, incorporates it into their beliefs with minimal hesitation. However, in many situations—including disputes and strategic deception, but also innocent misalignment in good-faith collaborations—new information generates *resistance* to belief revision. In these cases, not all updates fit so neatly. Sometimes, new information conflicts with the listener's prior understanding, challenges their assumptions, or signals a hidden agenda. Here, the process of updating belief states is not "frictionless." Instead, the listener encounters a kind of "resistance" to easy assimilation, a phenomenon we call *epistemic friction*.

Friction in conversational updates reflects an underlying complexity in how we process and accommodate new information, while pointing to deeper inferential processes within the participants' epistemic state. Understanding friction can help us identify when a speaker might be deceptive, when a conversation is strategically misaligned, or when a seemingly simple statement actually encodes a more complex epistemic move. In short, friction offers insight into the subtle interplay between logical inference, pragmatic reasoning, and the architecture of cognitive representations.

In physical systems, friction is a force that resists motion. By analogy, epistemic friction is a resistance to the smooth "motion" of belief revision. This resistance might be epistemically beneficial—encouraging the listener to scrutinize the new information more carefully, or to consider alternative explanations. It might also expose underlying strategic interests, deceptive behavior, or complexities in the conceptual structure of what is being communicated. Here, we explore frictive interactions in terms of evidence-based dynamic epistemic logic (DEL; Van Benthem and Pacuit (2011)), a well-established logical framework for modeling belief updates, as recently explored in (Khebour et al., 2024b).

We introduce a vector-based modeling approach, drawing on Holographic Reduced Representations (HRR) (Plate, 1995; Luo et al., 2018) and related vector symbolic architectures (Kanerva, 1988). This approach treats agents' belief states and propositions as high-dimensional vectors, allowing geometric notions like orthogonality and angle to characterize the friction that arises when assimilating new information. By bridging the gap between

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 323–333
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

symbolic logic and geometric intuition, this model provides a novel perspective on the cognitive and communicative processes underlying conversation.

Finally, we provide case studies from a situated collaborative task that demonstrate how this model of epistemic friction can be used to create a straightforward vectorization of task-relevant propositionalized beliefs and their subsequent updates in the face of new interlocutor assertions. Our analyses demonstrate the utility of epistemic friction in both modeling dialogues and in human-AI interactions, and we subsequently discuss how the model of belief alignment as a measure of epistemic resistance or friction can naturally be made more sophisticated to accommodate the complexities of real-world dialogue scenarios.

2 Related Work

Epistemic friction is clearly related to the classic notions of miscommunication and misalignment of common ground in conversation (Grice, 1975; Asher and Gillies, 2003; Stalnaker, 2002; Traum and Larsson, 2003). The concept of common ground refers to the set of shared beliefs among participants in a Human-Human interaction (HHI) (Markowska et al.; Traum, 1994; Hadley et al., 2022), as well as HCI (Krishnaswamy and Pustejovsky, 2020; Pustejovsky and Krishnaswamy, 2021; Ohmer et al., 2022) and HRI interactions (Kruijff et al., 2010; Fischer, 2011; Scheutz et al., 2011). When common ground is lacking or divergent, interlocutors experience misunderstandings or must exert an effort to clarify and realign their beliefs (Clark and Wilkes-Gibbs, 1986). Such effortful moments are essentially points of friction. Although friction is typically seen as something to overcome or mitigate in dialogue (Brown et al., 2003; Hunter et al., 2018), friction can also play a beneficial role in the interaction (Chen and Schmidt, 2024).

In Dynamic Epistemic Logic (DEL), degrees of evidence (or strength of belief) towards a proposition, can be seen as correlated to the friction that an agent has towards a public proposition (Van Ditmarsch et al., 2007; van Benthem and Smets, 2015; Van Benthem and Pacuit, 2011). Similarly, in argumentation theory, friction can be seen as analogous to the degree of acceptance or rejection of beliefs in an argument (Baumann and Brewka, 2015; Hunter et al., 2020). From this perspective, friction is not only about the endpoint of belief revision but about the trajectory: how beliefs resist, adapt, or transform as agents encounter a continuous stream of arguments and evidence.

Beyond logical and probabilistic formalisms, researchers have explored vector-space representations of propositions within distributional models (Baroni, 2013; Boleda, 2020; Lenci and Sahlgren, 2023), as well as hyperdimensional models (Plate, 1995; Kanerva, 1988; Ginzburg et al., 2024; Obiso, 2024). Within the areas of dialogue and multiparty interactions, vector models of propositional content have been employed in the service of tracking common ground (Khebour et al., 2024b; Zhu et al., 2024; Palmer et al., 2024).

3 Epistemic Friction in Communication

A core assumption in many theories of discourse, ranging from Grice's cooperative principle (Grice, 1975) to Stalnaker's common ground framework (Stalnaker, 2002), is that participants in a conversation share a basis of mutual knowledge and strive for coherence. However, in many situations new information generates *resistance* to belief revision. These situations may include adversarial or cooperative-competitive situations such as disputes or strategic deception (Niculae et al., 2015), but also ordinary good-faith collaboration. In these cases, a listener hears a speaker's assertion and, assuming trust and shared context, incorporates it into their beliefs with minimal hesitation. Nevertheless, not all updates fit so neatly. Sometimes, new information conflicts with the listener's prior understanding, challenges their assumptions, signals a hidden agenda, or this misunderstanding or misremembering mutates the information the listener believes they are incorporating . In these cases, the process of updating belief states is not "frictionless." Instead, the listener encounters a kind of "resistance" to easy assimilation, a phenomenon we call *epistemic friction*. In the context of a constantly updating dialgoue, we call this phenomenon *dynamic epistemic friction*.

In DEL, we use a standard modal model, $M = (W, \{R_a\}_{a \in \mathcal{A}}, V)$, where:

(1) a. W is a set of possible worlds;
    b. R_a is the accessibility relation for agent a,
    c. V is a valuation function assigning truth conditions to atomic propositions.

Knowledge or belief operators ($B_a\varphi$) are evaluated by requiring φ to hold in all R_a-accessible worlds.

DEL captures belief change by product updates with event models (Bolander, 2014). Formally, an event model $\mathcal{E} = (E, \{R_a^E\}, \text{pre})$ is combined with M as in (2), where $\otimes$ denotes the product update:

(2) a. $M \otimes \mathcal{E} = (W \times E,\ \{R_a^{\otimes}\},\ V^{\otimes})$
b. where $(w, e)R_a^{\otimes}(w', e')$ iff wR_aw', eR_a^Ee', and $M, w \models pre(e)$ and $M, w' \models pre(e')$.

If an event is *public*, each agent's belief set typically refines (or filters) to those worlds consistent with the event's precondition. Usually, we assume that all agents smoothly integrate the new proposition. But if the proposition conflicts strongly with the agent's prior beliefs, friction ensues.

We say friction occurs when an agent's newly updated beliefs *cannot* be derived by a simple monotonic restriction of the old ones. Formally, consider an agent a with old beliefs B_a^{old}, updated by ψ to B_a^{new}. Alignment is quantified by checking how trivially ψ is entailed by B_a. Friction occurs when updates require epistemic revision, formally:

(3) $B^{new} \nsubseteq B^{old} \cup \{\psi \mid B^{old} \vdash \psi\}$

Conversely, a lack of friction corresponds to minimal cognitive effort in integrating new propositions.

Khebour et al. (2024b) introduce the framework of *evidence-based DEL*, in which common ground is structured into:

(4) a. **QBank** (Questions Under Discussion): Propositions requiring evaluation.
b. **EBank** (Evidence Bank): Propositions with supporting evidence.
c. **FBank** (Fact Bank): Propositions accepted as true.

In this framework, one tracks how propositions move from the *Question Bank (QBank)* to the *Evidence Bank (EBank)* and eventually to *Fact Bank (FBank)* when evidence is deemed sufficient (Ginzburg et al., 1996). When new evidence $[E]\varphi$ enters, high friction signals that φ is *misaligned* with the agent's prior or insufficiently supported. As more supporting evidence accumulates, friction reduces.

How can we infer the beliefs B_a of an agent a? Following Bolander (2014) and Zhu et al. (2024), we can obtain evidence for what an agent believes from what they do, say, or perceive, formalized in the following axioms:

(5) a. **Acting is Believing:** $DO_a\varphi \rightarrow B_a\varphi$ (you believe your own actions)
As an agent participant in an event, you believe it has happened.
b. **Saying is Believing:** $SAY_a\varphi \rightarrow B_a\varphi$ (you believe what you say)
As actor of a declarative speech act, you believe the proposition you express.
c. **Seeing is Believing:** $SEE_a\varphi \rightarrow B_a\varphi$ (you believe what you see)
As witness to a situation or event, you believe it to have occurred.

4 Epistemic Alignment

Suppose an agent a has a belief state $B_a \subseteq W$, where W is the set of possible worlds that the agent considers viable. Let $\{w \in B_a \mid w \models \varphi\}$ be the subset of worlds in which φ holds, and let E be some set of "evidence worlds". In the context of modal logic, B_a functions as a modal operator; in the context of alignment and misalignment, B_a is interpreted as a predefined set. A straightforward way to define epistemic alignment is to define what fraction of a's currently possible worlds also satisfy φ (and are consistent with the evidence E). That is:

$$\text{alignment}(\varphi, B_a, E) = \frac{|\{w \in B_a \mid w \models \varphi\} \cap E|}{|B_a|}$$

If almost all of B_a already support φ, then alignment ≈ 1, so friction is low. If few or none of the worlds in B_a satisfy φ, alignment ≈ 0, so friction is high. One can define "consistent with E" in many ways (e.g., requiring each $w \in B_a$ to also satisfy whatever constraints the evidence imposes). The key idea is that alignment measures how large the overlap is between φ and the agent's current doxastic possibilities, modulated by the evidence.

If we consider the propositional content as dense vector encodings, then we can define $\mathbf{v}_{B_a}$ to be the vector encoding agent a's overall belief state, $\mathbf{v}_\varphi$ to be a vector encoding the proposition φ, and $\mathbf{v}_E$ to be a vector encoding relevant evidence E. A natural strategy is to use cosine similarity due to its prevalence in HRR (Plate, 1995). However, the choice of similarity function may depend on the algebras or symbolic logic used to represent propositions in a given system (Kleyko et al., 2022, 2023). This function should also be chosen based on the way propositional content is vectorized in the propositionalized vector.

In our case, a simple encoding treats "$\varphi + E$" as the combined proposition-plus-evidence vector, measuring its similarity to the agent's belief vector:

$$\text{alignment}(\varphi, B_a, E) = \text{CosSim}(\mathbf{v}_{B_a}, \mathbf{v}_\varphi + \mathbf{v}_E)$$

A large positive dot product indicates high alignment, while a near-zero or negative dot product indicates strong orthogonality or conflict, meaning the agent's existing beliefs are quite distant from φ, so friction is higher.

To weight the evidence differently in order to model uncertainty, one could add coefficients (e.g. $\lambda_1 \mathbf{v}_\varphi + \lambda_2 \mathbf{v}_E$) or use other similarity measures. The core idea is that "alignment" = "similarity" between the combined proposition/evidence vector and the agent's belief vector.

In the previous section, we have characterized friction $F(\varphi, B, E)$ as proportional to "misalignment." That is,

(6) $F(\varphi, B, E) \;\propto\; 1 - \text{alignment}(\varphi, B, E)$.

So when alignment is high, friction is low, and vice versa. We use the term "orthogonal" to indicate that the new proposition is "hard to assimilate." Orthogonality in vector spaces (cosine near zero) naturally corresponds to low alignment.

In both the set-theoretic and vector-based versions, one can incorporate E to reflect how evidence changes the "effective proposition." More (or stronger) evidence typically boosts alignment with B, reducing friction.

Friction in epistemic updates occurs when new evidence $[E]\varphi$ conflicts with or is near-orthogonal to the agent's current belief state $[B]\neg\varphi$. Given the evidence-based DEL framework from the previous section, we can assume that friction modifies how propositions transition between the different banks. The transition rules from bank to bank can be viewed as follows:

(7) a. QBank $\xrightarrow{E \text{ sufficient}, F \text{ low}}$ EBank;
b. EBank $\xrightarrow{F \text{ near-zero}}$ FBank.

4.1 Friction Equilibrium in Discourse

Dynamic Epistemic Friction (DEF) quantifies the resistance encountered during belief updates. Our goal is to iteratively reduce friction in discourse in order to guide participants toward a better epistemic equilibrium. To this end, we assume:

(8) a. $\mathcal{D} = [\varphi_1, \varphi_2, \ldots, \varphi_n]$: The set of propositions in the discourse;
b. $\mathcal{S} = [B_1, B_2, \ldots, B_m]$: The epistemic states of participants;
c. $E = [E_1, E_2 \ldots E_n]$: Evidence associated with each proposition φ_i.

We then proceed as follows:

(9) **Initialize** the belief set:
Start with $\mathcal{D}^0 = \mathcal{D}$ and $\mathcal{S}^0 = \mathcal{S}$.
Set iteration $k = 0$. This defines the basic elements required to measure friction and move toward equilibrium: the propositions discussed, the belief states of the participants, and the evidence supporting each proposition.

(10) **Measure Friction:** For each proposition $\phi_i \in \mathcal{D}^k$: $F_i(a) = 1 - \text{alignment}(\phi_i, B_a, E_i)$, where $F_i(a)$ is the friction for participant a. Start the iterative equilibrium process from an initial state (no friction measured yet). Here, friction is measured by how misaligned each participant's belief state is with each proposition, given the available evidence: high alignment means that the participant's beliefs easily incorporate the proposition, resulting in low friction; low alignment means substantial disagreement or conflict, indicating high friction and a need for epistemic revision.

(11) **Identify High-Friction Propositions:** For any a, extract propositions ϕ_i where $F_i(a) > T$, the threshold for high friction. Let $\mathcal{H}$ denote these high-friction propositions. Such propositions are difficult for at least one participant to integrate into their beliefs, signaling a need for further discussion or clarification.

(12) **Rank Propositions by Friction:** Rank $\mathcal{H}$ by their average friction:

$$\text{Rank}(\phi_i) = \frac{1}{m} \sum_{a=1}^{m} F_i(a).$$

Propositions are prioritized by how difficult (on average) they are to assimilate across all m participants. Propositions with the highest average friction are candidates for clarification or refinement first, representing the greatest obstacle to achieving shared understanding.

(13) **Refine High-Friction Propositions:** For the top-ranked $\phi_j \in \mathcal{H}$, propose a refinement ϕ_j^*:

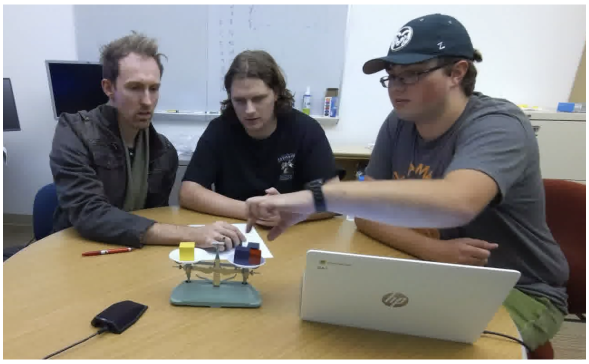


Figure 1: A group of 3 performing the Weights Task.

(i) add evidence E'_j, making the proposition easier to accept or (ii) modify ϕ_j for better alignment with the current belief state.

(14) **Update Belief States:** For each participant a:

$$B_a^{k+1} = B_a^k + \Delta B_a,$$

where $\Delta B_a = -\nabla F(\phi_j^*, B_a, E'_j)$. Beliefs are updated by applying a gradient step, effectively moving the belief states in a direction that reduces friction. The gradient descent step systematically adjusts participants' beliefs closer to propositions supported by evidence.

(15) **Check Equilibrium:** Measure net friction:

$$\mathcal{F}^k = \frac{1}{n \times m} \sum_{i=1}^{n} \sum_{a=1}^{m} F_i(a).$$

If $\mathcal{F}^k \leq T$, return equilibrium $\mathcal{D}^k, \mathcal{S}^k$. This computes the net friction averaged across all propositions and participants, and quantifies how well the group is aligned as a whole. If the net friction is less than or equal to some threshold T, equilibrium is achieved. The participants' beliefs are now sufficiently aligned and no more substantial cognitive effort is required to maintain common ground.

(16) **Iterate or Halt:** If $k < \mu$, the maximum number of iterations, set $k = k + 1$ and repeat. Otherwise, report no equilibrium.

5 Empirical Demonstration

In this section we illustrate how the formal model detailed above can be operationalized to show how DEF can predict updates in the implied beliefs of real dialogue participants in a situated collaborative task. We experiment on the Weights Task Dataset (WTD; (Khebour et al., 2024a)), in which triads collaborate to deduce the weights of differently-colored blocks using a balance scale (Fig. 1). The correct block weight assignments are $[red = 10g, blue = 10g, green = 20g, purple = 30g, yellow = 50g]$. The Weights Task is a collaborative task with one ideal convergent outcome. Use of this fixed condition allows the use of the aforementioned formal model in an analysis that can rigorously quantify the trajectory of convergence relative to a consistent ground truth and fit this model to the dynamics of any group, by controlling for the expected outcome while varying the individual participants.

We perform an experimental evaluation over 4 of of the 10 groups in the WTD, which are fully annotated with dialogue transcripts and the beliefs asserted by the three participants in each group (VanderHoeven et al., 2025), as indicated by speech, gesture, gaze, and action.

For these analyses we adopt a simplified model of propositionalized belief states that can be used to construct multidimensional sparse vectors according to the assumptions given in (5), with a specific emphasis on *Saying is Believing*. Belief states are vectorized such that logical operators can be realized as arithmetic and algebraic operations, which gives intuitive properties like "alignment" and "irrelevance" analogies in measures like similarity and orthgonality (Sec. 4).

Given the 5 blocks in the task, belief states are vectorized in $\mathbb{R}^5$, ordered component-wise as in Khebour et al. (2024a,b) ($[red, blue, green, purple, yellow]$). Thus, an assertion of $red = 10 \land blue = 10$ is represented as $[10, 10, 0, 0, 0]$ indicating affirmative assertions regarding the weights of the red and blue blocks. The 0 components for the other blocks represent that no information regarding them is being asserted. Similarly, $green \neq 20$ would be represented as $[0, 0, -20, 0, 0]$ (negative positioning toward $green = 20$, no other information asserted). Where blocks are related to other blocks by inequalities, the belief vector encodes a lower or upper bound regarding that block, such that $yellow < 40$ becomes $[0, 0, 0, 0, 40 - \mathcal{U}(0, 1)]$, thus anchoring the assertion relative to that weight value, in the appropriate direction.

Worked Example Consider the novel dialogue in Table 1, generated using GPT-4 given a description of the task setup and goals. P1's assertion that

P1:	Alright team, let's start weighing the blocks! Since we know the red block is 10 grams, should we weigh the blue block against it?
P2:	Great idea, I feel like the blue block will also be 10 grams. Let's do it!
P1:	[*Weighs red and blue blocks*] They balance! So, the blue block is also 10 grams.
P2:	Confirmed! That's awesome. Now, what should we weigh next?

Table 1: Example generated dialogue.

the red and blue blocks both weigh 10 grams would be vectorized as $[10, 10, 0, 0, 0]$. Now consider a "frictive" utterance that pushes back on some of this assertion, which may inserted by an AI agent or another participant: "*Hey, let's not jump to conclusions about the blue block's weight just yet. What if it's not 10 grams?*" This assertion, expressing (conditionally) that $blue \neq 10$, would be vectorized as $[0, -10, 0, 0, 0]$.

Now, letting $\vec{\varphi}_a$ be the focus participant's current belief vector, $\vec{\varphi}_b$ be the vector expressing the interlocutor's utterance, and $s = \text{sgn}(\vec{\varphi}_a \cdot \vec{\varphi}_b)$, consider an update operation akin to (14):

$$(17)\quad \vec{\varphi}'_a = \vec{\varphi}_a + \min(\beta, \alpha \times s) \times \text{CosSim}(\vec{\varphi}_a, \vec{\varphi}_b) \odot \vec{\varphi}_b.$$

Here, we introduce some *friction coefficients* that allow us to tune how much empirical effect friction has on the belief update: α expresses how much "force" to apply the friction with (e.g., a scalar multiple of the gradient step), and β establishes a "ceiling" on how much an assertion $\vec{\varphi}_b$ that is roughly aligned with $\vec{\varphi}_a$ can reinforce or "accelerate" it toward the status of an established belief, when compared to how much a contradictory or frictive assertion $\vec{\varphi}_b$ would suppress $\vec{\varphi}_a$.

Given the above $\vec{\varphi}_a = [10, 10, 0, 0, 0]$ and $\vec{\varphi}_b = [0, -10, 0, 0, 0]$, with $\alpha = 1$ and $\beta = 1$, the updated belief state $\vec{\varphi}'_a$ after applying (17) becomes $[10, 2.929, 0, 0, 0]$. The assertion contradictory to $blue = 10$ renders it a "frictive" proposition and has lessened P1's epistemic commitment toward it. The precise component-wise values in the vector should not be taken to indicate what the participant believes the weight of the relevant block to be, but rather as an indicator of the *degree of belief* they have in the block's weight being the value assigned to it by the ground truth value assignment.

An interesting effect of these operations is that in certain circumstances when an assertion expresses information contrary to certain elements of the belief state but aligned with others, the effect may be greater on the component of the belief state against which friction is exercised. I.e., given $\vec{\varphi}_a = [10, 10, 20, 0, 0]$, $\vec{\varphi}_b = [10, -10, 20, 0, 0]$, ($\vec{\varphi}_a$ and $\vec{\varphi}_b$ have the same red and green components but opposite blue components), $\alpha = 1$ and $\beta = 1$, the updated $\vec{\varphi}'_a = [10, 3.333, 20, 0, 0]$, but given $\vec{\varphi}_a = [10, 10, 20, 0, 0]$, $\vec{\varphi}_b = [0, -10, 20, 0, 0]$ (only the same green components, but opposite blue components), $\vec{\varphi}'_a = [10, 4.523, 20, 0, 0]$. That is, accordance on certain propositions gives differences more "weight" in the update.

5.1 Experimental Procedure

We adopt this procedure to evaluate the operationalization of our formal model of dynamic epistemic friction on the task of predicting what the final belief state (final state of FBank) of a target participant should be, given the utterances in the dialogue in order. This allows us to iteratively evaluate how the belief state evolves according to the DEF model. Since all groups in the Weights Task successfully deduced the weights of all blocks, the ground truth final state is a fixed $[10, 10, 20, 30, 50]$.

For tractability reasons we focus on modeling the only the belief state of the participant who speaks the *least* in each group.[1] This provides the greatest number of interlocutor utterances that affect the focus participant's belief state without updating it directly due to Saying is Believing (5).

As a consequence of Saying is Believing, if the focus participant makes a statement asserting a block weight or explicitly accepts another participant's positive and specific assertion about a block weight (e.g., $green = 20$, but not $green \neq 20$ or $green > blue$), then that value gets directly assigned to the relevant component in the focus participant's belief vector before the update function is run over subsequent utterances in the dialogue.

Under these conditions, we conduct the following procedure:

1. Initialize the focus participant's "belief vector" from a uniform distribution $\mathcal{U}(0, 10)$ and

---

[1] Which specific participant this is may vary across groups and is not further explicated here.

set the first ("red") element of the belief vector to 10. This reflects the initial state of the Weights Task where participants are told that the red block weighs 10g. The $\mathcal{U}(0, 10)$ initialization ensures that belief vectors are not strictly 0 in most components, allowing for updates to actually shift the vector and reflecting participants' assumption (apparent in the original data) that weight values are somewhere in intervals of 10g.

2. Iterate through the statements or acceptances of propositional content in each group dialogue. For each interlocutor utterance, appropriately encoded as described, apply the update function (17) to the focus participant's belief vector.

3. At the end of each dialogue, extract the focus participant's final vectorized belief state.

4. Fit a ridge regressor (L_2 scaling constant of 1) to map this extracted belief state to the ground truth final FBank $[10, 10, 20, 30, 50]$. We use a rotating leave-one-group-out split, such that we fit to the extracted final states from 3 of the annotated WTD dialogues and test on the remaining.

This procedure allows us to test how well the final extracted belief state, as constructed using DEF interpretation of the naturally occurring friction in the dialogue, predicts the actual final FBank at the conclusion of the task. Due to inherent stochasticity in steps 1 and 4, we ran the aforementioned loop 100 times and average over the outcomes. We also conduct a variant where the belief state features extracted in step 3 include the concatenated final k belief states in each dialogue, for $k \in \{1..4\}$. We use root mean-squared error for our primary metric, which puts error back in the original units and establishes how many "grams" the final prediction is off by in aggregate.

5.2 Results

Table 2 presents average weight prediction RMSE over each of the 4 test groups using 100 iterations of leave-one-group-out evaluation.

k	**Group 1**	**Group 2**	**Group 4**	**Group 5**
1	$2.613_{\pm 0.421}$	$2.946_{\pm 0.595}$	$7.678_{\pm 0.983}$	$2.229_{\pm 0.233}$
2	$1.889_{\pm 0.217}$	$2.573_{\pm 0.395}$	$10.731_{\pm 0.930}$	$1.953_{\pm 0.204}$
3	$4.449_{\pm 1.740}$	$2.873_{\pm 0.631}$	$13.292_{\pm 1.059}$	$2.368_{\pm 0.453}$
4	$5.187_{\pm 1.011}$	$2.366_{\pm 0.505}$	$17.501_{\pm 2.190}$	$3.112_{\pm 1.012}$

Table 2: Average RMSE on weight prediction from DEF-constructed FBank over the 4 test groups, using update function (17) with friction coefficients $\alpha = 5$ and $\beta = 2$ and dialogue history length k.

Using the FBank constructed with DEF, we are able to get very close to the true weight values, with an average RMSE of 2-3g for most test groups at low k, showing the efficacy of DEF in belief state tracking and prediction.[2] This figure represents error across *all blocks* in all groups. In most cases the block weight introducing the most error into prediction was that of the yellow block. This is likely because the participants deduce the weight of the yellow (largest) block at the very end of the task, and while many utterances in the dialogue reiterate and deliberate upon the weights of the other blocks, fewer utterances discuss the yellow block, meaning there are fewer instances that shift the yellow component of the belief state vector toward the correct value. Thus, propositions pertaining to the yellow block, and $yellow = 50$ particularly, appear to be "high friction" propositions as in (11) above.

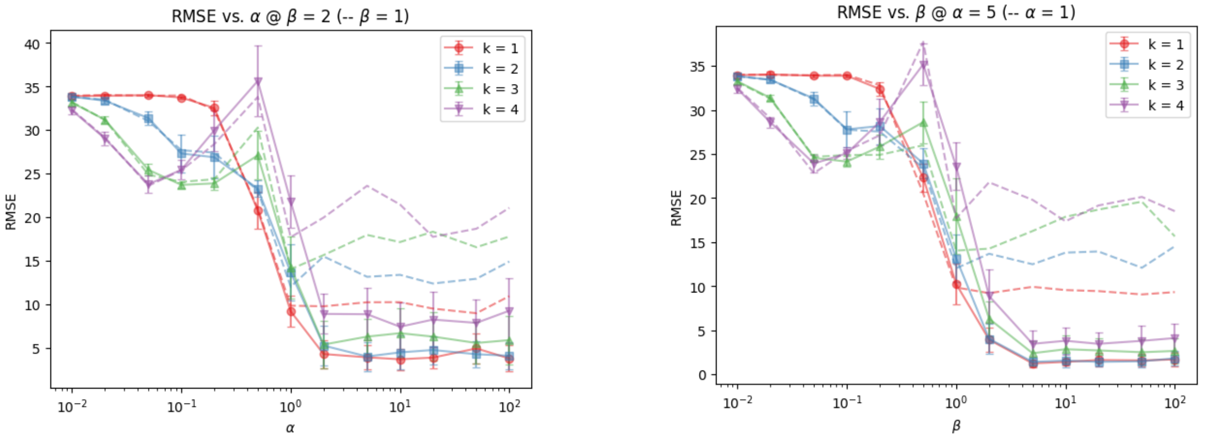
Fig. 2 shows how the different values of friction coefficients α and β as used in the vector update function (17) affect DEF's performance on belief state prediction. We performed a grid search through different values $\in \{0.01..100\}$ with a dialogue history window size k of up to 4, using leave-one-group-out cross-validation. From this search, $\alpha = 5$ and $\beta = 2$ emerged as the best-performing combination.[3] Fig. 2a presents RMSE as a function of α with β fixed at 2, and Fig. 2b presents RMSE as a function of β with α fixed at 5.

These figures show the importance of friction coefficients. α and β are complementary and have similar effects, particularly as strong regularizers. When $k = 4$, meaning longer dialogue history is used, prediction at lower α and β values is noisy, with high RMSE and standard error. The lowest values of α and β are effectively equivalent to a "minimal friction" or "no friction" setting in which interlocutor assertions are naively adopted by the listener. However, as the friction coefficients grow larger, meaning more friction is effected by each

---

[2] Group 4 is a shorter, sparser dialogue with fewer updates, and is therefore noisier.

[3] These values were used to compute the groupwise results in Table 2.



(a) DEF peformance as a function of α with $\beta = 2$ (dashed lines $\beta = 1$ as a default baseline).

(b) DEF peformance as a function of β with $\alpha = 5$ (dashed lines $\alpha = 1$ as a default baseline).

Figure 2: Effects of different α and β values in the vector update function (17) on DEF peformance in FBank prediction. Values shown are averaged over leave-one-group-out cross-validation. Error bars represent standard error over 100 iterations, after cross-validation.

update , error drops dramatically. In other words, without enough friction, beliefs shift too rapidly toward ultimately incorrect positions. With too much, they become unchangeable and a slow trend of increasing error may emerge, particularly with longer dialogue histories and higher β. However, the right modulation of epistemic friction in the dialogue facilitates arriving at equilibrium as in (15), where beliefs are guided toward agreed-upon propositions and remain there, achieving common ground.

Without consideration of epistemic friction, these propositions would naively be immediately adopted by every participant (cf. İnan et al. (2025)) A frictionless setting would not reflect the group dynamics of acceptance or refusal of propositions and would involve a greater error than all models involving friction.

6 Conclusion

In this paper, we presented a formal model of Dynamic Epistemic Friction (DEF) in dialogue, operationalized within the framework of Dynamic Epistemic Logic (DEL) and vector-based belief representations. We draw on the metaphor of friction as a physical force that changes the trajectory of a moving object as it encounters resistance and show that through the lens of DEL, analogous operations describe resistance to or accommodation of belief updates. Through empirical analyses using data from a situated collaborative task (the Weights Task Dataset), our results demonstrate that DEF effectively predicts participant belief updates by quantifying resistance encountered during belief revisions. Specifically, by operationalizing epistemic states and propositional assertions within an evidence-based dynamic logic with vector-based propositional encoding, we show that epistemic friction reliably indicates how smoothly participants integrate new evidence into their existing beliefs.

We should note that for a propositional vectorization as used in Sec. 5 to hold, the vectorized propositional space needs to be at least roughly isotopic (Ethayarajh, 2019; Nath et al., 2023). This property is known to be at best inconsistent in modern LLMs (Machina and Mercer, 2024), and for realistic data where the belief state may not be preannotated as in the WTD, a more sophisticated vectorization needs to be used such that arithmetic and algebraic operations have equivalent logical consequents. In order to retrieve high-quality vectorizations for realistic data, a vector-symbolic method (Goldowsky and Sarathy, 2024) could operate over a library of propositions. The extraction of these propositions from natural language or multimodal data is crucial for implementing dynamic epistemic friction in an end-to-end system (Venkatesha et al., 2024).

While the direct application of LLMs on this task is underexplored, off-the-shelf LLMs are unlikely to be able to operationalize the *quantitative* formalism to belief revision outlined in this paper. İnan et al. (2025) show that friction improves qualitative mental modeling, and OTS systems may be able to provide qualitative judgments about belief at a given point in the task. LLMs specifically aligned with a formal and functional definition of friction may be more adept at quantitative dialogue tasks. For instance, Nath et al. (2025) show that LLMs optimized to be "friction agents" provide more effective interventions and guidance when optimized to be directly sensitive to "frictive states" (dialogue occurrences similar to how we define epistemic fricton here). Pustejovsky and Krishnaswamy (2025) propose three types of optimization strategies that exploit representations of group beliefs at various levels of depth. Following such lines as the above, combining DEF with qualitative judgments may allow for an even more accurate representation of human belief revision. Our theoretical formalism and empirical data are an important stepping-stone that shows the validity of this work in isolation, laying the foundation for further experimentation and implementation in end-to-end systems, especially those involving extraction from natural language.

Future work should investigate a vectorized approach to belief revision in adversarial or competitive tasks. These tasks may explicitly involve deception and other actions unobserved in the Weights Task; they may commonly use forms of communication prohibitive to collaborative settings but conducive to high performance in competitive environments. For example, the game of Diplomacy has been an object of study as a challenging setting for benchmarking communicative AI (Wongkamjan et al., 2024), and contains both cooperative and adversarial elements. The formalism of epistemic friction alloes an analysis of when convergent properties as seen in alliances, suddenly change character, as when former allies become adversaries, but one of the players may not realize this change has occurred. These state changes are accommodated by the DEF formalism, our experimental procedure can be used to detect these changes (in terms of changes in the convergent properties over time), and our experimental results provide a baseline convergent condition to test collaborator and adversary behavior against.

One could also use the common ground framework and dynamic epistemic friction updates to predict agent behavior. The difference between an agent's current belief state and the proposed updated state, as well as task history and agent behavior, can inform a classifier of how an agent might respond (immediate acceptance, counterargument, asking clarifying questions, etc.) to a given scenario at any moment. This analysis would show how deeply our model of dynamic epistemic friction corresponds to agent behavior and how it may serve as a necessary link in instructive or monitoring systems.

Our empirical results show DEF's effectiveness as a model and the importance of properly modulating the amount of friction in a dialogue (as shown as in the tuning of selected friction coefficients), but we did not compare DEF to other approaches as this novel model of friction in dialogue and novel method of evaluating does not have at present any direct competitors in the literature. It is not clear that existing methods, such as those used in the Dialogue State Tracking Challenge (Williams et al., 2016) provide a meaningful comparison.

The friction metaphor serves as a bridge between logical updating operations (e.g., dynamic epistemic logic) and cognitively motivated geometric models (e.g., vector symbolic architectures). Importantly, it highlights the fact that belief change in dialogue is not always straightforward but can generate internal or inter-agent tension, where the appropriate coefficient of friction plays a crucial role in mitigating misunderstanding.

Acknowledgments

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

References

Nicholas Asher and Anthony Gillies. 2003. Common ground, corrections, and coordination. *Argumentation*, 17(4):481–512.

Marco Baroni. 2013. Composition in distributional semantics. *Language and Linguistics Compass*, 7(10):511–522.

Ringo Baumann and Gerhard Brewka. 2015. Agm meets abstract argumentation: expansion and revision for dung frameworks. In *IJCAI*, volume 15, pages 2734–2740.

Thomas Bolander. 2014. Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic.

In *European conference on social intelligence (ECSI 2014)*, pages 87–107.

Gemma Boleda. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6(1):213–234.

Ralf D. Brown, Rebecca Hutchinson, Paul N. Bennett, Jaime G. Carbonell, and Peter Jansen. 2003. Reducing boundary friction using translation-fragment overlap. In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.

Zeya Chen and Ruth Schmidt. 2024. Exploring a behavioral model of "positive friction" in human-ai interaction. In *International Conference on Human-Computer Interaction*, pages 3–22. Springer.

Herbert H Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65.

Kerstin Fischer. 2011. How people talk with robots: Designing dialog to reduce user uncertainty. *AI Magazine*, 32(4):31–38.

Jonathan Ginzburg, Chris Eliasmith, and Andy Lücking. 2024. Swann's name: Towards a dialogical brain semantics. In *Proceedings of the SemDial 2024 Workshop*, Trento, Italy. ACL. Accessed: 2025-03-14.

Jonathan Ginzburg and 1 others. 1996. Dynamics and the semantics of dialogue. *Logic, language and computation*, 1:221–237.

Howard Goldowsky and Vasanth Sarathy. 2024. Analogical reasoning within a conceptual hyperspace. *arXiv preprint arXiv:2411.08684*.

Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, pages 41–58. Brill.

Lauren V. Hadley, Graham Naylor, and Antonia F. de C. Hamilton. 2022. A review of theories and methods in the science of face-to-face social interaction. *Nature Reviews Psychology*, 1(1):42–54. Number: 1 Publisher: Nature Publishing Group.

Anthony Hunter, Sylwia Polberg, and Matthias Thimm. 2020. Epistemic graphs for representing and reasoning with positive and negative influences of arguments. *Artificial Intelligence*, 281:103236.

Julie Hunter, Nicholas Asher, and Alex Lascarides. 2018. A formal semantics for situated conversation. *Semantics and Pragmatics*, 11.

Mert İnan, Anthony Sicilia, Suvodip Dey, Vardhan Dongre, Tejas Srinivasan, Jesse Thomason, Gökhan Tür, Dilek Hakkani-Tür, and Malihe Alikhani. 2025. Better slow than sorry: Introducing positive friction for reliable dialogue systems. *arXiv preprint arXiv:2501.17348*.

Pentti Kanerva. 1988. *Sparse distributed memory*. MIT press.

Ibrahim Khebour, Richard Brutti, Indrani Dey, Rachel Dickler, Kelsey Sikes, Kenneth Lai, Mariah Bradford, Brittany Cates, Paige Hansen, Changsoo Jung, Brett Wisniewski, Corbyn Terpstra, Leanne M Hirshfield, Sadhana Puntambekar, Nathaniel Blanchard, James Pustejovsky, and Nikhil Krishnaswamy. 2024a. When text and speech are not enough: A multimodal dataset of collaboration in a situated task. *Journal of open humanities data*, 10.

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

Denis Kleyko, Dmitri Rachkovskij, Evgeny Osipov, and Abbas Rahimi. 2023. A survey on hyperdimensional computing aka vector symbolic architectures, part ii: Applications, cognitive models, and challenges. *ACM Computing Surveys*, 55(9):1–52.

Denis Kleyko, Dmitri A. Rachkovskij, Evgeny Osipov, and Abbas Rahimi. 2022. A survey on hyperdimensional computing aka vector symbolic architectures, part i: Models and data transformations. *ACM Computing Surveys*, 55(6):1–40.

Nikhil Krishnaswamy and James Pustejovsky. 2020. A formal analysis of multimodal referring strategies under common ground. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5919–5927.

Geert-Jan M Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, Ivana Kruijff-Korbayová, and Nick Hawes. 2010. Situated dialogue processing for human-robot interaction. In *Cognitive systems*, pages 311–364. Springer.

Alessandro Lenci and Magnus Sahlgren. 2023. *Distributional semantics*. Cambridge University Press.

Jiaming Luo, Yuan Cao, and Yonghui Wu. 2018. Towards decomposed linguistic representation with holographic reduced representation.

Anemily Machina and Robert Mercer. 2024. Anisotropy is not inherent to transformers. 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 4892–4907.

Magdalena Markowska, Adil Soubki, Gary Mar, Seyed Abolghasem Mirroshandel, Owen Rambow, and Anita Wasilewska. Formal representation of common ground in dialogue.

Abhijnan Nath, Carine Graff, Andrei Bachinin, and Nikhil Krishnaswamy. 2025. Frictional Agent Alignment Framework: Slow Down and Don't Break Things. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL.

Abhijnan Nath, Sheikh Mannan, and Nikhil Krishnaswamy. 2023. Axomiyaberta: A phonologically-aware transformer model for assamese. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11629–11646.

Vlad Niculae, Srijan Kumar, Jordan Boyd-Graber, and Cristian Danescu-Niculescu-Mizil. 2015. Linguistic harbingers of betrayal: A case study on an online strategy game. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1650–1659.

Timothy Obiso. 2024. Holographic embeddings for text and graphs a master's thesis presented to the faculty of the graduate school of arts and sciences.

Xenia Ohmer, Marko Duda, and Elia Bruni. 2022. Emergence of hierarchical reference systems in multi-agent communication. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5689–5706.

Derek Palmer, Yifan Zhu, Kenneth Lai, Hannah VanderHoeven, Mariah Bradford, Ibrahim Khebour, Carlos Mabrey, Jack Fitzgerald, Nikhil Krishnaswamy, Martha Palmer, and 1 others. 2024. Speech is not enough: Interpreting nonverbal indicators of common knowledge and engagement. *arXiv preprint arXiv:2412.05797*.

Tony A Plate. 1995. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3):623–641.

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

James Pustejovsky and Nikhil Krishnaswamy. 2025. Frictive policy optimization for llm agent interactions. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*.

Matthias Scheutz, Rehj Cantrell, and Paul Schermerhorn. 2011. Toward humanlike task-based dialogue processing for human robot interaction. *Ai Magazine*, 32(4):77–84.

Robert Stalnaker. 2002. Common ground. *Linguistics and philosophy*, 25(5/6):701–721.

David Traum. 1994. A computational theory of grounding in natural language conversation. *PhD thesis, University of Rochester*.

David R Traum and Staffan Larsson. 2003. The information state approach to dialogue management. *Current and new directions in discourse and dialogue*, pages 325–353.

Johan Van Benthem and Eric Pacuit. 2011. Logical dynamics of evidence. In *International Workshop on Logic, Rationality and Interaction*, pages 1–27. Springer.

Johan van Benthem and Sonja Smets. 2015. Dynamic logics of belief change.

Hans Van Ditmarsch, Wiebe van Der Hoek, and Barteld Kooi. 2007. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media.

Hannah VanderHoeven, Brady Bhalla, Ibrahim Khebour, Austin Youngren, Videep Venkatesha, Mariah Bradford, Jack Fitzgerald, Carlos Mabrey, Jingxuan Tu, Yifan Zhu, Kenneth Lai, Changsoo Jung, James Pustejovsky, and Nikhil Krishnaswamy. 2025. TRACE: Real-Time Multimodal Common Ground Tracking in Situated Collaborative Dialogues. In *Proceedings of the 2025 Meeting of the Nations of the Americas Chapter of the Association for Computational Linguistics: System Demonstrations*.

Videep Venkatesha, Abhijnan Nath, Ibrahim Khebour, Avyakta Chelle, Mariah Bradford, Jingxuan Tu, James Pustejovsky, Nathaniel Blanchard, and Nikhil Krishnaswamy. 2024. Propositional extraction from natural speech in small group collaborative tasks. In *Proceedings of the 17th International Conference on Educational Data Mining*, pages 169–180.

Jason D Williams, Antoine Raux, and Matthew Henderson. 2016. The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

Wichayaporn Wongkamjan, Feng Gu, Yanze Wang, Ulf Hermjakob, Jonathan May, Brandon M Stewart, Jonathan Kummerfeld, Denis Peskoff, and Jordan Boyd-Graber. 2024. More victories, less cooperation: Assessing cicero's diplomacy play. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12423–12441.

Yifan Zhu, Hannah VanderHoeven, Kenneth Lai, Mariah Bradford, Christopher Tam, Ibrahim Khebour, Richard Brutti, Nikhil Krishnaswamy, and James Pustejovsky. 2024. Modeling theory of mind in multimodal hci. In *International Conference on Human-Computer Interaction*, pages 205–225. Springer.

A Three-Tier LLM Framework for Forecasting Student Engagement from Qualitative Longitudinal Data

Ahatsham Hayat[1] **Helen Martinez**[1] **Bilal Khan**[2] **Mohammad Rashedul Hasan**[1]
University of Nebraska-Lincoln[1] Lehigh University[2]
aahatsham2@huskers.unl.edu, hasan@unl.edu

Abstract

Forecasting nuanced shifts in student engagement from longitudinal experiential (LE) data—multi-modal, qualitative trajectories of academic experiences over time—remains challenging due to high dimensionality and missingness. We propose a natural language processing (NLP)-driven framework using large language models (LLMs) to forecast binary engagement levels across four dimensions: Lecture Engagement Disposition, Academic Self-Efficacy, Performance Self-Evaluation, and Academic Identity and Value Perception. Evaluated on 960 trajectories from 96 first-year STEM students, our three-tier approach—LLM-informed imputation to generate textual descriptors for missing-not-at-random (MNAR) patterns, zero-shot feature selection via ensemble voting, and fine-tuned LLMs—processes textual non-cognitive responses. LLMs substantially outperform numeric baselines (e.g., Random Forest, LSTM) by capturing contextual nuances in student responses. Encoder-only LLMs surpass decoder-only variants, highlighting architectural strengths for sparse, qualitative LE data. Our framework advances NLP solutions for modeling student engagement from complex LE data, excelling where traditional methods struggle.

1 Introduction

Transformer-based (Vaswani et al., 2017) large language models (LLMs) have significantly advanced natural language processing (NLP), pushing boundaries in text understanding and generation across diverse applications (Bommasani et al., 2021). Beyond excelling in traditional NLP tasks such as summarization and translation (Zhao et al., 2025), LLMs have demonstrated a remarkable capacity for reasoning over complex, context-rich information, suggesting their potential for analyzing sequential and subjective data (Wei et al., 2022; Touvron et al., 2023). One particularly promising, yet relatively untapped, area for LLM application lies in the analysis of **longitudinal experiential (LE)** data—time-series records capturing individuals' evolving perceptions, emotions, and experiences (Xu et al., 2022).

Within educational contexts, LE data offers a unique and valuable perspective on students' subjective engagement, a well-established predictor of retention and academic achievement (Fredricks, 2014; Sinatra et al., 2015). Despite its richness, the inherent characteristics of LE data, including its qualitative nature, temporal dependencies, and frequent missingness, present substantial computational challenges that often limit the effectiveness of traditional machine learning approaches (Xu et al., 2023). Our research focuses on this underexplored intersection of LLMs and the complexities of LE data analysis in education.

In educational research, LE data systematically gathers real-time, self-reported insights—including emotional responses, shifts in motivation, and the development of self-efficacy—from individuals over time, complementing traditional cognitive assessments (Kolb, 1984; Palmer et al., 2010). Understanding these **non-cognitive (NC)** dimensions can reveal critical engagement patterns predictive of academic outcomes, informing timely interventions (Wang et al., 2014; Li et al., 2020).

Our research is based on a dataset of 28 distinct NC features collected weekly from 96 first-year college STEM (science, technology, engineering, mathematics) students across three semesters (Hayat et al., 2024a,b). These features aimed to capture a comprehensive view of their engagement. However, initial analysis revealed that many of these features suffered from **extreme missingness**, with some having up to 100% unanswered responses. To ensure a more robust analysis, we focused on 10 key qualitative NC features that exhibited a response rate of at least 35%. Our forecasting task specifically targets predicting weekly

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 334–347
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

binary engagement shifts (positive vs. negative) across four critical dimensions using 4-week historical sequences, where each prediction involves determining whether a student's engagement level in week 5 will exceed their average from weeks 1-4. These 10 features, despite the inherent challenges of qualitative data, form the basis of our investigation into student engagement forecasting.

Initial attempts to forecast engagement by converting the textual responses of these 10 features into numeric values (e.g., via Likert scale encoding) and training traditional machine learning models like Random Forest (Breiman, 2001) and Support Vector Machines (Hearst et al., 1998), as well as time-series models like LSTMs (Hochreiter and Schmidhuber, 1997), yielded poor forecasting performance. Similarly, directly fine-tuning standard decoder-only and encoder-only LLMs on the raw text of these 10 NC features also resulted in suboptimal forecasting accuracy, although showing marginal improvement over the numeric-based models. This suggests that while LLMs possess inherent advantages, directly processing all available qualitative features, even after initial filtering for missingness, can still introduce noise, hindering their ability to effectively discern predictive signals in this specific type of LE data.

The limitations observed with both traditional numeric approaches and direct LLM fine-tuning underscore the need for a more tailored strategy for analyzing this qualitative, time-series LE data with significant missingness. Unlike traditional time-series models (e.g., ARIMA (Box et al., 2015), LSTMs (Hochreiter and Schmidhuber, 1997)), which struggle with non-numeric input and are particularly vulnerable to biases introduced by missing data, LLMs offer the potential to directly process qualitative information.

As highlighted earlier, processing LE data presents a complex array of challenges, including its qualitative nature, temporal dependencies, and significant sparsity due to missing self-reports. These difficulties are further compounded by the prevalence of missing-not-at-random (MNAR) patterns (Rubin, 1976), where the absence of a report is often correlated with the very engagement phenomena we aim to study. This introduces biases that conventional statistical imputation techniques, such as Last Observation Carried Forward (LOCF), are often inadequate to handle effectively (Schafer, 1997).

To address this critical issue of biased missingness and the noise within the qualitative LE feature space, we propose a *three-tier LLM framework* specifically designed for the unique characteristics of this LE data: (1) **LLM-informed imputation**, using LLMs' contextual understanding to generate textual descriptors for missing values, mitigating MNAR bias where traditional methods fall short; (2) **LLM-based zero-shot feature selection**, employing a panel of expert LLMs to infer and select the most relevant subset of our 10 qualitative NC features via majority voting, thereby reducing noise; and (3) **fine-tuned forecasting**, comparing decoder-only and encoder-only LLMs to predict binary engagement levels for *four key dimensions: Lecture Engagement Disposition (LED), Academic Self-Efficacy (ASE), Performance Self-Evaluation (PSE), and Academic Identity and Value Perception (AIVP).*

Evaluated on 960 overlapping 4-week trajectories (weeks 1-4 predicting week 5) derived from our dataset, our three-tier approach significantly outperforms numeric baselines. Ablation studies further demonstrate the efficacy of each component: (1) zero-shot feature selection yields substantial gains compared to using all 10 NC features, highlighting the noise reduction achieved through expert LLM guidance; and (2) LLM-based feature selection surpasses numeric feature-based models that utilize all available features, directly justifying the need for our LLM-driven feature selection process for this qualitative data. Encoder-only architectures consistently outperform decoder-only variants in this sparse LE forecasting task. This work contributes to the advancement of NLP by reframing qualitative time-series forecasting as a language problem.

Our main contributions are summarized as follows.

- A three-tier LLM framework tackling qualitative LE data's noise and MNAR missingness via imputation and feature selection.
- A novel zero-shot LLM selection method, outperforming numeric baselines on textual time-series.
- Evidence of LLMs' superiority for sparse, subjective sequences, advancing NLP's temporal scope.

2 Related Work

This research leverages LLMs for time-series forecasting, extending their NLP strengths to qualitative LE data in education. Transformer-based LLMs like TimeGPT (Garza et al., 2024) and PromptCast (Xue and Salim, 2024) verbalize numeric time-series for prediction, with data-centric approaches transforming sequences into text for pre-trained LMs (Jin et al., 2024) and model-centric methods fine-tuning LMs for temporal tasks (Zhou et al., 2023). Our model-centric approach fine-tunes LLMs for subjective LE sequences, diverging from numeric trends to target engagement attributes—a domain underexplored by existing LLM-based time-series models despite their sequential modeling prowess.

Student engagement forecasting in educational analytics often relies on cognitive (e.g., grades) or behavioral (e.g., clickstreams) data, using ML methods like LSTMs and Random Forests (Xu and Ouyang, 2022). Recent work incorporates NC factors—self-efficacy, motivation—from surveys (Fredricks, 2014), yet struggles with textual responses, temporal dynamics, and MNAR missingness prevalent in LE data (Sinatra et al., 2015). Unlike these numeric-focused efforts, our framework verbalizes weekly NC trajectories for LLM processing, forecasting binary engagement levels and bridging educational analytics with NLP's textual capabilities, addressing a gap in longitudinal engagement modeling.

Handling missing data and feature selection in LE sequences poses further challenges. Traditional imputation (e.g., MICE (van Buuren and Groothuis-Oudshoorn, 2011)) assumes MCAR/MAR, faltering with MNAR patterns (e.g., disengagement-driven skips) and LE's qualitative heterogeneity (Rubin, 1976). Generative models like GAIN (Yoon et al., 2018) impute numeric values but lack context for textual NC features, while standard feature selection (e.g., variance thresholding (Jain et al., 2000)) misses nuanced semantic relevance. Our three-tier framework—LLM-informed imputation (GPT-4o), zero-shot feature selection, and fine-tuned forecasting—outperforms these by capturing MNAR context and selecting predictive NC subsets, leveraging LLMs' reasoning for sparse, subjective data. See Appendix A.2 for a detailed discussion.

3 Three-Tier LLM Framework

This section details our three-tier NLP framework for forecasting weekly student engagement levels from qualitative LE data, designed to address the challenges of MNAR missingness and noise in the feature space. The framework, illustrated in Figure 1, consists of: (1) LLM-informed imputation to address MNAR gaps, (2) zero-shot feature selection via an ensemble of expert LLMs, and (3) fine-tuned forecasting with diverse LLM architectures. These tiers transform sparse, qualitative NC sequences into predictive models, evaluated against numeric baselines.

3.1 Dataset

We utilize a dataset from 96 first-year college students in introductory programming courses at a U.S. public university, collected over 15 weeks per semester across three semesters (Hayat et al., 2024a,b). The data captures 78-dimensional academic experiential trajectories across three modalities: 9-dimensional background data (e.g., demographics, socioeconomic status), 41-dimensional cognitive data (e.g., quiz scores, coding assignment grades), and 28-dimensional NC data (e.g., self-reported motivation, lecture engagement). Background data derives from an initial web survey, cognitive data from the course learning management system, and NC data from daily, context-adaptive questions via a privacy-preserving smartphone app, stored anonymously on cloud servers.

For forecasting, we focus on the NC data, comprising responses to 28 questions targeting behavioral, emotional, and cognitive engagement dimensions (e.g., "How much are you looking forward to your CS1 class lecture today?"). Due to high missingness—over 90% for 18 questions, with some entirely unanswered—we curated 10 key qualitative NC features with at least 35% response rates, detailed in Appendix A.1. These 10 features represent our curated set of key qualitative non-cognitive indicators of student engagement, chosen after addressing the issue of high missingness in the initial 28 features. Using a sliding window, we construct 4-week sequences to predict the subsequent week's engagement shift (e.g., weeks 1-4 predict week 5), yielding 960 trajectories (96 students × 10 predictions per semester). Each trajectory targets **four binary engagement outcomes**—Lecture Engagement Disposition, Academic Self-Efficacy, Performance Self-Evaluation, and Academic Iden-

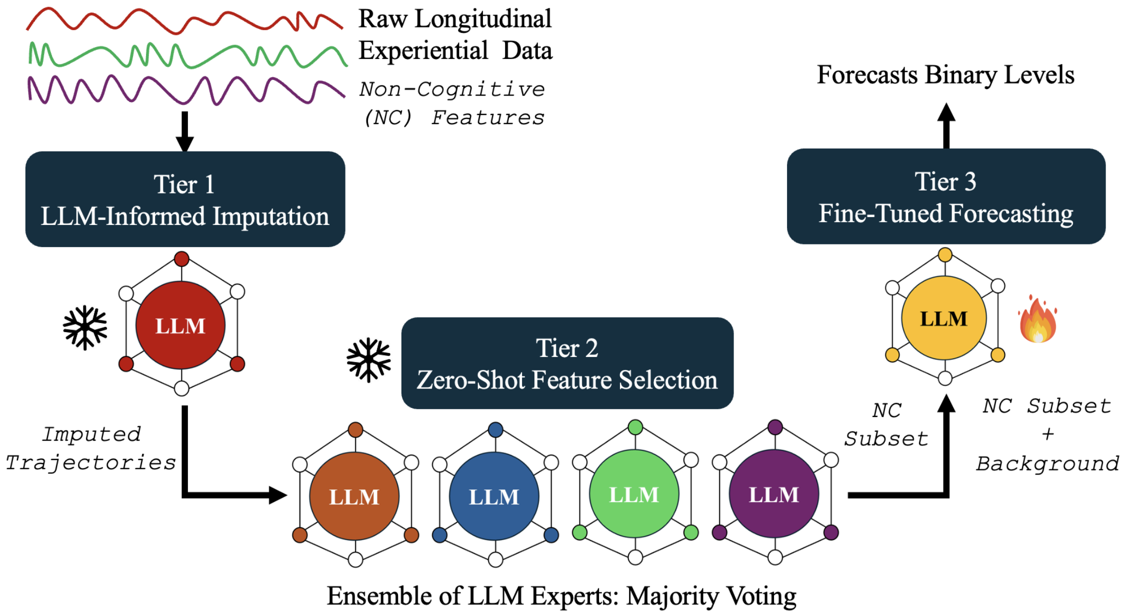


Figure 1: Three-tier LLM framework: (1) LLM-informed imputation fills MNAR gaps in LE trajectories, (2) zero-shot feature selection by expert LLMs curates NC subsets, and (3) fine-tuned LLMs forecast binary engagement levels, enhanced by background features.

tity and Value Perception—derived as composite scores from the 10 NC features.

3.2 LLM-Informed Imputation

The NC data exhibits significant missingness (e.g., 66% of responses missing in week 1, 37% of students skipping questions for over two weeks), often due to students skipping questions or uninstalling the app—patterns indicative of MNAR behavior (Rubin, 1976). Traditional imputation methods like LOCF (Liu, 2016) are unsuitable, as entire weekly response sets may be absent, leaving no prior values to propagate, and numeric imputation risks introducing bias by ignoring MNAR's semantic context (Little and Rubin, 2019). To address this critical issue of biased missingness, we employ GPT-4o (OpenAI, 2024) in a zero-shot manner to generate textual descriptors for missing responses (e.g., "The student skipped this question" or "No response due to app uninstallation"), preserving contextual meaning without forcing numeric assumptions.

For each missing response in our dataset, we construct a detailed prompt that includes surrounding NC data, such as responses from prior or subsequent weeks, and contextual metadata, such as question type and week number. This information serves as the dataset features information provided to GPT-4o, enabling it to infer a descriptor for the missing response. GPT-4o processes these prompts zero-shot—without task-specific training—leveraging its linguistic reasoning to infer descriptors that reflect MNAR dynamics (e.g., disengagement patterns). This approach enhances data quality by embedding semantic context into the 960 trajectories, enabling downstream feature selection and forecasting to exploit qualitative signals overlooked by statistical methods (Little and Rubin, 2019).

3.3 Zero-Shot Feature Selection

Our dataset's 10 curated NC features, reduced from an initial set of 28 due to extreme missingness, form a semantically rich yet sparse space requiring feature selection to optimize forecasting by reducing noise inherent in the qualitative feature space (Guyon and Elisseeff, 2003). Traditional methods—e.g., variance thresholding, correlation analysis (Jain et al., 2000), or attention-based deep learning (Ying et al., 2024)—rely on statistical distributions or labeled data, often missing qualitative, non-linear relationships in LE sequences. Instead, we propose a zero-shot feature selection method using **an ensemble of expert LLMs**: GPT-4o (OpenAI, 2024), Google Gemini (Team et al., 2024a), DeepSeek (DeepSeek-AI et al., 2025), and Microsoft Copilot (Copilot, 2024). This panel leverages each model's linguistic reasoning and world knowledge to identify predictive NC subsets for four engagement dimensions without accessing the data itself (Kojima et al., 2022).

Mathematical Formulation. Formally, let $\mathbf{X}$ denote the dataset of student responses, where each instance $\mathbf{X}^i = (X_1^i, \ldots, X_d^i)$ is a d-dimensional vector, and $d = 10$ represents the curated NC features (e.g., X_1^i: motivation, X_5^i: lecture enjoyment). For each engagement dimension k (e.g., $k = \text{LED}$ for Lecture Engagement Disposition), we define a candidate feature set $\mathbf{F} = \{X_1, \ldots, X_{10}\}$ and seek an optimal subset $\mathbf{S}_k \subseteq \mathbf{F}$ that maximizes predictive relevance for target Y_k. Unlike statistical methods that require data access, our ensemble operates zero-shot: given only semantic descriptions of the 10 features and target definitions, each LLM M_j independently produces a ranking R_j^k of features by inferred relevance. We aggregate these rankings via majority voting, where a feature X_i is included in $\mathbf{S}_k$ if selected by at least $\lceil J/2 \rceil$ models, where $J = 4$ is the ensemble size. This yields consensus-driven subsets $\mathbf{S}_k$ that capture semantic relationships across multiple expert perspectives.

Implementation via Unified Expert Prompt. To systematically guide the feature selection process across all four engagement dimensions, we employ a single comprehensive prompt that leverages psychological domain expertise. Each LLM in our ensemble receives the following structured prompt:

> *"You are an expert psychologist analyzing and predicting student engagement. Given a set of survey questions (e.g., **Q1**: How much are you looking forward to today's lecture?, **Q5**: How much did you enjoy today's lecture?, **Q18**: How confident are you in your programming skills?, . . .), identify the most predictive ones for forecasting students' engagement levels in the following domains for the upcoming week:"*
>
> 1. *Lecture Engagement Disposition*
> 2. *Academic Self-Efficacy*
> 3. *Performance Self-Evaluation*
> 4. *Academic Identity and Value Perception*

This unified prompting approach ensures consistency across the ensemble while allowing each LLM to apply its domain knowledge to identify dimension-specific feature subsets. The streamlined prompt structure enables each model to consider all available features and make informed selections based on psychological theory and semantic relationships between questions and target constructs.

The systematic application of this expert prompt yields tailored feature subsets: Lecture Engagement Disposition $\mathbf{S}_{\text{LED}} = \{Q1, Q5\}$, Academic Self-Efficacy $\mathbf{S}_{\text{ASE}} = \{Q18, Q19, Q20\}$, Performance Self-Evaluation $\mathbf{S}_{\text{PSE}} = \{Q21, Q22, Q23\}$, and Academic Identity and Value Perception $\mathbf{S}_{\text{AIVP}} = \{Q24, Q25\}$. By reasoning over semantic nuance (e.g., prioritizing "lecture enjoyment" over "general motivation" for LED), the ensemble captures contextual relationships statistical methods overlook. This data-agnostic, scalable approach leverages LLMs' prior knowledge, offering a novel alternative to traditional feature selection for qualitative time-series tasks (Kojima et al., 2022).

We acknowledge that our current implementation relies on closed-source LLMs (GPT-4o, Google Gemini, DeepSeek, and Microsoft Copilot), though the framework is adaptable to open-source alternatives such as Llama (Touvron et al., 2023) or Mistral (Mistral AI, 2024) for enhanced reproducibility.

3.4 Data Preprocessing

To generate binary labels for our 960 trajectories, we score NC responses on a scale capturing engagement intensity (e.g., for X_1: "I am really looking forward to it" = 1, "I am not planning to attend" = -1, "I am kind of looking forward" = 0.5, "I am not really looking forward" = -0.5). For each student and week, we compute a composite score per dimension by averaging the subset's scores selected for that dimension (e.g., for Lecture Engagement Disposition: $(X_1 + X_5)/2$, aggregating daily responses). For a 4-week sequence (e.g., weeks 1–4), we calculate the week 5 score; a positive shift ($Y_k = 1$) is assigned if the week 5 score exceeds the 4-week average, otherwise negative ($Y_k = 0$), yielding a positive-to-negative ratio between 60:40 and 70:30 across the four dimensions.

For **baseline models**, we convert NC responses into numeric features using these assigned scores (e.g., $X_1 = 1$ for "I am really looking forward to it"), preserving the 10-feature structure post-selection (Section 3.3). Missing values, reflecting MNAR patterns, are imputed with zeros, forming 36-D vectors (10 features × 4 weeks) or 4 × 10-D sequences for model input. For **LLMs**, we verbalize these imputed 4-week sequences into natural language narratives, integrating GPT-4o descriptors from Section 3.2 to leverage text-processing strengths (Radford et al., 2019). For example, a sequence might read: "Week 1, Monday: Prior to the lecture, the student reported *I am not looking forward to it*; in the evening, they reflected: *I did not enjoy the lecture*. Week 2: [imputed] *skipped the question*". This transformation embeds daily responses (e.g., X_1: motivation, X_5: enjoyment) and imputed MNAR patterns, tailored to the four engagement dimensions, preparing data for fine-tuning with qualitative context intact.

3.5 Fine-Tuned Forecasting with LLM Architectures

We forecast binary engagement levels (positive vs. negative) over 10 weeks (weeks 5–14) using the 960 verbalized trajectories (Section 3.4). *Two different LLM architecture classes* are fine-tuned for binary classification of student engagement across key dimensions: decoder-only models (Gemma2 9B (Team et al., 2024b), Mixtral 8x7B (Jiang et al., 2024), Llama 7B (Touvron et al., 2023)) and encoder-only models (RoBERTa (Liu et al., 2021), DistilBERT-base-uncased (Sanh et al., 2020)).

Decoder-only models leverage autoregressive reasoning to model the narrative complexity and temporal dependencies of verbalized NC trajectories (e.g., 4-week sequences with MNAR-imputed text), potentially capturing nuanced shifts in qualitative LE data. Specifically, we include Gemma2 9B for its strong performance and efficiency, Mixtral 8x7B as a sparse mixture-of-experts model known for its high quality and fast inference, and Llama 7B as a widely adopted and well-studied foundational model.

Conversely, encoder-only models excel at bidirectional sequence encoding, optimizing discriminative power for sparse, noisy inputs by focusing on contextual feature interactions—critical for our 960 trajectories with varying missingness (35%–100%). We select RoBERTa for its robust pretraining and state-of-the-art results on various classification tasks, and DistilBERT-base-uncased as a computationally efficient yet effective transformer model, allowing us to explore the trade-off between model size and performance.

This dual selection tests architectural suitability: generative flexibility for sequential coherence vs. compact representation for classification efficacy.

Performance is evaluated using balanced accuracy and macro-F1, against numeric baselines: Random Forest (Breiman, 2001), Support Vector Machines (Hearst et al., 1998), 1D CNN (O'Shea and Nash, 2015), LSTM (Hochreiter and Schmidhuber, 1997), and Transformer (Vaswani et al., 2017). Baselines use scored responses (no verbalization).

4 Experiments and Results

We evaluate our three-tier LLM framework—LLM-informed imputation, zero-shot feature selection, and fine-tuned forecasting—against numeric baselines to demonstrate its effectiveness in forecasting student engagement from qualitative LE data. Key comparisons assess: (1) baseline machine and deep learning models with numeric NC subset features, (2) LLMs with verbalized NC subset features, and (3) LLMs with NC subset plus background features. Two ablation studies further explore feature quantity (subset vs. all NC features) and input modality (textual LLMs vs. numeric baselines with all features), validating LLMs' superiority and selection benefits for noisy, MNAR-impaired data.

4.1 Experimental Setup

The dataset comprises 960 trajectories (Section 3.1), split into 70% training (672 trajectories), 15% validation (144), and 15% testing (144), with positive-to-negative class ratios ranging from 60:40 to 70:30 across four dimensions: Lecture Engagement Disposition (LED), Academic Self-Efficacy (ASE), Performance Self-Efficacy (PSE), and Academic Identity and Value Perception (AIVP). Three configurations are tested: (1) numeric NC subset features (e.g., LED: Q1, Q5 post zero-shot selection, Section 3.3), (2) verbalized NC subset features, and (3) verbalized NC subset features plus 9 background features (e.g., demographics) appended as "Background: Female, Mechanical Engineering Major...".

Baseline models—Random Forest (RF, 100 trees), SVM (RBF kernel), 1D CNN, LSTM, and Transformer—are implemented via scikit-learn, trained on numeric NC subset features (e.g., LED: Q1, Q5 as scores, Section 3.4). The 1D CNN uses two convolutional layers with max-pooling, followed by fully connected and dropout layers.

Table 1: Baseline Performance Across Dimensions Using Numeric NC Subset Features

Model	**LED**		**ASE**		**PSE**		**AIVP**	
	B.Acc.	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**
Random Forest	54.5	53.5	46.0	44.5	**53.5**	**52.5**	44.0	41.0
SVM	52.0	48.0	50.0	41.0	51.5	47.0	50.0	39.0
1D CNN	49.0	49.0	48.5	46.0	42.5	39.5	50.0	**41.5**
Transformer	49.5	47.0	50.0	**50.0**	50.0	46.5	49.5	38.5
LSTM	**55.5**	**54.0**	**53.5**	48.0	47.5	45.0	**51.5**	40.5

The LSTM employs two 50-unit layers, the first returning sequences, with dropout. The Transformer features two MultiHeadAttention layers, feed-forward networks, and global average pooling, with dropout. These run on 8× NVIDIA A40 GPUs with a batch size of 32, learning rate of 0.001, 50 epochs, and AdamW optimizer (Loshchilov and Hutter, 2019). LLMs—decoder-only (Gemma2 9B (Team et al., 2024b), Mixtral 8x7B (Jiang et al., 2024), LLaMA 7B (Touvron et al., 2023)) and encoder-only (RoBERTa (Liu et al., 2021), DistilBERT (Sanh et al., 2020))—are fine-tuned via Hugging Face Transformers on the same GPUs, with a batch size of 8, learning rate of 1×10^{-5}, 20 epochs, and AdamW with weight decay 0.01. Given class imbalance, we report balanced accuracy (B.Acc.) and macro-F1 score.

4.2 Results and Analysis

4.2.1 Baseline Performance

Table 1 shows the performance of baseline models trained on numeric NC subset features (e.g., LED: Q1, Q5; Section 3.3) across four dimensions: LED, ASE, PSE, and AIVP. LSTM leads in three dimensions, with balanced accuracy (B.Acc.) of 55.5% (LED), 53.5% (ASE), and 51.5% (AIVP), and macro-F1 peaking at 54.0% (LED), leveraging its sequential modeling capability. Random Forest (RF) excels for PSE (53.5% B.Acc., 52.5% F1), surpassing LSTM through robust feature aggregation. SVM and Transformer achieve moderate results, with Transformer's best F1 at 50.0% (ASE), while 1D CNN consistently underperforms (e.g., 39.5% F1 for PSE). Across dimensions, baselines average 50.8% balanced accuracy and 46.9% macro-F1, struggling with sparsity and MNAR patterns. These models have demonstrated a tendency to deliver unreliable results, with a significant skew towards predicting outcomes predominantly in the positive class, which makes these models unreliable for these tasks.

4.2.2 LLM Performance with NC Subset Features

Table 2 presents the performance of fine-tuned LLMs using verbalized NC subset features (e.g., 2–3 features per dimension, Section 3.3) across all four dimensions: LED, ASE, PSE, and AIVP. RoBERTa consistently achieves the highest macro-F1 scores, ranging from 65.0% (LED) to 70.5% (ASE, AIVP), with balanced accuracy peaking at 69.0% (AIVP), surpassing the best baseline (LSTM, 54.0% F1 for LED) by 11%–17%. Encoder-only models outperform decoder-only counterparts, with DistilBERT close behind RoBERTa (e.g., 68.5% F1 for AIVP vs. 70.5%), while decoder-only models show variability: Llama excels for PSE (73.0% F1) but drops to 56.5% for LED, and Mixtral lags across dimensions (55.5%–63.0% F1). Gemma2 performs well for ASE (70.0% F1) but averages lower elsewhere. The mean balanced accuracy (64.2%) and macro-F1 (64.4%) of LLMs highlight their textual reasoning advantage over numeric baselines, supporting their baseline superiority.

4.2.3 LLM Performance with NC Subset + Background Features

Table 3 reports LLM performance when NC subset features are augmented with background data (e.g., demographics). RoBERTa again dominates, with balanced accuracy improving to 72.5%–77.5% and macro-F1 to 73.0%–77.5% across dimensions, a 3%–12% gain over NC-only results (e.g., LED F1: 65.0% to 77.5%). This boost peaks for LED (77.5% F1), affirming background data's contextual value. DistilBERT follows closely, with notable gains (e.g., LED F1: 64.5% to 75.0%), while decoder-only models improve but remain inconsistent: Llama reaches 74.5% F1 for LED but dips to 66.5% for PSE, Gemma2 holds steady (e.g., 70.0% F1 for AIVP), and Mixtral trails (61.0%–66.0% F1). The mean balanced accuracy rises to 69.0% and macro-F1 to 69.5%, with encoder-only models (RoBERTa: 74.5% mean F1, DistilBERT: 68.3%) outperforming decoder-only

Table 2: LLM Performance Across Dimensions Using NC Subset Features Only

Model	**LED**		**ASE**		**PSE**		**AIVP**	
	B.Acc.	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**
Gemma2 9B	62.0	61.0	**72.0**	70.0	65.5	66.5	65.5	66.5
Mixtral 8x7B	62.0	61.5	63.5	63.0	55.5	55.5	59.0	59.0
Llama 7B	59.5	56.5	62.0	61.5	**73.0**	**73.0**	59.5	59.0
DistilBERT	65.0	64.0	63.5	67.0	67.5	67.0	67.5	68.5
RoBERTa	**65.0**	**65.0**	66.5	**70.5**	68.0	69.5	**69.0**	**70.5**

Table 3: LLM Performance Across Dimensions Using NC Subset and Background Features

Model	**LED**		**ASE**		**PSE**		**AIVP**	
	B.Acc.	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**
Gemma2 9B	71.5	72.0	69.0	69.0	65.5	64.5	69.0	70.0
Mixtral 8x7B	60.0	61.5	66.0	66.0	61.5	61.5	61.0	61.0
Llama 7B	72.5	74.5	68.0	69.0	66.5	66.5	66.5	66.5
DistilBERT	74.5	75.0	65.5	66.0	70.5	68.0	65.0	64.5
RoBERTa	**77.5**	**77.5**	**73.5**	**73.0**	**74.0**	**73.5**	**72.5**	**74.0**

Table 4: LLM Performance Across Dimensions Using All NC and Background Features

Model	**LED**		**ASE**		**PSE**		**AIVP**	
	B.Acc.	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**
Gemma2 9B	58.50	52.50	62.50	65.50	59.50	58.50	66.00	65.50
Mixtral 8x7B	55.50	56.50	59.00	63.00	58.00	56.50	52.00	53.50
Llama 7B	59.50	56.50	62.50	62.50	59.00	59.00	60.00	63.50
DistilBERT	62.00	62.50	64.00	63.50	60.50	62.50	65.50	64.00
RoBERTa	**66.50**	**65.00**	**64.50**	**65.00**	**63.00**	**62.50**	**64.50**	**64.00**

(Llama: 69.1%, Gemma2: 67.9%, Mixtral: 62.5%) by 5%–12%. Compared to baselines (max 54.0% F1), NC+background LLMs extend the gap to 19%–23%.

4.3 Ablation Study

We conduct two ablation studies to evaluate our LLM-based approach for forecasting student engagement levels across four dimensions.

Evaluating Feature Quantity: Subset vs. All NC Features. We compare LLMs fine-tuned on a zero-shot selected subset of NC features plus background features (Table 3) against those using all 10 NC features plus background features (Table 4). In the all-features case, RoBERTa achieves macro-F1 scores of 62.5%–65.0% and balanced accuracy (B.Acc.) of 63.0%–66.5%, markedly lower than the subset case's 73.0%–77.5% F1 and 72.5%–77.5% B.Acc. Dimension-specific F1 losses range from 8.0% (ASE) to 12.5% (LED), indicating that all 10 NC features introduce noise, weakening the signal distilled by expert LLM selection (Section 3.3). Encoder-only models (RoBERTa, DistilBERT) consistently outperform decoder-only variants (Gemma2 9B, Mixtral 8x7B, LLaMA 7B) across both configurations, though the gap narrows with all features—e.g., RoBERTa's LED F1 lead over LLaMA 7B shrinks from 18.0% (subset) to 8.5%—suggesting noise impacts decoder-only models less severely.

Assessing Input Modality: Textual LLMs vs. Numeric Baselines with All Features. We train baseline models—Random Forest (RF, 100 trees (Breiman, 2001)), Support Vector Machine (SVM), 1D CNN, Transformer, and LSTM—on numeric LE data with all 10 NC features (converted to scores, forming 960 × 36-D vectors, Section 3.4) and fine-tune RoBERTa (Liu et al., 2021), our top performer with subset features, on textual all NC features (verbalized responses). Table 5 reports results across 960 trajectories for four dimensions: LED, ASE, PSE, and AIVP. RoBERTa consistently outperforms numeric baselines in balanced accuracy and macro-F1 across most dimensions, leveraging textual reasoning to capture qualitative nuances and MNAR-impaired patterns that numeric models struggle to model. Notably, 1D CNN excels for ASE, suggesting some sequential patterns in numeric data align with convolutional strengths, yet RoBERTa's broader superiority—particularly for LED, PSE, and AIVP—underscores LLMs' advantage in processing raw verbalized sequences. Baselines like SVM and LSTM exhibit variability, often skewed by noise or positive-class bias, while Transformer and RF show moderate consistency but lack the discriminative power of textual LLMs. This complements the first ablation study (subset vs. all NC features), affirming that while subset selection enhances performance, even with all features, LLMs' textual modality outstrips numeric approaches for sparse, qualitative LE data.

Table 5: Performance of Numeric Baselines and Textual RoBERTa with All NC Features Across Dimensions

Model	**LED**		**ASE**		**PSE**		**AIVP**	
	B.Acc.	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**	**B.Acc.**	**F1**
Random Forest	52.5	49.5	50.5	45.5	49.0	46.5	48.0	41.0
SVM	50.5	45.5	50.0	40.5	44.5	40.0	50.0	38.0
1D CNN	52.0	51.5	**62.0**	**61.5**	48.0	47.5	44.5	43.0
Transformer	48.5	47.5	45.5	45.0	51.5	51.5	53.5	53.0
LSTM	44.0	43.5	47.5	47.5	47.0	47.0	50.0	47.5
RoBERTa	**61.0**	**60.5**	59.0	65.0	**64.0**	**62.5**	**62.5**	**62.0**

5 Conclusion

Our findings provide compelling evidence for the efficacy of our three-tier LLM framework in forecasting student engagement from qualitative LE data. We highlight three key insights. First, LLMs consistently outperformed traditional numeric baselines across all engagement dimensions, even when both were trained on the same selected non-cognitive feature subsets. This superiority underscores the inherent capability of LLMs to process and understand the nuanced information present in verbalized student responses, effectively capturing contextual patterns missed by numeric conversions and sequential models, particularly in the presence of MNAR missingness and data sparsity. Notably, this advantage persisted even when all available non-cognitive features were used, further emphasizing the limitations of traditional machine learning approaches for this type of qualitative time-series data.

Second, our analysis revealed a significant performance difference between LLM architectures. Encoder-only models, such as RoBERTa and DistilBERT, demonstrated a clear advantage over decoder-only models across various configurations. This suggests that their strength in creating robust representations from sparse textual sequences makes them particularly well-suited for the binary classification task of engagement forecasting. While decoder-only models showed occasional strong performance on specific dimensions, their overall variability indicates that their generative focus might be less optimal for the discriminative demands of this task. The consistent outperformance of encoder-only models, even with increased data complexity, highlights their robustness for this application.

Third, integrating background data significantly boosted LLM performance, particularly for specific engagement dimensions, emphasizing the importance of context. Furthermore, the synergy between our LLM-driven feature selection and forecasting tiers was validated by the enhanced performance achieved with selected feature subsets.

In conclusion, this work demonstrates the transformative potential of our three-tier LLM framework for analyzing complex, qualitative LE data in educational settings. By effectively addressing challenges such as MNAR missingness and noisy feature spaces, our approach offers a significant advancement over traditional numeric methods, paving the way for richer and more insightful analyses of student engagement and potentially other subjective, time-series datasets. **However, responsible deployment of such frameworks requires careful consideration of their limitations and ethical implications.**

6 Limitations

Our study acknowledges several important limitations. **Dataset scale and diversity**: Our analysis is based on data from 96 first-year STEM students at a single U.S. university, resulting in 960 trajectories. This relatively small and homogeneous sample limits generalizability to broader student populations, diverse educational contexts, or different demographic groups. **Validation constraints**: Our LLM-informed imputation method has not undergone human validation to verify the accuracy of generated missing value descriptors, affecting confidence in semantic quality and downstream forecasting performance. **Baseline limitations**: Our evaluation focuses on traditional machine learning and basic deep learning models, but does not benchmark against state-of-the-art multimodal or recent transformer-based time-series forecasting models. **Theoretical justification**: While empirical results demonstrate encoder-only LLMs' superior performance over decoder-only models, we provide limited theoretical explanation for this architectural advantage. **Dependency on proprietary models**: Our framework relies on closed-source LLMs (GPT-4o, Gemini, DeepSeek, Copilot), which may limit reproducibility and accessibility.

7 Ethical Considerations

7.1 LLM Biases and Educational Harms

Foundation models encode systemic biases from pretraining data (Bommasani et al., 2021), which can be amplified when fine-tuned on small educational datasets. LLMs characterized as "stochastic parrots" (Bender et al., 2021) exhibit stereotypical biases across gender, race, profession, and religion (Nadeem et al., 2021; Gallegos et al., 2024), with documented religious bias analogizing "Muslim" to "terrorist" in 23% of cases (Abid et al., 2021). A comprehensive risk taxonomy identifies discrimination, hate speech, and human-computer interaction harms as primary concerns (Weidinger et al., 2022).

Our framework's LLM-informed imputation and feature selection may inadvertently reflect these biases, potentially misrepresenting underrepresented student voices or reinforcing stereotypical engagement assumptions. Algorithmic bias in education disproportionately affects students based on race/ethnicity, gender, nationality, socioeconomic status, and disability (Baker and Hawn, 2022). Automated engagement predictions risk reinforcing inequalities through biased classifications that systematically disadvantage certain groups, as foundation model defects are inherited downstream (Bommasani et al., 2021).

7.2 Potential Harms and Mitigation

Self-fulfilling prophecies: Predictions may influence educator expectations, creating scenarios where students labeled "disengaged" receive reduced support. Automated decision-making risks "reducing a human being to a percentage," undermining student dignity (Binns et al., 2018). **Student autonomy**: Engagement monitoring may create surveillance environments compromising authentic self-expression and altering social dynamics the technology purports to measure (Weidinger et al., 2022). **Resource allocation**: Binary predictions could lead to misallocation if false positives/negatives disproportionately affect vulnerable populations (Corbett-Davies et al., 2017).

Privacy considerations: Our dataset involves sensitive student information including academic performance and personal reflections. While committing to full anonymization, evolving LLM capabilities may create unforeseen privacy risks not understood at consent time.

Mitigation strategies: We propose safeguards informed by responsible AI principles (Weidinger et al., 2022; Bommasani et al., 2021): (1) Regular bias auditing across demographic subgroups; (2) Human-in-the-loop validation requiring educator oversight before interventions (Binns et al., 2018); (3) Transparent communication about data use; (4) Supportive-only intervention guidelines; (5) Continuous monitoring of deployment outcomes. Our framework should augment, not replace, human educational judgment, emphasizing fairness, accountability, and transparency in high-stakes educational applications (Binns et al., 2018; Weidinger et al., 2022; Baker and Hawn, 2022).

Acknowledgments

This research was supported by grants from the U.S. National Science Foundation (NSF DUE 2142558), the U.S. National Institutes of Health (NIH NIGMS P20GM130461 and NIH NIAAA R21AA029231), and the Rural Drug Addiction Research Center at the University of Nebraska-Lincoln.

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, page 298–306, New York, NY, USA. Association for Computing Machinery.

Ryan S. Baker and Andrew Hawn. 2022. Algorithmic bias in education. *International Journal of Artificial Intelligence in Education*, 32:1052–1092.

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

Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 2018. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14, New York, NY, USA. Association for Computing Machinery.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models. *ArXiv*.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2015. *Time Series Analysis: Forecasting and Control*, 5th edition. Wiley Series in Probability and Statistics. Wiley, Hoboken, NJ.

Leo Breiman. 2001. Random Forests. *Mach. Learn.*, 45(1):5?32.

Microsoft Copilot. 2024. Generated content. Online. Accessed on January 13, 2025.

Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. 2017. Algorithmic decision making and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, page 797–806, New York, NY, USA. Association for Computing Machinery.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jennifer Fredricks. 2014. *Eight Myths of Student Disengagement: Creating Classrooms of Deep Learning*. Corwin Press, Thousand Oaks, California.

Isabel O. Gallegos, Ryan A. Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K. Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 50(3):1097–1179.

Azul Garza, Cristian Challu, and Max Mergenthaler-Canseco. 2024. Timegpt-1. *Preprint*, arXiv:2310.03589.

Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3(null):1157–1182.

Ahatsham Hayat, Bilal Khan, and Mohammad Hasan. 2024a. Improving transfer learning for early forecasting of academic performance by contextualizing language models. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 137–148, Mexico City, Mexico. Association for Computational Linguistics.

Ahatsham Hayat, Bilal Khan, and Mohammad Rashedul Hasan. 2024b. Leveraging language models for analyzing longitudinal experiential data in education. *arXiv:2503.21617*.

M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28.

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

A.K. Jain, R.P.W. Duin, and Jianchang Mao. 2000. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y. Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, and Qingsong Wen. 2024. Time-LLM: Time series forecasting by reprogramming large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

David A. Kolb. 1984. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice-Hall, Englewood Cliffs, NJ.

Xiang Li, Xinning Zhu, Xiaoying Zhu, Yang Ji, and Xiaosheng Tang. 2020. Student Academic Performance Prediction Using Deep Multi-source Behavior Sequential Network. In *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 567–579, Cham. Springer International Publishing.

R. J. A. Little and D. B. Rubin. 2019. *Statistical Analysis with Missing Data*. John Wiley & Sons.

Xian Liu. 2016. Methods for handling missing data. In Xian Liu, editor, *Methods and Applications of Longitudinal Data Analysis*, chapter 14, pages 441–473. Academic Press.

Zhuang Liu, Wayne Lin, Ya Shi, and Jun Zhao. 2021. A robustly optimized bert pre-training approach with post-training. In *Chinese Computational Linguistics: 20th China National Conference, CCL 2021, Hohhot, China, August 13–15, 2021, Proceedings*, page 471–484, Berlin, Heidelberg. Springer-Verlag.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Mistral AI. 2024. Announcing mistral 7b instruct v0.3. `https://mistral.ai/news/announcing-mistral-7b/`. Accessed: 2025-05-20.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. 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 5356–5371, Online. Association for Computational Linguistics.

OpenAI. 2024. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-09-02.

Keiron O'Shea and Ryan Nash. 2015. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458.

M. Palmer, M. Larkin, R. de Visser, and G. Fadden. 2010. Developing an interpretative phenomenological approach to focus group data. *Qualitative Research in Psychology*, 7(2):99–121.

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

D. B. Rubin. 1976. Inference and missing data. *Biometrika*, 63:581–592.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

J. L. Schafer. 1997. *Analysis of Incomplete Multivariate Data*. Chapman & Hall/CRC, London, UK.

Gale M. Sinatra, Benjamin C. Heddy, and Doug Lombardi. 2015. The challenges of defining and measuring student engagement in science. *Educational Psychologist*, 50(1):1–13.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, and et al. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak

Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024b. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*. ArXiv:2302.13971 [cs].

S. van Buuren and K. Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in r. *Journal of Statistical Software*, 45(3):1–67.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. 2014. StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 3–14, New York, NY, USA. Association for Computing Machinery.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, Courtney Biles, Sasha Brown, Zac Kenton, Will Hawkins, Tom Stepleton, Abeba Birhane, Lisa Anne Hendricks, Laura Rimell, William Isaac, Julia Haas, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 214–229, New York, NY, USA. Association for Computing Machinery.

Weiqi Xu and Fan Ouyang. 2022. The application of AI technologies in STEM education: a systematic review from 2011 to 2021. *International Journal of STEM Education*, 9(1):59.

Xuhai Xu, Xin Liu, Han Zhang, Weichen Wang, Subigya Nepal, Yasaman Sefidgar, Woosuk Seo, Kevin S. Kuehn, Jeremy F. Huckins, Margaret E. Morris, Paula S. Nurius, Eve A. Riskin, Shwetak Patel, Tim Althoff, Andrew Campbell, Anind K. Dey, and Jennifer Mankoff. 2023. GLOBEM: Cross-Dataset Generalization of Longitudinal Human Behavior Modeling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(4):190:1–190:34.

Xuhai Xu, Han Zhang, Yasaman Sefidgar, Yiyi Ren, Xin Liu, Woosuk Seo, Jennifer Brown, Kevin Kuehn, Mike Merrill, Paula Nurius, Shwetak Patel, Tim Althoff, Margaret Morris, Eve Riskin, Jennifer Mankoff, and Anind Dey. 2022. Globem dataset: Multi-year datasets for longitudinal human behavior modeling generalization. In *Advances in Neural Information Processing Systems*, volume 35, pages 24655–24692. Curran Associates, Inc.

Hao Xue and Flora D. Salim. 2024. Promptcast: A new prompt-based learning paradigm for time series forecasting. *IEEE Trans. on Knowl. and Data Eng.*, 36(11):6851–6864.

Wangyang Ying, Dongjie Wang, Haifeng Chen, and Yanjie Fu. 2024. Feature selection as deep sequential generative learning. *Preprint*, arXiv:2403.03838.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2018. Gain: Missing data imputation using generative adversarial nets. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 5689–5698. PMLR.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2025. A survey of large language models. *Preprint*, arXiv:2303.18223.

Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One fits all: power general time series analysis by pretrained lm. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Bridging the Socioeconomic Gap in Education: A Hybrid AI and Human Annotation Approach

Nahed Abdelgaber[1]*, Labiba Jahan[1], Arham Vinit Doshi[2], Rishi Suri[3],
Hamza Reza Pavel[4], Jia Zhang[1]
[1]Southern Methodist University, Dallas, TX 75205
[2]University of Texas at Arlington, Arlington, TX 76019
[3]University of Texas at Austin, Austin, TX 78712
[4]University of Maryland, College Park, MD 20742

Abstract

Students' academic performance is influenced by various demographic factors, with socioeconomic class being a prominently researched and debated factor. Computer Science research traditionally prioritizes computationally definable problems, yet challenges such as the scarcity of high-quality labeled data and ethical concerns surrounding the mining of personal information can pose barriers to exploring topics like the impact of socioeconomic status (SES) on students' education. Overcoming these barriers may involve automating the collection and annotation of high-quality language data from diverse social groups through human collaboration. Therefore, our focus is on gathering unstructured narratives from Internet forums written by students with low-SES using machine learning models and human insights. We developed a hybrid data collection model that semi-automatically retrieved narratives from the Reddit website and created a dataset five times larger than the seed dataset. Additionally, we compared the performance of traditional ML models with recent large language models (LLMs) in classifying narratives written by low-SES students, and analyzed the collected data to extract valuable insights into the socioeconomic challenges these students encounter and the solutions they pursue.

1 Introduction

Low socioeconomic status (SES) refers to a disadvantaged position in society determined by factors such as income, education, and occupation. Individuals with low SES typically have limited financial resources (Scott-Clayton, 2015), lower educational attainment (Titus, 2006), and reduced access to quality healthcare and academic opportunities (Adler and Newman, 2017). These disparities profoundly impact students' educational experiences, shaping their academic performance, career prospects, and long-term well-being. Students from low socioeconomic backgrounds often struggle with financial barriers, making it difficult to afford tuition and educational resources (Brown and Carr, 2013). They also have limited access to academic support and technology, which can impede their academic success. Additionally, these students may experience social isolation and psychological stress due to the pressure of competing with peers from more privileged backgrounds (Lee et al., 2008).

Research on students from low socioeconomic backgrounds is crucial for identifying and addressing the unique challenges they face in education. Understanding these struggles can inform policies and interventions that promote equity, ensuring that students receive the necessary support to succeed. Despite this importance, NLP research has largely overlooked socioeconomic status. A survey by (Cercas Curry et al., 2024) found only 20 papers in the ACL Anthology explicitly mentioning SES, highlighting a substantial gap in computational research. This lack of attention limits our understanding of how SES affects student life and contributes to the development of educational technologies that may not adequately address the needs of low-SES students, potentially widening the digital divide (Kelbessa et al., 2024).

A major challenge in computational research on low-SES students is the scarcity of high-quality labeled data. Most existing datasets rely on structured survey responses, which fail to fully capture the complexity of students' experiences. To address this, we analyze a dataset published by (Kelbessa et al., 2024), containing 74 narratives written by low-SES students on Reddit. These narratives offer valuable firsthand insights into the struggles and coping mechanisms of low-SES students, making them an important resource for NLP research. However, as the dataset was annotated by only two individuals and lacks gold-standard validation, it

---

*Major contributor. Contact: `nabdelgaber@smu.edu`

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 348–364
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

presents both an opportunity and a challenge for refinement and further analysis.

The remainder of this paper is structured as follows: In §2, we review previous research on SES and NLP. In §3, we describe our dataset, including its source and key attributes. In §4, we present our data evaluation and analysis methodology, which involves assessing data quality, filtering high-quality narratives, and applying linguistic metrics, sentiment analysis, and topic modeling. Finally, we summarize our findings (§5), discuss our key contributions (§6), explore the ethical and societal implications of our research (§7).

2 Related Work

Socioeconomic status (SES) influences various aspects of life, including education, health, and social mobility. Understanding SES disparities requires high-quality datasets that capture linguistic, behavioral, and demographic patterns. However, acquiring such datasets poses significant challenges, including ethical concerns, accessibility restrictions, and issues of representativeness.

Several datasets have been developed for SES research, particularly leveraging online sources and social media platforms. Twitter has been widely used due to its accessibility and large user base. For instance, (Lampos et al., 2014) used UK Twitter data to analyze how user-generated content predicts SES. Similarly, (Mentink, 2016) collected a dataset of 2.5 million Dutch Twitter users to infer their socioeconomic backgrounds. While Twitter data provides valuable large-scale insights, its brevity and informal nature limit its usefulness for in-depth socioeconomic analysis.

Beyond social media, narrative-based data provides rich contextual insights into SES struggles. (Kelbessa et al., 2024) compiled a dataset of 74 SES-related narratives as a foundation for further research. Unlike social media posts, these narratives offer longer, structured reflections on lived experiences, making them more informative for understanding the personal and systemic challenges faced by individuals from low-SES backgrounds.

Despite advancements in SES-related data collection, several challenges persist. Ethical concerns arise when gathering data from social media, as it raises privacy and consent issues (Stieglitz et al., 2018). Data accessibility is another major barrier, as many relevant datasets are either proprietary or require restrictive permissions. Additionally, existing SES datasets often overrepresent specific demographics, reducing their generalizability and limiting their applicability across diverse populations.

In this work, we aim to address these challenges by expanding SES-related narrative datasets with a focus on depth—the inclusion of rich, first-person accounts that describe the lived experiences, struggles, and coping strategies of low-SES individuals—and representativeness, meaning coverage of diverse challenges within the SES spectrum, such as financial hardship, social marginalization, and academic barriers. Unlike short-form social media posts, our dataset consists of longer, structured narratives that offer greater contextual detail. This work contributes to the growing body of research on SES in computational linguistics and provides a valuable resource for future studies examining socioeconomic barriers in education.

3 Data

Unlabeled Reddit Posts: This data is publicly available and was collected on 2019 for the ThinkPlayHack event hosted in July 2019 in Taos for Dr. Jo Guldi (Southern Methodist University (SMU), 2019). It contains over 1 terabyte of Reddit posts published from 2005 to 2013. To manage the extensive data, measured in terabytes, we initially selected posts exclusively from subreddits associated with low-SES, such as 'college', 'ApplyingToCollege', 'depression', 'askReddit', 'broke', 'financialaid', and 'fafsa'. After filtering for relevant subreddits and eliminating duplicates, the resulting dataset comprised 799,032 Reddit posts (total 106859972 words) with 7 average sentences and 134 average words per post.

Labeled Reddit Posts: (Kelbessa et al., 2024) gathered 74 low-SES narratives from Reddit. To ensure the validity of these narratives as data points, the following criteria were applied: 1) The narratives needed to shed light on the experience of being a low-SES student and attending higher education, focusing on financial, psychological, physical, or social struggles. 2) The narratives should describe the challenges faced by individuals with low-SES backgrounds, their efforts to improve their situation, and the outcomes of those efforts. 3) Narratives that primarily offered general commentary, described a condition, or provided advice were excluded, as they did not qualify as valid data points. Each narrative had to meet at least one of the first two criteria and also satisfy the third qualification to be included in the dataset. The narratives were doubly annotated by two annotators to ensure the consistency and quality of the data. To ensure the quality of this publicly available data (Kelbessa et al., 2024), we thoroughly applied the above criteria on the 74 narratives and identified 64 of them met the criteria (background of low-SES).

We note that this dataset is framed around college accessibility and affordability—an issue particularly salient within the context of the U.S. educational system. While such focus is not inherently limiting, we acknowledge that the effects and expressions of low socioeconomic status (SES) can vary significantly across different countries and educational models. As such, our framing may not fully capture the global diversity of SES-related challenges. We aim to address this limitation in future work by broadening the scope beyond U.S.-centric educational themes and exploring SES narratives in other domains such as employment, housing, or healthcare access. Additionally, the criteria used for selecting valid narratives could be adapted to include financial hardship contexts not strictly tied to higher education.

To process 64 non-low-SES data points, we applied semantic textual similarity to determine which posts in the unlabeled Reddit Posts (mentioned above) had the weakest correlation with the ground truth 64 low-SES narratives. We explored several similarity measures, such as the Negative Euclidean Distance, Negative Manhattan Distance, and Cosine Similarity, using the top-ranked Sentence Transformer model from the Massive Text Embedding Benchmark (Muennighoff et al., 2022) leaderboard on Hugging Face. We identified the 64 posts and manually evaluated them that had the lowest similarity to the ground truth data points, treating these as the non-low-SES ground truth. An example for illustration is provided in the appendix A.1. The final dataset comprised 64 non-low-SES Reddit posts with 15 average sentences and 299 average words per post.

4 Empirical Study

We started with the labeled dataset of 128 Reddit posts, consisting of 64 narratives from low-SES students and 64 from non-low-SES students. Over the course of three iterations, we curated and expanded this dataset, ultimately building a larger collection of narratives. Our primary goal is to collect enough data for future training and automation of the model. In each iteration, we added narratives in the training data that are newly labeled in the previous iteration as low-SES and then try to label the remaining unlabeled narratives. Every iteration followed a three-step process: first, we applied traditional binary ML classifiers and LLMs to categorize the unlabeled Reddit posts as either low-SES or non-low-SES; second, we used clustering techniques to identify and remove outliers from the dataset. Third, we manually annotated the narratives to perform the final evaluation. The numbers of the resulted labeled narratives at each step are shown at Table 2.

4.1 Step 1: Classification

We evaluated the performance of 22 traditional ML models and LLMs to distinguish between low-SES and non-low-SES texts. All the models we used are from scikit-learn (Pedregosa et al., 2011) and Hugging Face Transformers (Wolf et al., 2020). The evaluation followed a 5-fold cross-validation approach, with 70% of the dataset allocated for training, 15% for validation, and 15% for reporting the results. We conducted our experiments over three iterations, each with increasing complexity. In the first iteration, the dataset consisted of 64 low-SES narratives and 64 non-low-SES narratives, which were easily separable. This was due to our careful selection of the non-low-SES narratives, ensuring they were clearly distinguishable from the low-SES narratives. By the second iteration, the dataset had nearly doubled in size, and the classification task became more challenging. This time, the non-low-SES narratives were selected from the false positives of the first iteration, resulting in less clear separation between the classes. In the third iteration, the challenge intensified further as the dataset again doubled, with non-low-SES narratives chosen from the false positives of the second iteration. Consequently, the classes were significantly harder to separate, reflecting the increasing difficulty of the classification task. To address these increasingly complex classification tasks, we utilized a variety of fine-tuned pre-trained language models, models with few-shot capabilities, and traditional ML models. The overall results in three different iterations are shown at Table 1 and detailed parameters for all models are provided in Table 3 in Section A.2.

The Traditional models included Random Forest

Model	1st iteration			2nd iteration			3rd iteration			Model	1st iteration			2nd iteration			3rd iteration		
	P	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1		**P**	**R**	F_1	**P**	**R**	F_1	**P**	**R**	F_1
RF	0.55	0.55	0.54	0.74	0.74	0.74	0.61	0.61	0.61	BERT	0.71	0.70	0.70	0.71	0.69	0.68	0.56	0.56	0.56
MNB	0.81	0.70	0.67	0.79	0.76	0.75	0.65	0.63	0.62	OPT-13B few	0.50	0.50	0.40	0.52	0.52	0.52	0.51	0.50	0.49
SVM	0.60	0.60	0.60	**0.87**	**0.87**	**0.87**	0.60	0.60	0.60	Claudia few	0.60	0.55	0.49	0.49	0.50	0.47	0.50	0.50	0.46
LR	0.60	0.60	0.60	**0.87**	**0.87**	**0.87**	0.64	0.64	0.64	LLamA-7B few	0.34	0.35	0.34	0.69	0.64	0.66	0.47	0.48	0.46
XGBoost	0.66	0.65	0.64	0.74	0.74	0.74	0.55	0.54	0.54	LLaMA-1.3B few	0.50	0.50	0.45	0.38	0.41	0.38	0.55	0.54	0.53
RoBERTa	0.75	0.75	0.75	0.76	0.76	0.76	0.73	0.67	0.65	LoRA DistillGPT	0.25	0.50	0.33	0.27	0.52	0.35	0.25	0.50	0.33
RoBERTa-large	0.75	0.75	0.75	0.74	0.74	0.74	**0.72**	**0.72**	**0.72**	LoRA LLaMA-1.3B	0.25	0.50	0.33	0.30	0.41	0.32	0.25	0.50	0.33
DeBERTa	0.77	0.75	0.74	0.74	0.74	0.74	0.56	0.55	0.55	LoRA GPT-2	0.25	0.50	0.33	0.27	0.52	0.35	0.25	0.50	0.33
ELECTRA	0.77	0.75	0.74	0.83	0.80	0.79	0.59	0.56	0.53	bart-large-mnli Zero	**0.80**	**0.80**	**0.80**	0.22	0.43	0.29	0.46	0.46	0.46
ALBERT	0.66	0.65	0.64	0.65	0.65	0.65	0.55	0.55	0.55	DistilBert	0.75	0.75	0.75	0.77	0.74	0.74	0.63	0.62	0.62
XLNet	0.60	0.55	0.49	0.63	0.61	0.60	0.66	0.65	0.65	T5	0.50	0.50	0.48	0.75	0.72	0.72	0.55	0.55	0.55

Table 1: Comparison of average Precision (P), Recall (R), and F_1 scores for both classes across three iterations for various classification models.

Iteration No.	Unlabeled Texts	Labeled (Classifier)	Labeled (Cluster)	Labeled (Human)
Iteration 1	799,032	13,635	289	110
Iteration 2	798,743	390	381	167
Iteration 3	798,362	5195	444	121

Table 2: Summary of the annotated narratives at three different steps- Classification, Clustering, and Human Annotation.

(RF), Multinomial Naive Bayes (MNB), Support Vector Machine (SVM), Logistic Regression (LR), and Gradient Boosting (XGBoost). Overall, MNB, SVM, and LR demonstrated effective performance across the iterations, while RF and XGBoost struggled, particularly in the more challenging contexts. The fine-tuned pre-trained language models included Robustly Optimized BERT (RoBERTa), Robustly Optimized BERT-Large (RoBERTa-Large), Decoding-Enhanced BERT (DeBERTa), Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA), A Lite BERT (ALBERT), eXtreme Language Model (XLNet), Text-to-Text Transfer Transformer (T5), Distilled BERT (DistilBERT), and Bidirectional Encoder Representations from Transformers (BERT). Overall, RoBERTa, RoBERTa-Large, and ELECTRA demonstrated strong and consistent performance across iterations, while ALBERT and XLNet had difficulty handling the increasing complexity of the task. The few-shot models included Open Pre-trained Transformer (OPT-13B), Claudia, LLM Meta AI (LLaMA-7B), and LLM Meta AI (LLaMA-1.3B). Few-shot models were not as effective in distinguishing between increasingly similar narratives as the dataset complexity grew. The LoRA fine-tuning models included Distilled Generative Pre-trained Transformer (DistillGPT), LLM Meta AI (LLaMA-7B), and Generative Pre-trained Transformer (GPT-2). The results suggest that the LoRA fine-tuning models were not well-suited for the increasing complexity of the classification task, possibly due to their limited adaptation to more challenging data. The zero-shot model used was bart-large-mnli Zero (Yin et al., 2019). The sharp decline in performance in the second and third iterations suggests that the zero-shot model struggled to handle the increasing similarity between the low-SES and non-low-SES texts, as it lacked the fine-tuning capabilities of other models.

In the first iteration, the BART-large-mnli zero-shot model showed superior performance in terms of balanced metrics. We applied a confidence threshold of 0.7, meaning a text was classified as low-SES only if the model predicted it with confidence greater than 0.7. This threshold was empirically selected after testing a range of values above and below it; 0.7 offered the best balance between high-confidence predictions and getting a manageable number of samples. As a result, this model filtered the data, yielding 13,635 low-SES texts out of 799,032 unlabelled texts. In the second iteration, we employed an ensemble approach using SVM and LR, the two models that performed best in the first phase. Both models were trained separately and combined to improve classification accuracy. The SVM provided decision scores, and LR produced class probabilities. We again applied a 0.7 threshold to both models, ensuring that a text was classified as low-SES only if both classifiers agreed with high confidence. This approach yielded 390 low-SES texts.

Traditional ML Models Versus LLMs: In iteration 1, traditional models, such as SVM and LR, achieved balanced performance, with macro-averaged F_1 scores of 0.60. These models demonstrated robust precision and recall across both classes, although they did not outperform more advanced models. XGBoost performed slightly better, with an F_1 score of 0.66, particularly excelling in classifying non-low-SES texts.

Among LLMs, RoBERTa achieved an F_1 score of 0.75, showcasing strong performance with a balanced precision and recall across both SES classes. Similarly, DeBERTa and ELECTRA performed well, both achieving F_1 scores of 0.74. These models demonstrated better balance than some traditional models in classifying both low-SES and non-low-SES texts. However, other LLMs, such as ALBERT and XLNet, underperformed compared to their counterparts, with F_1 scores of 0.64 and 0.60, respectively. XLNet particularly struggled with the low-SES class, achieving an F_1 score of 0.31, highlighting its difficulty in accurately identifying low-SES narratives. Interestingly, zero-shot models like bart-large-mnli demonstrated strong performance, with an F_1 score of 0.80, matching the best-performing models in this iteration. In contrast, few-shot models, such as LoRA Fine-Tuned LLaMA and DistilGPT, performed poorly, indicating that few-shot learning in this context was less effective than fine-tuning. Detailed results are shown in section A.3 at Table 4.

In iteration 2, the comparison between traditional ML and LLMs highlights the distinct strengths and weaknesses of each group. Traditional models, such as SVM and LR, outperformed most LLMs, achieving the highest F_1 scores of 0.87 across both low-SES and non-low-SES categories. These models demonstrated strong precision and recall, particularly in the non-low-SES class, underscoring their robustness in effectively balancing both classes. For example, SVM achieved a precision of 0.88 and recall of 0.85 for low-SES, while LR displayed a precision of 0.91 for low-SES and a recall of 0.93 for non-low-SES, making them reliable in these classification tasks.

In contrast, fine-tuned LLMs, such as RoBERTa and ELECTRA, also performed well but fell slightly behind the top traditional models. RoBERTa achieved a weighted F_1 score of 0.76, showing a strong balance between precision (0.74) and recall (0.82) for the non-low-SES class. However, its recall for low-SES (0.69) was lower compared to traditional models, meaning it missed more low-SES texts. ELECTRA achieved a higher F_1 score of 0.79 and performed exceptionally well in identifying non-low-SES texts, with a precision of 0.73 and recall of 0.96. Nevertheless, ELECTRA struggled with low-SES classification, showing a precision of 0.94 but a much lower recall of 0.62, indicating it missed more low-SES examples. DeBERTa displayed performance similar to traditional models like RF and XGBoost, with an F_1 score of 0.74. Its precision and recall were balanced across both classes but did not achieve the standout performance of models like RoBERTa or ELECTRA. Other LLMs, including ALBERT and XLNet, significantly underperformed compared to both traditional models and other LLMs, with F_1 scores of 0.65 and 0.60, respectively. XLNet particularly struggled with the non-low-SES class, achieving a recall of only 0.46, indicating significant difficulty in identifying non-low-SES texts accurately. ALBERT exhibited more balanced but lower performance across both classes.

These results for iteration 2 indicate that while LLMs have shown potential—particularly models like RoBERTa and ELECTRA—traditional models such as SVM and LR remain more reliable for tasks involving both low-SES and non-low-SES classification. Their superior balance between precision and recall across both categories demonstrates their robustness, whereas LLMs, though effective in certain areas such as precision for non-low-SES, may require further fine-tuning to achieve the same comprehensive balance seen in traditional models. Detailed results are shown in section A.3 at Table 5.

A further evaluation of traditional ML models and LLMs reveals interesting trends in model performance as the dataset complexity increases. While traditional models like SVM and LR continued to show stability, their dominance observed in earlier iterations has now been matched or exceeded by fine-tuned LLMs in certain aspects. Among the traditional models, LR demonstrated consistency, achieving an average 0.64 F_1 across both SES classes. While it excelled in the non-low-SES class, with a precision of 0.84 and recall of 0.93, it showed less robustness in the low-SES class, with precision and recall hovering around 0.64. Similarly, RF maintained stable performance, with an F_1 score of 0.61, although it underperformed compared to LR, particularly in the non-low-SES class (precision: 0.63, recall: 0.53).

Fine-tuned LLMs displayed notable improvements. RoBERTa-large emerged as one of the top performers, achieving the highest average F_1 score of 0.72 across both classes, surpassing traditional models like SVM and LR. This model exhibited a well-balanced performance with precision, recall, and F_1 scores closely aligned (precision: 0.70, recall: 0.76 for non-low-SES; precision: 0.74, recall: 0.67

for low-SES), indicating its capability to handle both classes. DeBERTa also showcased a solid performance, with an average F_1 of 0.55, though it struggled with the low-SES class (F_1: 0.49) compared to RoBERTa-large. Similarly, ELECTRA achieved an average F_1 of 0.53 but encountered challenges in classifying low-SES examples, where recall dropped to 0.31. These results suggest that while LLMs like RoBERTa-large are emerging as strong contenders, certain models such as ELECTRA and DeBERTa still require fine-tuning to handle the low-SES class. Few-shot models, such as Claudia and LLaMA-1.3B, presented mixed results. Claudia, in its few-shot configuration, achieved an average F_1 of 0.46, highlighting difficulties in identifying low-SES narratives (F_1: 0.32). On the other hand, LLaMA-1.3B fared slightly better, with an average F_1 score of 0.53, performing consistently across both SES classes. However, neither of these models surpassed fine-tuned LLMs or traditional models in overall performance. Fine-tuned models with LoRA (Low-Rank Adaptation), such as LoRA GPT-2 and LoRA LLaMA-1.3B, delivered uneven outcomes. Both models demonstrated high precision for the non-low-SES class (0.50) but struggled significantly with the low-SES class, where they failed to capture any true positive examples (recall: 0.00, F_1: 0.00).

Iteration 3 reveals a growing strength of fine-tuned LLMs, particularly RoBERTa-large, which outperforms traditional models. This model's balanced precision and recall across SES classes emphasize its versatility. Models like DeBERTa and ELECTRA show that while LLMs are improving, they can still struggle with the low-SES class. Few-shot models and LoRA fine-tuned models exhibited less consistent results, often failing to achieve the comprehensive balance required for SES classification. This highlights the continued importance of fully fine-tuning LLMs for this task, as parameter-efficient models and few-shot learning may not yet match the robustness of more thoroughly fine-tuned counterparts like RoBERTa-large. Detailed results are shown in section A.3 at Table 6.

4.2 Step 2: Clustering

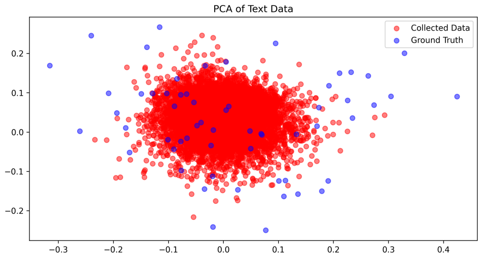
Once the classification step was completed, we used clustering to group similar texts and remove outliers. In the first iteration, we analyzed 13,635 posts identified as low-SES by the classifier. To compare the similarity between the ground truth samples and the collected posts, we applied PCA (Jolliffe, 2002) for dimensionality reduction using a tf-idf vectorizer (Salton and Buckley, 1988) with a maximum of 1,000 features, visualizing the clusters in 2D space. Some outliers were detected in the ground truth data. These outliers were removed by applying the interquartile range (IQR) method, reducing the dataset to 55 points (Fig. 1a).

Next, we computed cosine similarity between the normalized vectors of the collected data and the outlier-free ground truth data. A threshold of 0.9 was set to classify data points as similar or dissimilar. This process revealed that 289 from the collected data points met or exceeded the total similarity score(where the summation of the classified data is similar to ground truth data points) of 35, 940 data points had a similarity score of at least 34, and 10,821 data points had a score of 33 or less.

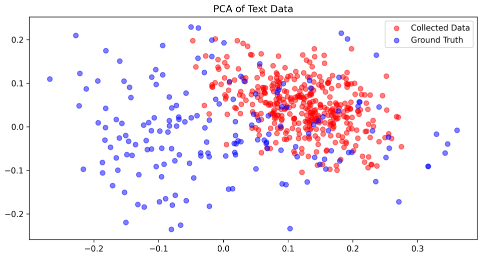
In the second iteration, after applying PCA to the collected data and the ground truth data, visualizing the results in a 2D space (see Fig. 1b). The collected data points and ground truth data were plotted to observe clustering patterns, allowing us to assess the similarity between the two datasets. Outliers were removed using the Interquartile Range (IQR) method, where values outside 1.5 times the IQR from the first (Q1) and third quartiles (Q3) were identified and excluded from both datasets. This process reduced the total number of collected data points from 390 to 381.

In the Third iteration, we analyzed 5195 posts identified as low-SES. To compare the similarity between the ground truth samples and the collected posts, we applied PCA for dimensionality reduction using a tf-idf Vectorizer with a maximum of 1,000 features, visualizing the clusters in 2D space. Some outliers were detected in the ground truth data. These outliers were removed by applying the interquartile range (IQR) method, reducing the dataset to 331 points (Figure 1c). Next, we computed cosine similarity between the normalized vectors of the collected data and the outlier-free ground truth data. We used the same threshold of 0.9 to classify data points as similar or dissimilar. This process revealed that 121 from the collected data points met or exceeded the total similarity score of 86, 260 data points with similarity score of at least 85, and 444 data points had a score of 84 or less.

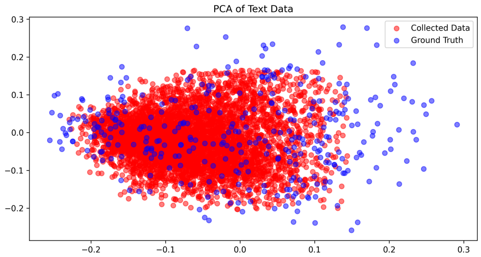
In iteration 1, the collected data predominantly covers the central portion of the ground truth spectrum,



(a) Iteration 1 Clustering Without Outliers



(b) Iteration 2 Clustering Without Outliers



(c) Iteration 3 Clustering Without Outliers

Figure 1: Clustering of Ground Truth and Collected Data Using PCA Across Three Iterations Without Outliers

indicating that the initial classification managed to capture a concentrated part of the low-SES class but left much of the outer spectrum of the ground truth unexplored. Moving to iteration 2, the collected data begins to diverge, covering less of the ground truth spectrum compared to Iteration 1. This suggests that the classification in this iteration was more selective but also less comprehensive in capturing the full range of the low-SES data. Finally, in iteration 3, we see a significant improvement, with the collected data covering over 70% of the ground truth spectrum. This indicates a better alignment between the collected and ground truth data, suggesting that the classification in this iteration successfully captured a much broader range of the ground truth low-SES examples, resulting in a more balanced and comprehensive dataset.

5 Data Annotation

Human Annotation: The first three authors followed the criteria described in Section 3 to annotate the 289 texts from the clustering step at iteration 1 and getting the data points with a similarity score of at least 35. This process resulted in 110 texts being annotated as low-SES out of 289. In the second iteration, we annotated 381 texts, of which 167 were classified as low-SES and in the third iteration, we annotated 121 texts that has a similarity score of at least 86, we got 46 low-SES Text.

In iteration 1, most of the texts annotated as not low-SES contained general advice and lacked the background indicative of being from a low-SES perspective. These texts often focused on providing broad recommendations rather than sharing personal experiences tied to financial difficulties. The absence of key indicators, such as struggles with income, debt, or reliance on social services, made it clear that these individuals did not face the same economic constraints typical of low-SES situations. As a result, their narratives were more aligned with middle or higher SES backgrounds, where financial stability was not a central concern. In iteration 2, as the task became more challenging, some of the texts annotated as not low-SES included background information and challenges but lacked personal experience that would validate them as low-SES. Upon further analysis, clear patterns emerged from the texts classified as low-SES. Many highlighted the need for financial aid, with individuals working multiple jobs or living in single-parent households with little to no income. Debt, particularly from educational loans or basic expenses, was a recurring theme, as was the lack of family support, indicating broader financial instability. These socio-economic markers—multiple jobs, debt, and minimal family support—are crucial for refining the model to better detect low-SES cases in future iterations. In iteration 3, after extracting 277 low-SES texts, some of the texts annotated as not low-SES described challenges and solutions but appeared to originate from individuals of medium SES rather than low-SES. We noticed that some texts annotated as not low-SES described challenges like juggling multiple jobs or balancing full-time work and studies. These individuals often discussed FAFSA loans or supporting a family due to a relative's disability. However, many of these cases appeared to stem from medium SES backgrounds, as they had access to basic loans or even an inheritance. This suggests that while these individuals faced financial difficulties, their situations were distinct from those typically associated with low-SES, highlighting the nuanced differences between SES classifications.

LLM Annotation: We used a pre-trained LLaMA model with 7 billion parameters (Touvron et al., 2023) to annotate the dataset of low-SES student narratives. The model was configured with a maximum of 200 tokens, a temperature of 0.5 for less randomness, and deterministic sampling. The goal was to extract information about students' background, struggles, and solutions related to financial, psychological, physical, or social challenges. Comprehensive descriptions of the prompts are provided in Section A.4. Although LLM was effective, the extraction of structured data was a challenge. The model occasionally produced extraneous text or improperly formatted outputs, requiring post-processing.

For visualization, We applied Principal Component Analysis (PCA) to reduce high-dimensional sentence embeddings on extracted background information from ground truth and collected datasets for visualization. K-means clustering was then used to group semantically similar sentences, identifying key thematic clusters. The resulting scatter plot (Figure 2) shows how the collected data expands the thematic coverage by displaying cluster distributions for both datasets. Collected data cluster 0 (Work Struggles) shows an expansion of themes around work experience and internships, academic support systems, and mental health or emotional struggles, which are less represented in the original data. Data cluster 1 (Family Dynamics) has more diverse and specific family backgrounds, struggles with independence and support, and impact of wider social and economic systems. Data cluster 2 (Mental Health) introduces more detailed reflections on emotional struggles and mental health challenges. Data cluster 3 (Societal Challenges) significantly enriches the thematic representation of challenges faced by low-SES students, particularly by introducing broader societal and personal insights that were underrepresented in the original dataset. Data cluster 4 (Systemic Critiques) reflects more detailed critiques of systemic issues affecting students, such as the cost of education, the student debt crisis, and the unrealistic promises of higher education as a golden ticket to success.

The final dataset began with a seed of 64 data points and, through the application of the proposed methodology, expanded to include 323 new data points. Both the ground truth and collected datasets were processed using prompt engineering with the LLaMA model to extract background information, struggles during higher education, and solutions students devised to improve their situations. While some data points have missing background, struggle, or solution information due to limitations in LLM extraction, we are actively working on refining the dataset through a rigorous gold-standard validation process. Additionally, we performed sentiment analysis on the entire text of both datasets using a sentiment fine-tuned model (Camacho-collados et al.). In the ground truth dataset, the sentiment distribution consisted of 10 positive, 29 neutral, and 25 negative texts. For the collected data, the distribution shifted to 37 positive, 138 neutral, and 148 negative texts. This significant increase in both neutral and negative sentiments in the collected dataset provides a broader scope for future exploration of the emotional landscape of low-SES students.

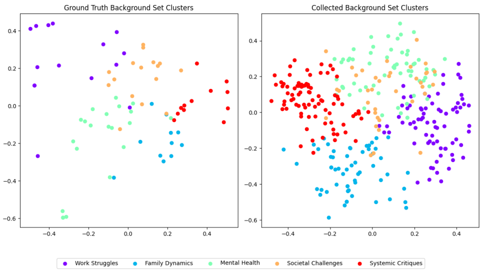


Figure 2: Ground Truth and Collected Data Clusters of Background Information

6 Contributions

First, we developed a dataset of unstructured narratives from low-SES students by semi-automatically collecting and expanding data from Reddit, producing a dataset five times larger than the initial seed. We will make our code and data publicly available for the research community. Second, we designed a hybrid model that combines machine learning and human insights to classify low-SES student narratives, comparing the performance of traditional ML models with recent LLMs. Third, given the challenges associated with data collection in this underexplored area, our work paves the way for fully automating this process, encouraging future research to focus on the educational barriers faced by low-SES students. Finally, by releasing a large, high-quality corpus of low-SES student narratives, our work enables a wide range of downstream applications such as profiling socioeconomic discourse, detecting financial or psychological hardship, and supporting educational policy research. The dataset can also serve as a valuable benchmark for future NLP models targeting underrepresented demographic dimensions.

7 Ethical and Societal Impact

First, while we strive for fairness, it is challenging to ensure equal representation across geographic regions and genders in our dataset. This could lead to unintentional biases that affect the results and interpretations of our work. Second, although we will release the model under the appropriate license to ensure compliance with legal and ethical standards, there remains a risk of misuse. Specifically, the model could be used to classify low-SES individuals from publicly available narratives, potentially exposing them to harmful activities such as discrimination or exploitation. To mitigate this, we will enforce user agreements that explicitly prohibit harmful uses. Finally, although the data we collected is anonymous, it was sourced from public online forums, and we, as authors, cannot edit or delete this data once retrieved. This raises privacy concerns, as individuals may not have anticipated their posts being used for research purposes, even in an anonymized form. Additionally, although the narratives are public and anonymous, we still make sure we have IRB exempt status before publishing our collected narratives. Despite these concerns, we believe our work will have a positive societal impact. By providing a deeper understanding of the challenges faced by low-SES students, our findings could inform educational policies and initiatives aimed at addressing socioeconomic disparities. Ultimately, our research could contribute to greater equity and inclusion for marginalized communities.

References

Nancy E. Adler and Katherine Newman. 2017. The impact of socioeconomic status on access to healthcare: A review of the literature. *Social Science Medicine*, 181:25–33.

Jacob H. Brown and Michael L. Carr. 2013. The impact of financial aid on college enrollment and completion: Evidence from a randomized study.

Jose Camacho-collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, and Eugenio Martínez Cámara. TweetNLP: Cutting-edge natural language processing for social media.

Amanda Cercas Curry, Zeerak Talat, and Dirk Hovy. 2024. Impoverished language technology: The lack of (social) class in NLP. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC–COLING 2024)*, pages 8675–8682, Torino, Italia. ELRA and ICCL.

Ian T. Jolliffe. 2002. *Principal Component Analysis*. Springer Series in Statistics. Springer, New York.

Motti Kelbessa, Estephanos Jebessa, and Labiba Jahan. 2024. Addressing educational inequalities of low ses students: Leveraging natural language processing for impact. In *Proceedings of the 17th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 388–391.

Vasileios Lampos, Nikolaos Aletras, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2014. Predicting and characterising user impact on twitter. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 405–413.

Sang Min Lee, M Harry Daniels, Ana Puig, Rebecca A Newgent, and Suk Kyung Nam. 2008. A data-based model to predict postsecondary educational attainment of low-socioeconomic-status students. *Professional School Counseling*, 11(5):2156759X0801100504.

Fons Mentink. 2016. Machine driven predictions of the socio-economic status of twitter users. Master's thesis, University of Twente.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2022. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Judith Scott-Clayton. 2015. The role of financial aid in promoting college access and success: Research evidence and proposals for reform. *Journal of Student Financial Aid*, 45(3):3.

Southern Methodist University (SMU). 2019. Think-play-hack: World views.

Stefan Stieglitz, Milad Mirbabaie, Björn Ross, and Christoph Neuberger. 2018. Social media analytics–challenges in topic discovery, data collection, and data preparation. *International journal of information management*, 39:156–168.

Marvin A Titus. 2006. Understanding college degree completion of students with low socioeconomic status: The influence of the institutional financial context. *Research in Higher Education*, 47:371–398.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. *arXiv preprint arXiv:1909.00161*.

A Appendix

A.1 Data

You need to buy textbooks or find PDFs. Talk to your faculty adviser if you ever have questions, and make sure you're talking to the general university adviser too. Welcome, if you've ever been to the bank. Some teachers might inflate grades, but in the end, don't skip class or slack off on homework. Stop studying for exams the night before—it's a terrible habit. Get an internship as quickly as possible, and try to have a job lined up before graduation. Well, go to the career center and make friends with people who are getting jobs. Put together a serious resume and cover letter as soon as you can. No one is going to be impressed with fancy colors or formatting in the professional world. The career center can help you critique them for free. You're also going to get wrecked by student loans in a few years, so prepare wisely unless you're Richie Rich.

A.2 Classification

Model	Parameter values
RF	n_estimators = 100, random_state = 46. Random Forest model using 100 trees to ensure a balanced performance.
MNB	Default parameters, suitable for text data with TF-IDF representation. The Multinomial Naive Bayes assumes feature independence and is efficient for large-scale text data.
SVM	kernel = linear, probability = True, random_state = 46. A linear kernel is efficient for text classification, with probability estimates enabled for evaluation purposes.
LR	max_iter = 500, random_state = 46. LR with a limit on the number of iterations to ensure convergence.
BERT	Model = bert-base-uncased, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned for binary classification with early stopping, patience = 8.
DistilBERT	Model = distilbert-base-uncased, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned for binary classification with early stopping, patience = 8.
ALBERT	Model = albert-base-v2, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned for binary classification with early stopping, patience = 8.
BART	Model = facebook/bart-large-mnli, MAX_LEN = 512, TRAIN_BATCH_SIZE = 16, VALID_BATCH_SIZE = 16, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 1024, dropout = 0.3. Fine-tuned BART with a binary classification head and early stopping, patience = 8.
DeBERTa	Model = microsoft/deberta-base, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned for binary classification using CLS token with early stopping, patience = 8.
ELECTRA	Model = google/electra-base-discriminator, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned for binary classification using CLS token with early stopping, patience = 8.
XLNet	Model = xlnet-base-cased, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, hidden size = 768, dropout = 0.3. Fine-tuned XLNet model for binary classification with early stopping, patience = 8.
T5	Model = t5-base, MAX_LEN = 512, TRAIN_BATCH_SIZE = 32, VALID_BATCH_SIZE = 32, EPOCHS = 40, LEARNING_RATE = 1e-05, output = logits for binary classification with early stopping, patience = 8.
LLaMA	Model = princeton-nlp/Sheared-LLaMA-1.3B, MAX_LEN = 512, TRAIN_BATCH_SIZE = 8, VALID_BATCH_SIZE = 8, EPOCHS = 40, LEARNING_RATE = 1e-05, dropout = 0.3, with LoRA fine-tuning for binary classification with early stopping, patience = 8.
GPT-2	Model = gpt, MAX_LEN = 512, TRAIN_BATCH_SIZE = 2, VALID_BATCH_SIZE = 2, EPOCHS = 10, LEARNING_RATE = 1e-04, dropout = 0.3, early stopping, patience = 5.
DistilGPT-2	Model = distilgpt2, MAX_LEN = 512, TRAIN_BATCH_SIZE = 2, VALID_BATCH_SIZE = 2, EPOCHS = 10, LEARNING_RATE = 1e-04, dropout = 0.3, early stopping, patience = 5.
OPT-13B	Model = KoboldAI/OPT-13B-Erebus, MAX_LEN = 64, batch_size = 1, gradient checkpointing enabled, mixed precision used, early stopping, patience = 5.
LoRA GPT-2	Model = gpt2, MAX_LEN = 512, TRAIN_BATCH_SIZE = 2, VALID_BATCH_SIZE = 2, EPOCHS = 10, LEARNING_RATE = 1e-04, dropout = 0.3, with LoRA fine-tuning, early stopping, patience = 5.
LoRA LLaMA-1.3B	Model = princeton-nlp/Sheared-LLaMA-1.3B, MAX_LEN = 512, TRAIN_BATCH_SIZE = 8, VALID_BATCH_SIZE = 8, EPOCHS = 40, LEARNING_RATE = 1e-05, dropout = 0.3, with LoRA fine-tuning, early stopping, patience = 8.
LoRA Distill-GPT	Model = distilgpt2, MAX_LEN = 512, TRAIN_BATCH_SIZE = 2, VALID_BATCH_SIZE = 2, EPOCHS = 10, LEARNING_RATE = 1e-04, dropout = 0.3, with LoRA fine-tuning, early stopping, patience = 5.
Claudia few-shot	Model = Claudia few-shot, few-shot prompt-based learning, early stopping patience = 5, uses structured prompting with the dataset of examples.
LLaMA-7B few-shot	Model = LLaMA-7B, few-shot learning using a structured prompting with the dataset of examples.

Table 3: Summary of the architecture and parameters for each model used for classification.

Model	**SES**	**Precision**	**Recall**	**F_1**	**Model**	**SES**	**Precision**	**Recall**	**F_1**
Random Forest	not-Low	0.57	0.40	0.47	Multinomial Naive Bayes	not-Low	1.00	0.40	0.57
	Low	0.54	0.70	0.61		Low	0.62	1.00	0.77
	Avg.	0.55	0.55	0.54		Avg.	0.81	0.70	0.67
Support Vector Machine (SVM)	not-Low	0.58	0.70	0.64	Logistic Regression	not-Low	0.58	0.70	0.64
	Low	0.62	0.50	0.56		Low	0.62	0.50	0.56
	Avg.	0.60	0.60	0.60		Avg.	0.60	0.60	0.60
Gradient Boosting	not-Low	0.62	0.80	0.70	RoBERTa	not-Low	0.78	0.70	0.74
	Low	0.71	0.50	0.59		Low	0.73	0.80	0.76
	Avg.	0.66	0.65	0.64		Avg.	0.75	0.75	0.75
DeBERTa	not-Low	0.69	0.90	0.78	ELECTRA	not-Low	0.69	0.90	0.78
	Low	0.86	0.60	0.71		Low	0.86	0.60	0.71
	Avg.	0.77	0.75	0.74		Avg.	0.77	0.75	0.74
ALBERT	not-Low	0.71	0.50	0.59	XLNet	not-Low	0.53	0.90	0.67
	Low	0.62	0.80	0.70		Low	0.67	0.20	0.31
	Avg.	0.66	0.65	0.64		Avg.	0.60	0.55	0.49
T5	not-Low	0.50	0.30	0.38	OPT-13B fewshot	not-Low	0.50	0.90	0.64
	Low	0.50	0.70	0.58		Low	0.50	0.10	0.17
	Avg.	0.50	0.50	0.48		Avg.	0.50	0.50	0.40
Claudia fewshot	not-Low	0.53	0.90	0.67	LLamA-1.3B fewshot	not-Low	0.50	0.80	0.62
	Low	0.67	0.20	0.31		Low	0.50	0.20	0.29
	Avg.	0.60	0.55	0.49		Avg.	0.50	0.50	0.45
LoRA Fine-Tune GPT-2	not-Low	0.50	1.00	0.67	LLaMA-7B fewshot	not-Low	0.38	0.50	0.43
	Low	0.00	0.00	0.00		Low	0.29	0.20	0.24
	Avg.	0.25	0.50	0.33		Avg.	0.34	0.35	0.34
LoRA Fine-Tune DistilGPT	not-Low	0.50	1.00	0.67	LoRA Fine-Tune LLaMA-1.3B	not-Low	0.50	1.00	0.67
	Low	0.00	0.00	0.00		Low	0.00	0.00	0.00
	Avg.	0.25	0.50	0.33		Avg.	0.25	0.50	0.33
bart-large-mnli Zero Shot	not-Low	0.80	0.80	0.80	DistilBERT	not-Low	0.78	0.70	0.74
	Low	0.80	0.80	0.80		Low	0.73	0.80	0.76
	Avg.	**0.80**	**0.80**	**0.80**		Avg.	0.75	0.75	0.75
BERT	not-Low	0.75	0.60	0.67	RoBERTa-large	not-Low	0.78	0.70	0.74
	Low	0.67	0.80	0.73		Low	0.73	0.80	0.76
	Avg.	0.71	0.70	0.70		Avg.	0.75	0.75	0.75

Table 4: First Iteration Performance of different models for classifying socioeconomic classes. Avg. = Macro average.

A.3 Results

This Section has the results tables from the three iterations

Model	**SES**	**Precision**	**Recall**	$\mathbf{F_1}$	**Model**	**SES**	**Precision**	**Recall**	$\mathbf{F_1}$
Random Forest	not-Low	0.72	0.82	0.77	Multinomial Naive Bayes	not-Low	0.89	0.61	0.72
	Low	0.77	0.65	0.71		Low	0.69	0.92	0.79
	Avg.	0.74	0.74	0.74		Avg.	0.79	0.76	0.75
Support Vector Machine (SVM)	not-Low	0.86	0.89	0.88	Logistic Regression	not-Low	0.84	0.93	0.88
	Low	0.88	0.85	0.86		Low	0.91	0.81	0.86
	Avg.	**0.87**	**0.87**	**0.87**		Avg.	**0.87**	**0.87**	**0.87**
Gradient Boosting	not-Low	0.75	0.75	0.75	RoBERTa	not-Low	0.74	0.82	0.78
	Low	0.73	0.73	0.73		Low	0.78	0.69	0.73
	Avg.	0.74	0.74	0.74		Avg.	0.76	0.76	0.76
DeBERTa	not-Low	0.75	0.75	0.75	ELECTRA	not-Low	0.73	0.96	0.83
	Low	0.73	0.73	0.73		Low	0.94	0.62	0.74
	Avg.	0.74	0.74	0.74		Avg.	0.83	0.80	0.79
ALBERT	not-Low	0.66	0.68	0.67	XLNet	not-Low	0.68	0.46	0.55
	Low	0.64	0.62	0.63		Low	0.57	0.77	0.66
	Avg.	0.65	0.65	0.65		Avg.	0.63	0.61	0.60
T5	not-Low	0.84	0.57	0.68	OPT-13B fewshot	not-Low	0.53	0.57	0.55
	Low	0.66	0.88	0.75		Low	0.50	0.46	0.48
	Avg.	0.75	0.72	0.72		Avg.	0.52	0.52	0.52
Claudia fewshot	not-Low	0.51	0.71	0.60	LLamA-1.3B fewshot	not-Low	0.45	0.61	0.52
	Low	0.47	0.27	0.34		Low	0.31	0.19	0.24
	Avg.	0.49	0.50	0.47		Avg.	0.38	0.41	0.38
Lora finetune GPT-2	not-Low	0.52	1.00	0.68	LLamA-7B fewshot	not-Low	0.79	0.33	0.46
	Low	0.00	0.00	0.00		Low	0.74	0.78	0.76
	Avg.	0.27	0.52	0.35		Avg.	0.69	0.64	0.66
Lora finetune DistillGPT	not-Low	0.52	1.00	0.68	LoRA Fine-Tuning LLaMA-1.3B	not-Low	0.46	0.75	0.57
	Low	0.00	0.00	0.00		Low	0.12	0.04	0.06
	Avg.	0.27	0.52	0.35		Avg.	0.30	0.41	0.32
bart-large-mnli Zero Shot	not-Low	0.00	0.00	0.00	DistilBert	not-Low	0.85	0.61	0.71
	Low	0.45	0.88	0.60		Low	0.68	0.88	0.77
	Avg.	0.22	0.43	0.29		Avg.	0.77	0.74	0.74
BERT	not-Low	0.79	0.54	0.64	RoBERTa-large	not-Low	0 0.85	0.61	0.71
	Low	0.63	0.85	0.72		Low	0.68	0.88	0.77
	Avg.	0.71	0.69	0.68		Avg.	0.74	0.74	0.74

Table 5: Second Iteration Performance of different models for classifying socioeconomic classes on second iteration. Avg. = Weighted average by the number of narratives.

Model	**SES**	**Precision**	**Recall**	$\mathbf{F_1}$	**Model**	**SES**	**Precision**	**Recall**	$\mathbf{F_1}$
Random Forest	not-Low	0.63	0.53	0.57	Multinomial Naive Bayes	not-Low	0.70	0.45	0.55
	Low	0.60	0.69	0.64		Low	0.60	0.81	0.69
	Avg.	0.61	0.61	0.61		Avg.	0.65	0.63	0.62
Support Vector Machine (SVM)	not-Low	0.59	0.63	0.61	Logistic Regression	not-Low	0.84	0.93	0.88
	Low	0.61	0.58	0.59		Low	0.64	0.65	0.65
	Avg.	0.60	0.60	0.60		Avg.	0.64	0.64	0.64
Gradient Boosting	not-Low	0.53	0.65	0.58	RoBERTa	not-Low	0.61	0.92	0.73
	Low	0.56	0.44	0.49		Low	0.85	0.42	0.56
	Avg.	0.55	0.54	0.54		Avg.	0.73	0.67	0.65
DeBERTa	not-Low	0.54	0.69	0.60	ELECTRA	not-Low	0.54	0.82	0.65
	Low	0.58	0.42	0.49		Low	0.64	0.31	0.42
	Avg.	0.56	0.55	0.55		Avg.	0.59	0.56	0.53
ALBERT	not-Low	0.54	0.61	0.57	XLNet	not-Low	0.62	0.75	0.68
	Low	0.57	0.50	0.53		Low	0.69	0.56	0.62
	Avg.	0.55	0.55	0.55		Avg.	0.66	0.65	0.65
T5	not-Low	0.55	0.55	0.55	OPT-13B fewshot	not-Low	0.50	0.69	0.58
	Low	0.56	0.56	0.56		Low	0.52	0.33	0.40
	Avg.	0.55	0.55	0.55		Avg.	0.51	0.50	0.49
Claudia fewshot	not-Low	0.49	0.76	0.60	LLamA-1.3B fewshot	not-Low	0.53	0.71	0.61
	Low	0.50	0.23	0.32		Low	0.57	0.38	0.46
	Avg.	0.50	0.50	0.46		Avg.	0.55	0.54	0.53
Lora finetune GPT-2	not-Low	0.50	1.00	0.66	LLamA-7B fewshot	not-Low	0.48	0.65	0.55
	Low	0.00	0.00	0.00		Low	0.47	0.31	0.37
	Avg.	0.25	0.50	0.33		Avg.	0.47	0.48	0.46
Lora finetune DistillGPT	not-Low	0.50	1.00	0.66	LoRA Fine-Tuning LLaMA-1.3B	not-Low	0.50	1.00	0.66
	Low	0.00	0.00	0.00		Low	0.12	0.04	0.06
	Avg.	0.25	0.50	0.33		Avg.	0.25	0.50	0.33
bart-large-mnli Zero Shot	not-Low	0.45	0.45	0.45	DistilBert	not-Low	0.65	0.51	0.57
	Low	0.46	0.46	0.46		Low	0.60	0.73	0.66
	Avg.	0.46	0.46	0.46		Avg.	0.63	0.62	0.62
BERT	not-Low	0.56	0.55	0.55	RoBERTa-large	not-Low	0.70	0.76	0.73
	Low	0.57	0.58	0.57		Low	0.74	0.67	0.71
	Avg.	0.56	0.56	0.56		Avg.	**0.72**	**0.72**	**0.72**

Table 6: Third Iteration Performance of different models for classifying socioeconomic classes on second iteration. Avg. = Weighted average by the number of narratives.

A.4 Prompt Engineering for Extracting Background, Solutions, and Struggles Information

This subsection provides a detailed explanation of the prompt engineering techniques used to extract background information, solutions, and struggles from the narratives of low-SES students. By constructing specific prompts and using the LLaMA model, we ensured the precise extraction of information in a structured format, relying only on direct quotes from the texts. These prompts aim to assist in understanding the challenges and efforts described by low-SES students, ensuring that no additional information is added or altered during extraction.

A.4.1 Model Pipeline Setup

We utilized the `transformers` library from Hugging Face to create a pipeline for text generation and extraction. The LLaMA model was fine-tuned for generating outputs that align with our prompt design. The following configuration was applied to the pipeline for all tasks:

- **max_new_tokens=300**: Sets the maximum number of tokens to generate during extraction. This ensures that the output is concise and focused.
- **do_sample=False**: Sampling is disabled to provide deterministic and consistent outputs from the model.
- **temperature=0.5**: A lower temperature value ensures less randomness in the output, resulting in more controlled and accurate text generation.
- **device**: The model was configured to run on either GPU (if available) or CPU, ensuring flexibility in processing.

The prompts were specifically designed to elicit structured information, such as family background, solutions, and struggles, from the students’ narratives. Below, we describe each function used to extract these key elements.

A.4.2 Extracting Background Information

The first step was to extract background information, particularly focusing on the family situations described in the narratives. The goal was to identify direct quotes that describe the family context of the students, such as financial hardships or living conditions.

The following function was designed to handle this task:

```
def extract_background(text):
    prompt = f’’’
    All the texts provided are written by low-SES (SES) students
        who are writing about their struggles.
    ...
    Important: Extract the following information exactly from the
        text without adding or changing any words:
    - background or any texts about family situations (directly
        quoted from the text)

    Text: {text}

    valid JSON Output (only with direct quotes from the text):
    ’’’

    output = llama_pipeline(prompt, max_new_tokens=200, do_sample=
        False, temperature=0.5)
    generated_text = output[0][’generated_text’]
    # Process output for background quotes
    ...
```

This prompt ensures that only direct quotes describing the students’ family background are extracted and returned in a valid JSON format.

A.4.3 Extracting Solutions Information

In addition to background information, we extracted the solutions that students employed to address their struggles. These solutions may involve actions taken to overcome financial or social barriers, as well as any efforts to improve their academic or personal circumstances.

The function below is responsible for extracting the solutions from each text:

```
def extract_solutions(text):
    prompt = f’’’
    All the texts provided are written by low-SES (SES) students
        who are writing about their struggles.
    ...
    Important: Extract the following information exactly from the
        text without adding or changing any words:
    - Solutions or actions they took to address these struggles (
        directly quoted from the text)

    Text: {text}

    valid JSON Output (only with direct quotes from the text):
    ’’’

    output = llama_pipeline(prompt, max_new_tokens=200, do_sample=
        False, temperature=0.5)
    generated_text = output[0][’generated_text’]
    # Process output for solutions quotes
    ...
```

This function captures the strategies or actions the students took to manage or overcome their struggles, returning the data in a structured JSON format for analysis.

A.4.4 Extracting Struggles Information

The third aspect of our extraction was to focus on the specific struggles described by the students. These struggles include financial, psychological, physical, or social hardships. The function uses a similar approach, instructing the model to identify and extract direct quotes related to the students’ difficulties.

The function for extracting struggles is as follows:

```
def extract_struggles(text):
    prompt = f’’’
    All the texts provided are written by low-SES (SES) students
        who are writing about their struggles.
    ...
    Important: Extract the following information exactly from the
        text without adding or changing any words:
    - Struggles they faced (directly quoted from the text)

    Text: {text}

    Output valid JSON with only direct quotes related to struggles:
    ’’’

    output = llama_pipeline(prompt, max_new_tokens=300, do_sample=
        False, temperature=0.5)
    generated_text = output[0][’generated_text’]

    # Process and return the generated text as JSON
    ...
```

This function extracts the struggles faced by the students and returns them as direct quotes in a JSON structure.

A.4.5 Post-processing and Valid JSON Output

In all cases, after the output is generated by the LLaMA model, the generated text is processed to extract the relevant information in JSON format. The output is validated to ensure it contains the correct fields (e.g., background, solutions, or struggles), and any parsing errors are handled gracefully by returning a fallback structure if needed.

The extracted data is then consolidated into a structured format for further analysis. This structured data helps in understanding the key themes and experiences described by the low-SES students.

A.4.6 Conclusion

By employing these prompt engineering techniques, we were able to extract detailed and structured information regarding the backgrounds, struggles, and solutions described by the students in their narratives. The use of precise prompts, alongside the LLaMA model, allowed for accurate extraction of direct quotes, preserving the authenticity of the students' experiences. This extracted data provides valuable insights into the challenges faced by low-SES students and their efforts to overcome them.

A.4.7 Limitations

We acknowledge several limitations in our current research that we plan to address in future work. First, although our dataset offers valuable insights into the experiences of low-SES students, it is limited to narratives collected from Reddit, primarily from a small number of subreddits. This platform and genre bias may affect the generalizability of our findings, as narratives from other forums or formats (e.g., blogs, surveys, or interviews) could present significantly different linguistic and contextual patterns.

Second, while our data were annotated semi-automatically, it has not yet undergone a rigorous double-annotation or gold-standard validation process, which we are currently working on to enhance the dataset's reliability. Implementing this more precise annotation method will improve the consistency and robustness of our results.

Third, the thresholding heuristic used in our model—a fixed confidence cutoff of 0.7—was chosen based on empirical observation but lacks formal justification. This may have led to the exclusion of valid narratives or inclusion of false positives, affecting the overall quality of the extended dataset. In future iterations, we plan to explore more adaptive or learned thresholding techniques, possibly incorporating human-in-the-loop validation.

Fourth, although our semi-automatic data collection model showed promising results, we have not yet evaluated the utility of the expanded dataset in downstream NLP tasks. For example, it remains to be seen whether the additional 323 narratives improve the performance of SES classification or sentiment models. We plan to address this by testing our dataset's impact on real-world applications.

Fifth, while our analysis reveals meaningful linguistic patterns, the paper would benefit from a deeper engagement with educational or sociological theories. Connecting our findings more explicitly to existing research on SES, educational inequality, or psychological well-being could strengthen the interpretability and societal relevance of our work.

Finally, our dataset is currently constrained to a specific time frame. Expanding it to cover a wider range of years will allow us to better capture the evolving nature of SES-related challenges in education. We also plan to further refine our model using active learning to increase its scalability and independence.

Construction Identification and Disambiguation Using BERT: A Case Study of NPN

Wesley Scivetti **Nathan Schneider**
Georgetown University
{wss37, nathan.schneider}@georgetown.edu

Abstract

Construction Grammar hypothesizes that knowledge of a language consists chiefly of knowledge of form–meaning pairs (“constructions”) that include vocabulary, general grammar rules, and even idiosyncratic patterns. Recent work has shown that transformer language models represent at least some constructional patterns, including ones where the construction is rare overall. In this work, we probe BERT’s representation of the form and meaning of a minor construction of English, the NPN (noun–preposition–noun) construction—exhibited in such expressions as *face to face* and *day to day*—which is known to be polysemous. We construct a benchmark dataset of semantically annotated corpus instances (including distractors that superficially resemble the construction). With this dataset, we train and evaluate probing classifiers. They achieve decent discrimination of the construction from distractors, as well as sense disambiguation among true instances of the construction, revealing that BERT embeddings carry indications of the construction’s semantics. Moreover, artificially permuting the word order of true construction instances causes them to be rejected, indicating sensitivity to matters of form. We conclude that BERT does latently encode at least some knowledge of the NPN construction going beyond a surface syntactic pattern and lexical cues.

1 Introduction

The “black box” nature of Language Models (LMs) like has spawned a great deal of research investigating the extent to which these LMs are able to represent and understand a variety of linguistic phenomena (Linzen and Baroni, 2021; Rogers et al., 2021; Chang and Bergen, 2024). There has been substantial work focusing on many aspects of linguistic knowledge, including hierarchical structure (Clark et al., 2019; Hewitt and Manning, 2019; Jawahar et al., 2019), lexical semantics (Chang and Chen, 2019; Vulić et al., 2020), negation (Ettinger, 2020), agreement phenomena (Linzen et al., 2016; Weissweiler et al., 2023), and filler-gap dependencies (Wilcox et al., 2018, 2024). Broadly, these results show that even relatively modest sized LSTMs and transformer models are able to demonstrate nontrivial (though far from perfect) linguistic knowledge. However, there is some indication that these models are sometimes reliant on more surface level heuristics, and fail in situations which are straightforward to humans (McCoy et al., 2019; Ettinger, 2020). More generally, language models have been generally shown to struggle in out-of-domain situations (McCoy et al., 2024) and have some difficulty applying linguistic paradigms to nonce words (Weissweiler et al., 2023) and rare syntactic constructions (Scivetti et al., 2025).

Thus, there is need to evaluate language models on a range of linguistic tasks which go beyond the more studied “core” linguistic phenomena. Such work serves to provide a more complete picture of how language models succeed and fail across the broad spectrum of phenomena in language. Indeed, beyond the more mainstream notions of linguistic structure and information, there is also work on investigating LM knowledge of more idiosyncratic *constructions*, as defined by Construction Grammar. Construction Grammar is broadly a family of linguistic theories which consider all parts of language to be made up of constructions, which are pairings of linguistic *forms* with *meaning* or function (Goldberg 1995; Croft 2001, *inter alia*). It remains unclear the extent to which LMs may implicitly view constructions as distinct units. Because of their emphasis on pairing form with meaning, CxG theories provide possibilities for testing language model capabilities at the interface of form and meaning for different aspects of language, in contrast to past work which has focused on either syntax (e.g. Hewitt and Manning 2019) or semantics (e.g. Vulić et al. 2020) in isolation. A substantial and growing amount of research has recently

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 365–376
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

focused on the intersection of LM knowledge and Construction Grammar (Tayyar Madabushi et al., 2020; Tseng et al., 2022; Pannitto and Herbelot, 2023; Veenboer and Bloem, 2023, *inter alia*), with a particular focus on argument structure constructions (Li et al., 2022), the English Comparative Correlative (Weissweiler et al., 2022), and the English AANN construction (Chronis et al., 2023; Mahowald, 2023). While these studies have provided valuable insight into LM processing of constructions with varying levels of schematicity, there remain many constructions which have not been addressed at all in previous work. Furthermore, while Zhou et al. (2024) do test model understanding of constructions which are similar in form, no past work has focused on individual constructions as polysemous units. We argue this is a gap in past work, as constructions, like words, can have related but distinct meanings that must be properly disambiguated in context in order for correct interpretation. We address this gap by providing experiments which pair formal sensitivity with semantic disambiguation in a controlled manner for a single construction.

This work is the first to study whether language models capture the NPN construction (Jackendoff, 2008), an infrequent yet productive pattern exhibited in expressions like *face to face* and *day to day*. Even for the subset where two instances of the same noun are linked by the preposition *to*, the pattern is polysemous, and sequences matching this pattern on the surface are not always instances of the construction (§2). Guided by CxG theory, we separate our inquiry in terms of the construction's *form* and *meaning* in context. To investigate language modeling of NPN, we:

- Construct and annotate a novel dataset of natural NPN examples from COCA (§3).
- Probe BERT's ability to distinguish true constructional instances from related constructions and artificial orders (§4 and §5).
- Introduce the task of construction sense disambiguation and perform experiments using our dataset (§6).

To summarize our findings, we show that probes using BERT embeddings are able to both identify correct instances of NPN and disambiguate the construction within context at respectable accuracy. Overall, these findings indicate that BERT latently encodes relevant information to the NPN construction, leading to strong sensitivity to both the construction's form and its meaning.

2 The NPN Construction

The NPN construction (Jackendoff, 2008) follows the general pattern of Noun + Preposition + Noun. Below are 2 examples of the NPN construction. These examples, along with all others, are taken from the Corpus of Contemporary American English (COCA, Davies 2010).

(1) There is a rebellious quality to your **day to day** responses which have not gone unnoticed.

(2) I need you to get this **word for word**.

Given the general rules of English, the NPN construction has several unique properties, which we argue separate it from more "core" linguistic phenomena. Firstly, the nouns almost always lack determiners, which is unusual for count nouns like "day". Secondly, the construction can occur in a variety of syntactic positions, including as an adverbial modifier (as in (2)) and as a prenominal modifier (as in (1)). Finally, it conveys a meaning which is not entirely predictable from its components, and varies considerably depending on the preposition. Common meanings of the NPN construction are the SUCCESSION meaning (shown in (1)) and the MATCHING/COMPARISON meaning (shown in (2)). See Jackendoff (2008) for an overview of the NPN construction and the common meanings associated with various prepositional lemmas.

While it is conceptually and intuitively appealing to think of NPN as a single construction, some work has argued in favor of viewing NPN as a group of related constructions, which are linked within the mind but not necessarily dominated by a single overarching abstract NPN construction (Sommerer and Baumann, 2021). Due to the wide variety of meanings and distributions of the different NPN constructions, we choose to limit our focus to a single subtype of NPNs, which all share the lemma "to" as their preposition, which we refer to as the N*to*N construction. There is still considerable semantic variation even within the N*to*N construction, with 2 broad meanings that we highlight: SUCCESSION (shown in (3)) and JUXTAPOSITION (shown in (4)).

(3) I was living **moment to moment**.

(4) You can preserve core warmth by huddling with a buddy, **chest to chest**.

While there are additional meanings of NPN that do not occur with "to" as the preposition, it is one of the only prepositions that is ambiguous in the NPN construction. By not considering examples of NPN with other prepositions, we remove the prepositional lemma as a potential shallow cue that models could learn to predict the construction's semantics. While there are arguably examples of NPNs where the two nouns are not identical, we limit our analysis to cases where the two nouns in the construction match exactly. This allows us to easily gather examples of the construction from corpus data.

3 Dataset

3.1 Corpus Gathering and Cleaning

In this work, we endeavor to use natural corpus data to the extent that it was possible. First, we use a simple pattern matching query to extract instances of the sequence Noun + "to" + Noun from COCA. We extract the examples from the corpus in a fixed window of +/- 50 tokens from the construction, and then used Stanza (Qi et al., 2020) to segment the results into sentences and extract the sentences which contained N*to*Ns. We automatically exclude sentences which contained "from" preceding the construction, because *from* N *to* N does not have exactly the same distribution as the more general N*to*N (Jackendoff, 2008), and is sometimes studied as a separate (but closely related) construction (Zwarts, 2013).

After extracting all sentences which contained a possible instance of N*to*N, we then manually clean the data, removing sentences that were either too short (<5 tokens) or contained too many typos. We annotate all instances of the construction for their semantic subtype, and double annotate roughly 25% of the dataset, achieving an agreement of 84% and a Cohen's kappa value of .754 between the two annotators, indicating strong agreement.[1] The final dataset has 6599 instances of N*to*N, of which 1885 were double annotated.

3.2 Near Minimal Pairs

In addition to true instances of the N*to*N construction, we also find grammatical corpus instances of Noun + "to" + Noun patterns, which are not instances of the construction. These patterns often occur when a verb licenses a direct object and a "to" prepositional phrase, and the direct object and the object of the preposition happen to have the same lemma. Three examples are shown below in (5), (6), and (7).

(5) Then there's the problem of sticking plastic to plastic.

(6) In Rome largesse was doled out by individuals to individuals.

(7) I don't have time to time travel ...

We do not consider such cases to be examples of the N*to*N construction because the surface pattern of Noun + Preposition + Noun clearly arises from a different syntactic context (e.g. a verb licensing a direct object and a PP modifier). Furthermore, the meanings of these examples do not evoke the unique semantics that accompany the N*to*N construction. While these cases are not instances of the N*to*N construction, they do provide a set of negative examples which we can use to probe the model's ability to recognize true N*to*N constructions. Throughout this paper, we refer to this set of examples as instances of the N*to*N *distractors*, since we test of if the model is "distracted" by the shallow similarity of the examples to the NPN construction. We refer to true examples of N*to*N as instances of the N*to*N *construction*. Since these N*to*N examples exhibit the same surface form as the N*to*N *construction*, we consider them to be near minimal pairs, following Weissweiler et al. (2022) who extract near minimal pairs from corpus data based on part-of-speech patterns. While these sentences inevitably contain more lexical biases than a true minimal pair dataset, they are completely natural, and provide a good comparison point for a construction where creating true minimal pairs is otherwise difficult (because there is no obvious minimal change that can be made to result in a grammatical sentence that is not an example of the construction, similar to the struggles of Weissweiler et al. (2022) regarding the Comparative Correlative construction). In total, we collect 456 total instances of N*to*N *distractors* from COCA.

3.3 Train/Test Split

The resulting dataset contains many instances of very common N*to*N constructions, such as "day to day". We control for the effect of these frequent lemmas in two ways. Firstly, we artificially shrink the dataset by randomly sampling 20 sentences for each noun lemma which occurs more than 20

---

[1] Disagreements between the two annotators were resolved through discussion and a gold label was chosen jointly.

	SUCCESSION	JUXTAPOSITION	Distractors
train	289	287	287
test	731	678	72

Table 1: Number of noun–*to*–noun sequences: two meanings of the NPN Construction, as well as *distractors*. Train sets are balanced to be equal between the categories. The remaining examples are left for testing.

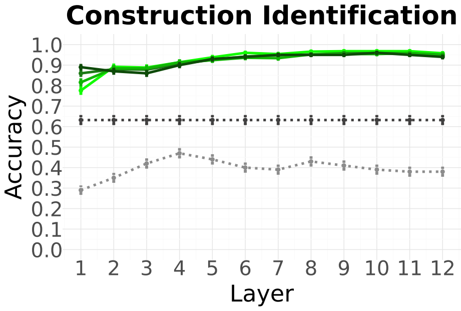


Figure 1: Accuracy of N*to*N *construction* across layers of BERT-base, averaged across 5 random seeds. Maximal accuracy in the mid to late layers. Reducing the number of training examples does not drastically harm performance. The light grey line represents control probe (Hewitt and Liang, 2019) accuracy, which hovers around chance. The dark grey line represents accuracy of the lexical semantic GloVe baseline. Darker lines indicate larger amounts of training examples, with possible values of 10, 25, 100, and 287. Reducing the amount of training examples for the probes does not lead to drastically changed performance. Error Bars indicate 95% confidence intervals over the mean accuracies across the 5 runs.

times, and discard the remaining sentences for the purposes of model training and testing. This is to make sure that no overly common lemmas have an overstated impact on the probing classifier performance.

Secondly, we generate random train/test splits based on lemma of the noun in the N*to*N, meaning that there are no lemmas that are seen in both the training set and the testing set. In other words, if an example with “day to day” is seen during training, a sentence with “day to day” will never be seen during testing (but a sentence with “week to week” might be). Each sentence in the dataset has one target instance of the N*to*N construction.

In Table 1, we report the final dataset sizes, split by semantic subtype for the construction examples. N*to*N *constructions* are much more frequent than the N*to*N *distractor* patterns which serve as their near minimal pairs. We choose to balance the sizes of the two types of examples during training. We take 80 percent of the N*to*N *distractor* patterns for training and withhold twenty percent. We take a similar number of N*to*N *constructions* for training and then test on the remainder, ensuring training sets are balanced between *constructions* and *distractors*.

4 Experiment 1: Constructions vs. Distractors

4.1 Methodology

We probe the ability for BERT to distinguish natural instances of the N*to*N *construction* from natural examples of the N*to*N *distractor* pattern. To address the issue of lexical overlap, we control for the lexical cue of the nouns in N*to*N by making sure there is no overlap of nouns in the training and testing data splits, as described in §3.3. However, it is still entirely possible that the classifier learns to utilize lexical similarity of the nouns in the construction, or even other words beyond the construction. We address this by providing two baseline systems which give perspective on performance based on lexical cues: a *control classifier* (Hewitt and Liang, 2019) and a non-contextual baseline based on GloVe embeddings (Pennington et al., 2014).

Control classifiers involve training new classifiers based on data where the labels are randomized and correspond deterministically to word type, ideally leading to chance performance. Following Hewitt and Liang (2019), who deterministically assign each word a POS tag for their probing experiments, we assign a random positive or negative label deterministically based on the first noun word type in the construction. The performance of these control classifiers should be near chance, in the absence of any spurious correlations which allow the classifier to solve the task given arbitrary labels.

We provide an additional, non-contextual baseline by training a linear classifier on GloVe embeddings for the nouns in the construction as input. It is well known that the NPN construction is biased towards certain lexical types of nouns, such as temporal phrases and body parts (Jackendoff, 2008). Thus, we expect that a classifier trained on the static embedding of the noun alone will achieve nontrivial performance. We argue that if a BERT-based classifier substantially outperforms this baseline, the difference in performance is an indication of nontrivial contextual understanding of the construction as a whole, beyond the lexical semantics of the present nouns.

Following previous probing work which tracks performance layer by layer Liu et al. (2019); Weissweiler et al. (2022), we train a separate probe based on embeddings from each layer of BERT and track performance across layers. We use the BERT-base-cased model, available through the Huggingface transformers library (Wolf et al., 2020), and choose logistic regression as our linear classification architecture.[2] For all experiments and data settings, we run probes with 5 random seeds and report the average results.

4.2 Results

For the probing classifier results, we graph accuracy on the N*to*N *construction* in Figure 1. As we can see, the classifier is relatively strong at distinguishing the N*to*N *construction* from *distractors* even in the early layers, with an accuracy over .90 by layer 5 with full training examples. Additionally, the classifiers are robust to sharp reductions in the number of training examples (shown in lighter shades of green in Figure 1), showing strong performance even with as few as 10 per-class training examples, echoing similar findings for other constructions (Tayyar Madabushi et al., 2020). The control classifier achieves roughly chance performance, meaning that our trained probes have high *selectivity* (Hewitt and Liang, 2019). The lexical semantic baseline using GloVe achieves performance well above chance (≈68%), though its performance lags far behind the BERT-based probes, regardless of how many training example those BERT-based probes receive. This shows that overall, the probing classifier seems to be picking up on some sort of information in BERT which can reliably distinguish the N*to*N *construction* from its near minimal pair N*to*N *distractor* counterparts, beyond what is possible through lexical semantic clues alone. However, the *distractor* examples generally have syntactic structure which is divergent from the *construction* examples. To provide another comparison point, we now test if the existing probes can distinguish true instances of the N*to*N construction from examples with artificially altered word orders.

5 Experiment 2: Perturbing Word Order

As we have seen in §4.2, a BERT-based probe can generally distinguish the N*to*N *distractor* patterns from the N*to*N *construction*. However, we wish to further test how robust the model is at distinguishing the construction from related patterns. While we have compared to naturally occuring near minimal pairs, we now test the classifier on a set of examples with artificially perturbed word order. If the classifier is robust at recognizing the N*to*N *construction*, it should be able to correctly distinguish *construction* instances from artificial sentences with altered non-NPN word orders. To illustrate this point, consider the following two sentences:

(8) I need you to get this **word for word**.

(9) I need you to get this **for word word**.

Example (8) is a copy of (2) and is a true NPN construction. On the other hand, (9) is not an instance of the construction (because it does not follow the NPN word order), and is a generally ungrammatical sentence. We hypothesize that if the probe trained in §4 is not robust to the actual word order pattern of N*to*N, it will be unable to distinguish sentences like (8) from those like (9). If indeed the lexical cues are influencing classifier performance independent of word order, we expect that the classifier will predominantly classify examples like (9) as positive instances of the N*to*N *construction*.

To test this hypothesis, we manipulate the test set of the probe by creating 4 perturbed orderings of each test example sentence: ***PNN***, ***PN***, ***NNP***, ***NP***. A true N*to*N example is shown in (10) the corresponding 4 different perturbed orderings are shown below in (11), (12), (13), and (14).

(10) Go **room to room** removing anything you don't need and selling it. (Original N*to*N)

(11) Go **to room room** removing anything you don't need and selling it. (PNN Perturbed Order)

(12) Go **to room** removing anything you don't need and selling it. (PN Perturbed Order)

(13) Go **room to** removing anything you don't need and selling it. (NP Perturbed Order)

(14) Go **room room to** removing anything you don't need and selling it. (NNP Perturbed Order)

Crucially, we do not retrain the linear probe on this perturbed data. This means that during training, the classifier only saw instances with the correct N + *to* + N ordering, either positive instances of

---

[2] We take the embedding of "to" as the input into the classifier, as some past work has considered it the "head" of the overall construction (Jackendoff, 2008).

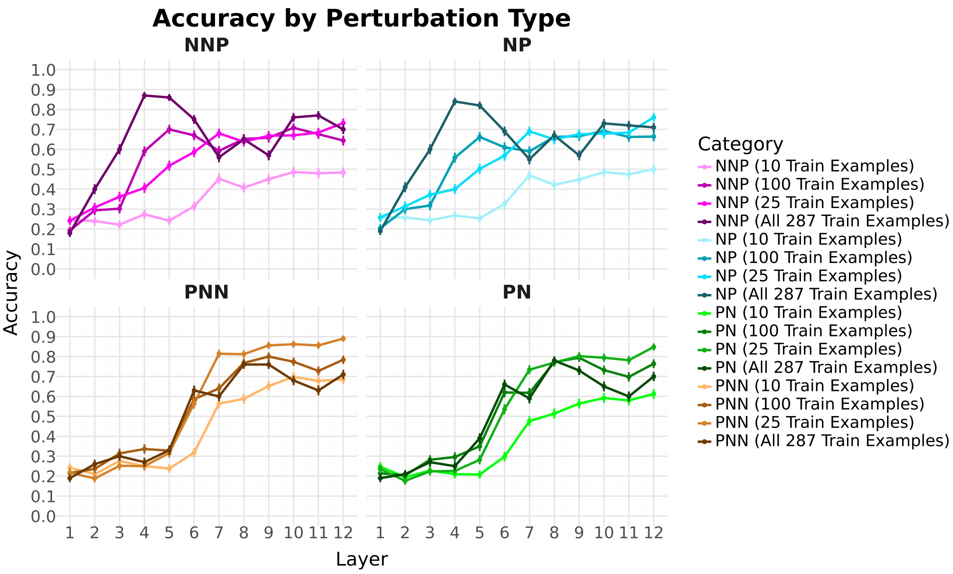


Figure 2: Accuracy of perturbed orderings of original N*to*N *constructions*. Since the perturbed word orders are not true instances of the construction, the true class is negative for all instances. High accuracy indicates that probes are rejecting the validity of the artificial orderings. Lighter colors represent fewer training examples for the probings. Error bars indicate 95% confidence intervals over the average of 5 random seeds.

the N*to*N *construction* (like in (1) and (2)), or near minimal pairs of the N*to*N *distractor* patterns (like in (5), (6), and (7)). Thus, this experiment tests the robustness of the original probing classifier when it is confronted with out of domain word orders that contain the same lexical cues as positive instances of the construction.

5.1 Results

Figure 2 shows the probe's performance on the perturbed test sets for the N*to*N *construction*. We see that in the very early layers (1-3), the probe often predicts the N*to*N *construction* despite the word order shifts, leading to relatively low accuracy. This possibly means that the classifier is biased by the lexical cues in the sentence early on. Interestingly, performance on *PN* and *PN* perturbations is substantially worse than performance on *NP* and *NNP* in the early layers. Accuracy on all perturbations trends upwards in the later layers, with reduction in training examples leading to drops in performance especially for *NP/NNP*.

5.2 Analysis

Overall, we find that classifier probes are able to distinguish instances of the N*to*N *construction* from both near minimal pairs (N*to*N *distractor* patterns) and artificial examples (perturbed word orderings). This finding aligns with the strong performance on form-based recognition that has been observed in previous work on other constructions (Li et al., 2022; Weissweiler et al., 2022; Mahowald, 2023). The peak in performance in the late-middle layers is consistent with much previous work on linguistic probing, which show that the middle and late-middle layers perform best for a variety of linguistic tasks (Goldberg, 2019; Hewitt and Manning, 2019; Lin et al., 2019; Liu et al., 2019).

The differences in the performance between the *NP/NNP* and the *PN/PNN* perturbed orderings is an unexpected finding. According to Rogers et al. (2021), the earlier layers of BERT encode "word order", while the middle layers are where syntactic capabilities emerge. Based on this logic, it is unsurprising that the classifier's ability to distinguish *PN/PNN* emerges in the middle and later layers. Why might the *NP/NNP* instances be distinguished so much quicker? Our intuition is that in general, preposition tokens probably attend more to their immediately following word than their immediately preceding word. This is because prepositions are often immediately followed by objects, while their syntactic governor may or may not be directly adjacent to them. Perhaps in the early layers of the model (before hierarchy is as explicitly represented) prepositions attend to their following token more quickly because this is a surface word order pattern that feeds quite well into syntax.

One alternative explanation is that *PN/PNN* may produce generally more grammatical sounding sentences than *NP/NNP*. For instance, (12) sounds much closer to a real sentence than (14). It could be that the classifier probe takes into account the ungrammaticality of *NP/NNP*, even though it was not explicitly trained to do this, since the classifier probe is only trained on grammatical sentences. How exactly the ungrammaticality is represented in these embedding representations is unknown, but provides one possible explanation for the differential performance of the perturbed word ordering patterns.

Having established that performance on identifying the N*to*N construction is strong, we now turn to the task of disambiguating the meaning of the construction within context.

6 Experiment 3: Semantic Disambiguation

6.1 N*to*N Subtypes

We have established that classifier performance is strong at identifying instances of the N*to*N construction relative to similar patterns. However, the construction itself is ambiguous, and can have different meanings in context. The two primary meanings are SUCCESSION and JUXTAPOSITION, which are shown in (3) and (4) respectively.

The two types co-occur with different nouns at different frequencies. The SUCCESSION subtype most often occurs with spatiotemporal nouns (e.g. *day to day* or *coast to coast*). On the other hand, the JUXTAPOSITION subtype most often occurs with body parts or humans (e.g. *face to face* or *friend to friend*). However, the noun meaning is not determinative, and within context some noun lemmas occur with the less common meaning. Furthermore, both constructions occur with rare noun lemmas for which it is not clear what type would be more common.

6.2 Methodology

In this section, we train a classifier to distinguish semantic subtypes of N*to*N. We focus on the two main subtypes that are well attested in the data: SUCCESSION and JUXTAPOSITION. We also include examples of the N*to*N *distractor* patterns which are not examples of the *construction*. Thus, the probe is faced with a 3-class classification problem: it must distinguish between the SUCCESSION subtype, the JUXTAPOSITION subtype, and non-examples of the construction (*distractors*). Following Hewitt and Liang (2019), we train *control classifiers* with a random label assigned to each lemma. If the probes are properly selective, the control classifiers should have accuracies of around 33 percent.

6.3 Results

Figure 3 shows the precision and recall scores of the semantic probing experiments. Across all semantic types, performance is generally high for the classifiers trained on the full split of data, with recall on all 3 classes near 80%, and strong performance even in the early layers. This is in contrast to some other semantic tasks, for which probes only reach their peaks in the mid to late layers of BERT.

Across all layers, both SUCCESSION and JUXTAPOSITION perform worse with only 10 training examples, but performance stabilizes after only 25 examples for the probe. The relatively low recall for JUXTAPOSITION and SUCCESSION when the classifiers are only trained with 10 examples indicates that the probe has not fully learned to correctly distinguish the two main semantic subtypes. It is somewhat striking that there is not a larger difference between SUCCESSION and JUXTAPOSITION in performance, given that SUCCESSION accounts for roughly 68% of all instances of the construction in our dataset. While probes are trained with balanced training sets, the relative frequency of these semantic subtypes within our dataset (and by extension COCA) is a strong indication that SUCCESSION is the more frequent meaning. Nevertheless, performance is roughly comparable between the two semantic subtypes. In all cases, the *distractor* class is overpredicted, leading to a relatively low precision compared to the subtypes of the construction. As expected, the control classifiers achieve roughly chance performance across layers, indicating that our probes have high selectivity. The GloVe-based baseline achieves an average recall of around .54 across the subtypes, but has widely variable performance depending on the semantic subtype. In general, the GloVe based classifier is much more likely to underpredict SUCCESSION, leading to very high precision and very low recall for this class.[3]

7 Related Work

There has been substantial research on investigating the linguistic information that is encoded by

---

[3] We report GloVe and control results using the full training set. Performance of the GloVe baselines degrades with fewer examples, while the control classifiers remain near chance.

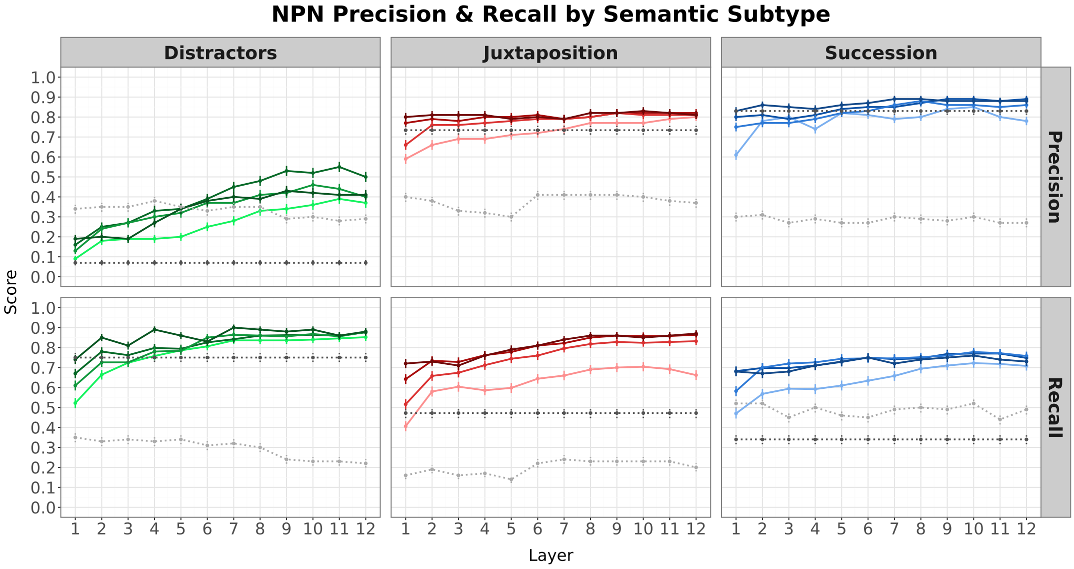


Figure 3: Precision and Recall of different semantic subtypes of NPN in 3-way classification. Lighter colors indicate fewer training examples, with possible values of 10, 25, 100, and 287 training examples per class. Classifiers trained with at least 25 per-class training examples begin to show strong performance across classes. JUXTAPOSITION takes substantially more training examples for classifiers to learn compared with SUCCESSION. Each line represents the average of 5 random seeds. Dotted lines represent baselines: GloVe (black) and control (gray). Error Bars indicate 95% confidence intervals over the average of the random seeds.

BERT. Much of this work has focused on syntactic structure (Hewitt and Manning, 2019; Jawahar et al., 2019; Liu et al., 2019; Hu et al., 2020), agreement phenomena (Lin et al., 2019) and semantics (Vulić et al., 2020; Chang and Chen, 2019; Ettinger, 2020), with the BLiMP (Warstadt et al., 2020) and SyntaxGym (Gauthier et al., 2020) providing key evaluation datasets. Belinkov (2022) and Elazar et al. (2021) provide critiques of the probing classifier methodology for its indirectness and susceptibility to spurious correlations. Various improvements on the methodology have been suggested, with a general focus on providing more controlled probing environments (Pimentel et al., 2020; Kim et al., 2022) and causal claims through counterfactuals (Ravfogel et al., 2021; Elazar et al., 2021). Of particular relevance to this work is Hewitt and Liang (2019), who propose the control classifier methodology as one methodology for controlling for spurious correlations in classifier performance. We believe our use of control classifiers and non-contextual baselines provide proper context for our probing results.

Earlier computational linguistic work on English trained classifiers for such grammatico-semantic phenomena as identifying argument structure constructions (Hwang and Palmer, 2015) and disambiguating functions of tense and definiteness (Reichart and Rappoport, 2010; Bhatia et al., 2014), as well as generally to disambiguate the senses of prepositions (Litkowski and Hargraves, 2007; Schneider et al., 2018). Tayyar Madabushi et al. (2020) were the first to investigate BERT's performance on learning constructions, finding that BERT is able to identify a large set of hundreds of automatically identified constructions. Regarding well-established argument structure constructions, Li et al. (2022) find that RoBERTa implicitly contains abstract knowledge of the constructions beyond specific lexical cues. Weissweiler et al. (2022) find that BERT-scale models are able to correctly distinguish the COMPARATIVE-CORRELATIVE construction from similar looking patterns, but find that the models fail on reasoning tests related to the construction's semantics. Mahowald (2023) finds that the larger GPT-3 model can provide acceptability judgments for the Article+Adjective+Numeral+Noun (AANN) construction which generally align with human judgements, and find that the model is sensitive to constraints on the slots in the construction. Chronis et al. (2023) test BERT's knowledge of the same AANN construction by projecting tokens in the construction into an interpretable embedding space, finding that features aligning with measure-words are evoked by tokens in the construction. Beyond BERT-scale models, Zhou et al. (2024), Bonial and Tayyar Madabushi (2024) and Scivetti et al. (2025) all test LLM knowledge of constructions in more complex scenarios, finding that their performance generally lags behind humans regarding construction understanding, though there is variation depending on the construction. Zhou et al. (2024) test a range of LLMs on understanding the CAUSAL-EXCESS constructions in comparison to constructions with highly similar forms, showing that the model is often misled by form-based cues. Their experiments most closely mirror our inquiries into construction sense disambiguation, though they disambiguate between similar but distinct constructions while we focus on a single polysemous construction. While Zhou et al. (2024) find that LLMs largely are unsucessful at meaning-based disambiguation, and Weissweiler et al. (2022) also find negative results regarding the semantics of the COMPARATIVE-CORRELATIVE, our relatively positive results on construction disambiguation in this present work demonstrate that for N*to*N, models may possess more robust models of constructional semantics than would be previously expected.

While NPN has not been the major focus of past analysis Weissweiler et al. (2024) do consider it as one of the constructions which they include in their UCxn dataset, which is compiled by automatically using Universal Dependencies (de Marneffe et al., 2021) graphs to find indications of constructions across 10 languages. We do not use this dataset due to its limited size (it contains under 50 total examples of the NPN construction in English).

8 Conclusion

In this work, we constructed a novel dataset of N*to*N construction by extracting all instances of the construction which we found in COCA. Using our dataset, we have probed BERT's knowledge of the N*to*N construction by training a linear probe to distinguish instances of the construction from near minimal pairs from corpus data. We show that a linear probe is largely able to distinguish true instances construction from naturally occurring *distractor* patterns, as well as from artificially perturbed versions of the construction, though the probe is more robust to recognizing the effect of some word order changes than others. Furthermore, we show that a BERT-based classifier can disambiguate the sense of the N*to*N construction in context, beyond the lexical semantic cues that are present. For both form- and meaning-based experiments, we show that the classifier results are robust even in the face of dramatic reductions in the number of training examples. This indicates that constructional knowledge is likely latently encoded within BERT and not due to spurious correlations learned by the classifiers. Overall, these results contribute to the growing body of evidence that LMs have some ability to acquire grammatical properties of rare and idiosyncratic constructions.

9 Limitations

This work is limited in several ways. Due to natural relative frequencies of various constructions, the dataset used for N*to*N is unbalanced between the N*to*N *construction* and *pattern*. This means that the training set for the classifier was quite small, because we ensured that training was balanced between the different classes. While the probing classifiers do achieve high accuracy, it is unclear how much accuracy is being capped by the limited data available. However, this fact, alongside our experiments with reduced training set sizes, indicate that the probes can learn with relatively little training signal.

This is experiment is also limited in only considering N*to*N, as opposed to the broader NPN construction. This is an intentional choice, as "to" has the most semantic subtypes of NPN associated with it. Future work is needed to see if the results here are robust to the inclusion of additional NPN examples with other lemmas into the dataset. We also only consider the English NPN construction, though the construction has been observed in a range of languages, including Dutch, English, French, German, Norwegian, Japanese, Mandarin, Polish, and Spanish (Weissweiler et al., 2024). We also limit our experiments to cases where the nouns match. This choice greatly simplifies our process of detecting true constructions as well as distractors, but also excludes some interesting examples of the construction, as pointed out by Jackendoff (2008).

Finally, this work utilizes the probing classifier methodology, which has been criticized for providing indirect/correlational evidence of linguistic information in LM representations (Belinkov, 2022). Future work is needed to broaden the analysis to include causal probing methodologies (e.g. AlterRep, Ravfogel et al. 2021; MaPP, Karidi et al. 2021; Reconstruction Probing, Kim et al. 2022).

References

Yonatan Belinkov. 2022. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219.

Archna Bhatia, Chu-Cheng Lin, Nathan Schneider, Yulia Tsvetkov, Fatima Talib Al-Raisi, Laleh Roostapour, Jordan Bender, Abhimanu Kumar, Lori Levin, Mandy Simons, and Chris Dyer. 2014. Automatic classification of communicative functions of definiteness. In *Proc. of COLING*, pages 1059–1070, Dublin, Ireland.

Claire Bonial and Harish Tayyar Madabushi. 2024. A Construction Grammar Corpus of Varying Schematicity: A Dataset for the Evaluation of Abstractions in Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, page 243–255, Torino, Italia. ELRA and ICCL.

Ting-Yun Chang and Yun-Nung Chen. 2019. What Does This Word Mean? Explaining Contextualized Embeddings with Natural Language Definition. 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)*, page 6064–6070, Hong Kong, China. Association for Computational Linguistics.

Tyler A. Chang and Benjamin K. Bergen. 2024. Language Model Behavior: A Comprehensive Survey. *Computational Linguistics*, 50(1):293–350.

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)*, page 242–261, Toronto, Canada. 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*, page 276–286, Florence, Italy. Association for Computational Linguistics.

William Croft. 2001. *Radical Construction Grammar: Syntactic Theory in Typological Perspective*. Oxford University Press.

Mark Davies. 2010. The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and Linguistic Computing*, 25(4):447–464.

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

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175.

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.

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*, page 70–76, Online. Association for Computational Linguistics.

Adele E. Goldberg. 1995. *Constructions: A Construction Grammar Approach to Argument Structure*. University of Chicago Press. Google-Books-ID: HzmGM0qCKtIC.

Yoav Goldberg. 2019. Assessing BERT's Syntactic Abilities. (arXiv:1901.05287). ArXiv:1901.05287 [cs].

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)*, page 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A Systematic Assessment of Syntactic Generalization in Neural Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 1725–1744, Online. Association for Computational Linguistics.

Jena D. Hwang and Martha Palmer. 2015. Identification of caused motion constructions. In *Proc. of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 51–60, Denver, Colorado.

Ray Jackendoff. 2008. "construction after Construction" and Its Theoretical Challenges. *Language*, 84(1):8–28.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What Does BERT Learn about the Structure of

Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3651–3657, Florence, Italy. Association for Computational Linguistics.

Taelin Karidi, Yichu Zhou, Nathan Schneider, Omri Abend, and Vivek Srikumar. 2021. Putting Words in BERT's Mouth: Navigating Contextualized Vector Spaces with Pseudowords. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, page 10300–10313, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Najoung Kim, Jatin Khilnani, Alex Warstadt, and Abed Qaddoumi. 2022. Reconstruction Probing. (arXiv:2212.10792). ArXiv:2212.10792 [cs].

Bai Li, Zining Zhu, Guillaume Thomas, Frank Rudzicz, and Yang Xu. 2022. Neural reality of argument structure constructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 7410–7423, Dublin, Ireland. Association for Computational Linguistics.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open Sesame: Getting inside BERT's Linguistic Knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, page 241–253, Florence, Italy. Association for Computational Linguistics.

Tal Linzen and Marco Baroni. 2021. Syntactic Structure from Deep Learning. *Annual Review of Linguistics*, 7(Volume 7, 2021):195–212. Publisher: Annual Reviews.

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.

Ken Litkowski and Orin Hargraves. 2007. SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. In *Proc. of SemEval*, pages 24–29, Prague, Czech Republic.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, page 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Kyle Mahowald. 2023. A Discerning Several Thousand Judgments: GPT-3 Rates the Article + Adjective + Numeral + Noun Construction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, page 265–273, Dubrovnik, Croatia. Association for Computational Linguistics.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121. Publisher: Proceedings of the National Academy of Sciences.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, page 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ludovica Pannitto and Aurélie Herbelot. 2023. CALaMo: a Constructionist Assessment of Language Models. In *Proceedings of the First International Workshop on Construction Grammars and NLP (CxGs+NLP, GURT/SyntaxFest 2023)*, page 21–30, Washington, D.C. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-Theoretic Probing for Linguistic Structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, page 4609–4622, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, page 101–108, Online. Association for Computational Linguistics.

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual Interventions Reveal the Causal Effect of Relative Clause Representations on Agreement Prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, page 194–209, Online. Association for Computational Linguistics.

Roi Reichart and Ari Rappoport. 2010. Tense sense disambiguation: a new syntactic polysemy task. In *Proc. of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 325–334, Cambridge, MA.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. Comprehensive supersense disambiguation of English prepositions and possessives. In *Proc. of ACL*, pages 185–196, Melbourne, Australia.

Wesley Scivetti, Melissa Torgbi, Austin Blodgett, Mollie Shichman, Taylor Hudson, Claire Bonial, and Harish Tayyar Madabushi. 2025. Assessing Language Comprehension in Large Language Models Using Construction Grammar. (arXiv:2501.04661). ArXiv:2501.04661 [cs].

Lotte Sommerer and Andreas Baumann. 2021. Of absent mothers, strong sisters and peculiar daughters: The constructional network of English NPN constructions. *Cognitive Linguistics*, 32(1):97–131.

Harish Tayyar Madabushi, Laurence Romain, Dagmar Divjak, and Petar Milin. 2020. CxGBERT: BERT meets Construction Grammar. In *Proceedings of the 28th International Conference on Computational Linguistics*, page 4020–4032, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yu-Hsiang Tseng, Cing-Fang Shih, Pin-Er Chen, Hsin-Yu Chou, Mao-Chang Ku, and Shu-Kai Hsieh. 2022. CxLM: A Construction and Context-aware Language Model. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, page 6361–6369, Marseille, France. European Language Resources Association.

Tim Veenboer and Jelke Bloem. 2023. Using Collostructional Analysis to evaluate BERT's representation of linguistic constructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, page 12937–12951, Toronto, Canada. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 7222–7240, Online. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The Benchmark of Linguistic Minimal Pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Leonie Weissweiler, Nina Böbel, Kirian Guiller, Santiago Herrera, Wesley Scivetti, Arthur Lorenzi, Nurit Melnik, Archna Bhatia, Hinrich Schütze, Lori Levin, Amir Zeldes, Joakim Nivre, William Croft, and Nathan Schneider. 2024. UCxn: Typologically informed annotation of constructions atop Universal Dependencies. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16919–16932, Torino, Italia. ELRA and ICCL.

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.

Leonie Weissweiler, Valentin Hofmann, Abdullatif Köksal, and Hinrich Schütze. 2022. The better your Syntax, the better your Semantics? Probing Pretrained Language Models for the English Comparative Correlative. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, page 10859–10882, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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*, page 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Gotlieb Wilcox, Richard Futrell, and Roger Levy. 2024. Using Computational Models to Test Syntactic Learnability. *Linguistic Inquiry*, 55(4):805–848.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. (arXiv:1910.03771). ArXiv:1910.03771 [cs].

Shijia Zhou, Leonie Weissweiler, Taiqi He, Hinrich Schütze, David R. Mortensen, and Lori Levin. 2024. Constructions Are So Difficult That Even Large Language Models Get Them Right for the Wrong Reasons. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, page 3804–3811, Torino, Italia. ELRA and ICCL.

Joost Zwarts. 2013. From N to N: The anatomy of a construction. *Linguistics and Philosophy*, 36(1):65–90.

Evidence of Generative Syntax in Large Language Models

Mary Katie Kennedy
Linguistics
University of Southern California
`mkkenned@usc.edu`

Abstract

The syntactic probing literature has been largely limited to shallow structures like dependency trees, which are unable to capture the subtle differences in sub-surface syntactic structures that yield semantic nuances. These structures are captured by theories of syntax like generative syntax, but have not been researched in the LLM literature due to the difficulties in probing these complex structures that have many silent, covert nodes. Our work presents a method for overcoming this limitation by deploying Hewitt and Manning (2019)'s dependency-trained probe on sentence constructions whose structural representation is identical in a dependency parse, but differs in theoretical syntax. If a pretrained language model has captured the theoretical syntax structure, then the probe's predicted distances should vary in syntactically-predicted ways. Using this methodology and a novel dataset, we find evidence that LLMs have captured syntactic structures far richer than previously realized, indicating LLMs are able to capture the nuanced meanings that result from sub-surface differences in structural form.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable language capabilities that have been steadily increasing ever since BERT (Devlin et al., 2019). This impressive performance has prompted a body of research interested in investigating *why* these models are so successful. From this came a subset of research seeking to understand what, if any, linguistic features or knowledge these models have acquired (Jawahar et al., 2019; Belinkov and Glass, 2019; He et al., 2024; Kallini et al., 2024) as means to better understand their language performance. The focus of linguistic inquiry can vary from the semantic (Nikolaev and Padó, 2023; Kamath et al., 2024) to the morphological (Coleman, 2020; Anh et al., 2024) to the syntactic (Clark et al., 2019; Chi et al., 2020; Kulmizev et al., 2020; Maudslay and Cotterell, 2021), the latter of which our current research seeks to extend.

Much of the research into syntactic representation in LLMs have utilized dependency parses to represent a sentence's syntactic structure (Hewitt and Manning, 2019; Maudslay and Cotterell, 2021; Tucker et al., 2022; Eisape et al., 2022; Buder-Gröndahl, 2024). However, these relatively shallow representations can fail to capture features of a sentence and the nuanced differences in meaning that result from different sub-surface syntactic structures. Adopting a dependency framework makes theoretical assumptions and imposes limitations on the richness of meaning that can be expressed, the consequences of which are often not addressed. These simple, compact representations stand in stark contrast to the deeper, hierarchically-complex structures that are posited in theoretical syntax, particularly the generative frameworks, which postulate these complexities in order to account for the difference in semantic meaning and syntactic patterning of certain syntactic constructions. Because the structures posed by theoretical syntax are far more complex with more tree nodes than words in the sentence, attempts to probe for generative syntactic structures have been stymied, and it remains unclear whether LLMs have captured these richer sub-surface structures.

Our work seeks to overcome this through our unprecedented application of Hewitt and Manning (2019)'s dependency-trained probe to test for theory-backed syntactic structures. To implement this, we identified two sentences structures—Subject Raising (SR) and Subject Control (SC)—whose surface and dependency representations are identical, but *whose generative structures **differ*** as their complement sizes differ (control predicates take larger complements than raising predicates). Using this method, we are able to circumvent the issue of handling empty nodes in the syntactic tree

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 377–396
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

as the probe only recovers dependency distances. Our hypothesis holds that if the probe predicts distances for the two sentence types that significantly differ in syntactically-predicted ways, then this is evidence that LLMs have captured the more complex structures of generative syntax, which can yield structures to explain the pattern variations and semantic nuances of certain constructions.

To test this, we developed a novel dataset of over 33,000 SR/SC sentences, which we fed into our dependency-trained probes using pretrained models of **BERT**, **RoBERTa**, **GPT2**, and **Qwen2.5**. When we probe the content words, our results find strong evidence of the SR/SC difference that aligns with SC taking a larger complement than SR as generative syntax proposes. However, probing the distances with infinitival "to" suggest that the functional word may encode syntactic structure in an aberrant manner. Together, our work suggests LLMs have encoded structure that is more complex than previously realized, and provides a novel method to probe for theoretical syntactic structure in LLMs.

2 Related Work

The impressive language abilities of recent LLMs have prompted researchers to ask whether this performance is due to some probabilistic modeling, or if these language models have managed to capture linguistic structures. To answer this question, a line of research known as **probing** was developed. This methodology feeds the model's contextualized vector representations into a neural network whose training objective is to predict a targeted linguistic structure from the representations alone (see Alain and Bengio, 2017 or Conneau et al., 2018 for example). The argument follows that if such a neural network probe is in fact able to predict the target pattern or structure, then it can be concluded that the language model has indeed implicitly learned that linguistic feature; otherwise, the probe task would have been doomed to failure.

This area of research has largely focused specifically on investigating whether models have learned to properly encode syntactic phenomenon (Mueller et al., 2020; Hu et al., 2020; Warstadt et al., 2020; Ravfogel et al., 2021; Davis et al., 2022). However, much of this structural syntactic research has relied on dependency parses as a means of representing syntactic structure (Hewitt and Manning, 2019; Chi et al., 2020; Maudslay and Cotterell, 2021; Tucker et al., 2022; Eisape et al., 2022), with one notable exception being Arps et al. (2022), which sought to (and largely succeeded) in training a probe to reconstruct a skeletal constituency tree. While this line of research is of value and great interest, there are theoretical assumptions made by using dependency parses, and there are limitations to using that particular syntactic framework.

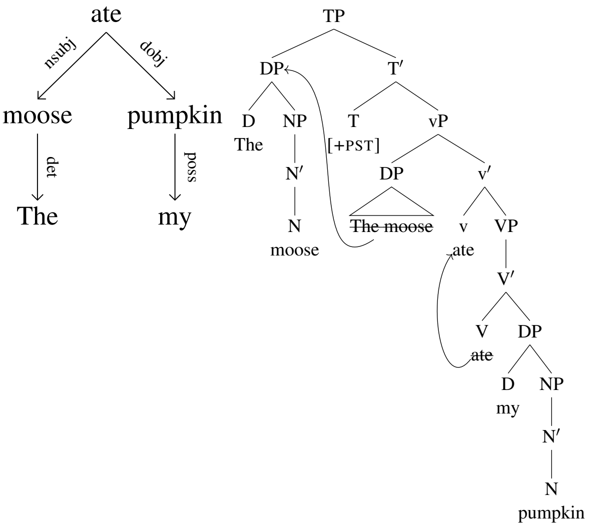


Figure 1: An example of the dependency tree (left) and generative syntax tree (right) for the sentence "The moose ate my pumpkin." Note how the dependency tree has a flatter structure with a one-to-one mapping of words to nodes in the tree. Compare this to the deeper generative tree where there are far more nodes in the tree than words in the sentence.

2.1 Syntactic Theories

Dependency parses derive from French linguist Lucien Tesnière's (1959) theory of syntax known as Dependency Grammar (DG), which focuses on the head-dependent relationship between words (see Figure 1). In these trees, each word can have one and only one incoming arc that indicates it is the dependent of its head, excepting the root of the sentence (often the matrix verb), which has no head.

DG trees are relatively flat structures with one-to-one mappings between words in the sentence and nodes in the tree. The appeal of such trees are largely three-fold: (1) the representations are compact and efficient due to the one-to-one mapping, (2) learning to parse a dependency tree is relatively easy once one understands the head-dependent relationships that exist, and (3) the dependency tree does not need to capture the sentence's linear order of words. The last factor makes DG an appealing theory for researchers working on languages with freer word-order (Müller, 2019); however, this "feature" can become a bug when it loses nuance or creates ambiguous parses (see Figure 2).

An alternative, more structurally-rich approach to syntax has built off the theories of Chomsky (1957; 1981; 1986; 1995) and others who have refined this phrase-structure (also known as a constituency-based) framework to build up the syntactic framework known as generative syntax (GS). This family of syntactic theories are built on the X-bar theory, which proposes the operations *External Merge* and *Internal Merge* (formerly known as "Move"), and stipulates that nodes are binary-branching and that every phrase has a head (Chomsky, 1995). After all operations are applied in the course of derivation, the end result is the linearization of the sentence when read from left to right along the children nodes.[1] The generative framework is concerned with identifying the operations and rules that together generate licit sentences but do *not* generate illicit constructions.

Unlike DG, generative syntax and other phrase-structure grammars thus yield deeper, more complex trees with hierarchical structures and phonologically null nodes whose presence must be deduced through testing. While this complexity is well-warranted (in that it can generate sentences that are grammatical and explain what causes ungrammaticality), the tree's size and complexity creates a complicated and unwieldy structure that is difficult for non-linguists to implement.

The result of this has been a limitation in the scope of feasible GS research in Natural Language Processing (NLP). Even work that has sought to test for the deeper, more complex phrase-structures in NLP has largely focused either on only seeking to recover a phrase's boundaries (Tenney et al., 2019; Kallini et al., 2024) or has otherwise trained their probe on the overly-simplified, *n*-branching constituency trees of the English Penn Treebank (PTB) (Marcus et al., 1993), which are only annotated with “skeletal” syntactic structure schema that is relatively atheoretical. The PTB's annotation is often used for automatic conversion into a dependency parse with little issue since the simplicity of PTB syntax is non-problematic for the simpler structures and principles of an NLP dependency parse. However, the result of this is a corpora that is not theoretically-sound for many of the deeper linguistic inquiries into phrase-structure grammars.

(1) Do you know who stole the Crown Jewels?

(2) Who do you know stole the Crown Jewels?

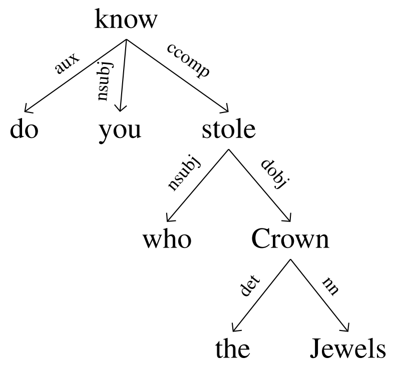


Figure 2: Above are two different sentences that yield *identical* dependency parses. While similar, the sentences have different meanings (imagine you are being questioned about the theft of the Crown Jewels: the first question merely seeks to inquire whether or not you know who the thief is, while the latter presumes you know who the thief is and seeks to learn the identity). This illustrates both the ways in which dependency trees do not capture linear order, and highlights some of the limitations of dependency trees.

Because of this, it has yet to be discovered whether LLMs have managed to capture the deeper, hierarchical structures of GS. However, there are three major barriers to testing whether models have captured the richer hierarchical structures as proposed by generative frameworks:

1. GS and similar frameworks often have "empty" nodes that are not overtly realized.[2] As such, they are not overtly present in the texts LLMs train on, and so probing at their presence is difficult because it raises the question: how can you probe at something that is not overtly represented?

2. Probes largely require a gold tree that indicates the correct structure or parse. Human annotation, while crucial when handling such

---

[1] In earlier versions of GS, it was argued that when an element moves, it leaves behind a coindexed trace element (Chomsky, 1973; Fiengo, 1977). Because such an derivation introduces a *new* element, Chomsky (1993) revised the approach to the Copy Theory of Movement, claiming that movement leaves behind a copied element that is not phonologically realized. Thus, the sentence used above would be "Do you [do] [you] know [know] who [who] [who] stole [stole] the Crown Jewels?" For a discussion on which elements are phonologically realized and why, see Corver (2007).

[2] This can be due to movement (see Footnote 1) or the feature not having an overt representation (e.g., there is no specific word or morpheme that indicates present tense for plural subjects as in "They $walk_{pres}$ to the store"). See Figure 1 for demonstration.

fine-grained analysis, is laborious and costly in resources and time.

3. Even if one is able to secure the resources necessary to create such a gold standard, there are competing theories even within the generative framework that would change a sentence's representation. As such, a gold parse would be subject to great theoretical scrutiny and likely face present or future dissenting opinions.

Our research develops a method that circumvents these obstacles while still addressing the fundamental research question of whether LLMs have captured something of the deeper, sub-surface syntactic representation theorized by many linguists.

3 Methodology

3.1 Probing Method

To combat the first issue of accounting for structures that are not phonologically realized, we have opted for the novel approach of re-purposing the original Hewitt and Manning (2019), which was trained to recover dependency trees, to investigate whether LLMs have encoded theoretically-motivated generative phrase-structure trees.

The structural probe developed by Hewitt and Manning (2019) proposes a model M that encodes a sequence of vector representations $\boldsymbol{h}_{1:n}^{l}$ from an input sequence of n words $\boldsymbol{w}_{1:n}^{l}$ where l identifies the sentence index. From there, they define a linear transformation matrix $\boldsymbol{B} \in \mathbb{R}^{k \times n}$ to parameterize the parse tree-encoding distances:

$$d_B(\mathbf{h}_i^l, \mathbf{h}_j^l)^2 \quad = \quad (B(\mathbf{h}_i^l - \mathbf{h}_j^l))^T(B(\mathbf{h}_i^l - \mathbf{h}_j^l))$$

where i and j are the words in the sentence and where the transformation matrix B's objective is to reproduce the gold parse distances between each pair of words (w_i^l, w_j^l) for all sentences l in the parsed training corpus T^l. The training uses a gradient descent objective:

$$\min_{\boldsymbol{B}} \sum_{l} \frac{1}{\mid s^l \mid^2} \sum_{i,j} \mid d_{T^l}(w_i^l, w_j^l) - d_B(h_i^l, h_j^l)^2 \mid$$

In this equation, $\mid s^l \mid$ is the length of sentences, which the function normalizes using the square of the sentence's length as each sentence contains $\mid s^l \mid^2$ pairs of words. The probe's objective thus seeks to approximate a matrix of distances that most closely resembles the gold-standard distances. Because Hewitt and Manning (2019) use a dependency parse, gold-standard parses were converted into gold-standard distance matrices where distances are defined such that the distance between a parent node and its child nodes is 1, the distance between a child node and its grandparent node is 2, the distance between a child node and a so to speak "aunt" or "uncle" node is 3, and so on and so forth. Evaluation of the probe involved calculating the minimum spanning tree for each sentence's predicted distances to derive the sentence's predicted undirected, unlabeled attachment score (UUAS) compared to the gold tree, and the average Spearman correlation of the predicted matrix of distances compared to the gold-standard matrix.

We chose this method specifically *because* it is a probe trained *only* to capture dependency parses with their one-to-one mappings between a sentence's words and a tree's nodes. Though there are critical limitations to dependency parses as discussed in Section 2.1, we argue that its simplicity and overgeneralization can in fact be converted into a benefit. It is because the probe is superficially only supposed to capture shallow-level, generalized syntactic structures that we can use the method to tease apart syntactic structures whose representations are **identical** in a *dependency* parse but **vary** in a *generative* framework.

3.2 Syntactic Structures of Interest

Our method hinges on testing syntactic structures whose representations are crucially different in generative accounts, but are invariant in a dependency parse. In doing so, we propose turning the limitations of a dependency probe to an asset. If the probe's predicted dependency distances vary between the sentences in question in ways that align with generative theoretical predictions, then we have evidence that not only do LLMs' contextualized vector representations capture generative syntactic structures, but that a probe trained only to recover dependency parses is additionally sensitive to hierarchical phrase-structure distances.

To test this, we have selected the well-researched Subject Control (SC) and Subject Raising (SR) constructions as our experimental condition. Observed first by Rosenbaum (1967), SR constructions are those that consist of two clauses: a matrix clause and an infinitival Tense Phrase (TP) complement. Since its initial observation by Rosenbaum (1967), it's largely been accepted that the subject position of the embedded clause is occupied by a trace element (later revised to a copy element, see footnote 1) due to the subject being raised into the matrix clause by the EPP features[3] in the matrix clause. SC constructions, meanwhile, are assumed to take a larger complement than a raising verb, with many typically assuming an SC complement to be a Complement Phrase (CP). Many theories follow Chomsky and Lasnik (1993) and posit a silent PRO element that is co-indexed with and controlled by the matrix's subject. This PRO receives its theta-role from the embedded verb while the matrix subject receives its theta-role from the matrix verb, thus satisfying the Theta Criterion (Chomsky, 1957).[4]

For the purposes of our experiment, the crucial things to know are: Subject Raising takes a Tense Phrase (TP) as its complement, while Subject Control takes the larger Complement Phrase (CP) as its complement, which inherently contains a TP itself. Thus, the result are two structures whose surface forms and dependency parses are *identical*, but whose hierarchical syntactic representations are **different**. Thus, we would expect that if the LLM has not acquired any knowledge of deeper syntactic representations or if the dependency-trained probe is insensitive to phrase-structure representations, then the probe's predicted distances between relevant word-pairs should *not differ* between the two structures. However, if such hierarchical representations are indeed captured and if the probe is sensitive to these structures, then we would anticipate that the distances between certain word-pairs in an SC construction are **longer** than the equivalent word-pairs in an SR construction due to SCs containing the larger CP complement as opposed to the smaller TP complement of SR predicates.

4 Experiments

4.1 Generating Data

For our experiment, we identified 6 SR verbs and 6 SC verbs, which we permutationally paired with a set of 8 subject words, 61 embedded verbs, and a set of possible direct objects (either a single pronominal direct object or a two-word definite object that was matched to a specific embedded verb). Thus, we yielded 33,120 unique sentences, such as "They wanted/seemed to annoy him."

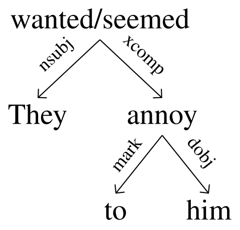


Figure 3: Dependency parse for the two sentences "They seemed/wanted to annoy him." The two trees are identical, and the distance between the subject and embedded verb is 2 while the distance between the subject and the infinitive or direct object is 3. This is true even if one were to use extended Universal Dependencies, which also conflates SC and SR verbs.

Metrics Should the LLMs *not* have any awareness of the deeper hierarchies or should the probe be insensitive to such differences, then should be *no difference* between SR's and SC's distances between words in the matrix clauses and words in the complement clauses. However, if such structures are captured and if the probe is sensitive to this, then we anticipate that the distance between a word in the matrix clause and a word in the complement clause will be **longer** in an SC construction compared to an SR construction since the CP complement is larger (see Figure 4).

For this reason, we opted to investigate the probe's predicted distances between the following word-pairs: subject and the infinitive (`subj-infin`, e.g., "they" and "to"), subject and the embedded verb (`subj-embed`, e.g., "they" and "annoy"), subject and the direct object (`subj-dobj`, e.g., "they" and "him"), and lastly, embedded verb and the direct object (`embed-dobj`, e.g., "annoy" and "him"), which serves as our baseline. We should acknowledge at this point that excepting our baseline comparison, none of our word-pairs have any direct dependent or syntactic relationship to each other. This is not a problem. Recall that the probe was trained on a the gold parses for dependency trees where the distance between two nodes can be counted as the number of edges between the two. Because of this design, we are able to probe the distances between the words in the matrix clause and the words in the complement clause despite there being no direct dependency or syntactic relationship.

As the dependency parses do not differ between

---

[3] Chomsky (1995) proposed the Extended Projection Principle, which stipulates that Tense bears a strong D-feature that requires a subject in its Specifier. This can be satisfied by either moving the subject to Spec,TP or by inserting an expletive like "it."

[4] For further discussion on Subject Control and Subject Raising and their structural and semantic differences, see Appendix A.

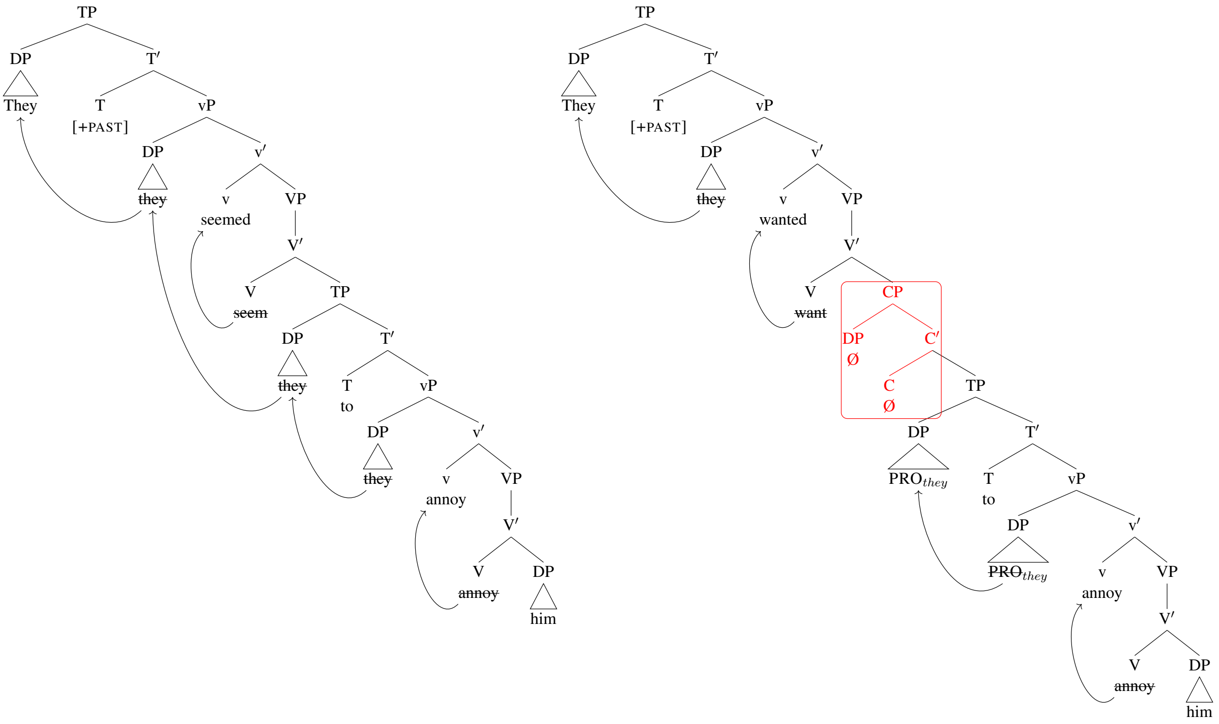


Figure 4: Syntactic trees of the SR sentence "They seemed to annoy him" (left) and the SC sentence "They wanted to annoy him" (right). The two structures are nearly identical, except SC contains a CP above the TP (red), which makes the hierarchical distances between the subject and the embedded clause's elements (i.e., the infinitive "to", the embedded verb "annoy", and the direct object "him") longer in the SC sentence.

the two structures, the gold-parse distances are also stable: `subj-embed` has a dependency distance of 2 while `subj-infin` and `subj-dobj` have dependency distances of 3 (see Figure 3). For this reason, if the LLMs do not capture generative syntactic hierarchies or if the probe is insensitive to such differences, then we should see *no difference in predicted distances between the two experimental conditions*. If, however, the models **do** capture this deep structural difference and if the probe is an adequate tool to measure this, then we should anticipate that the SC distances should be **longer** than their equivalent SR distances. To verify that our probe is working as anticipated, we included the baseline word-pair `embed-dobj`, which should *not* show any differences in distances as these words are *not* affected by the SC/SR distinction.

4.2 Experimental Setup

Models We probed three pre-trained Transformer (Vaswani et al., 2017) models: **BERT** (Devlin et al., 2019), **RoBERTa** (Liu et al., 2019), **GPT-2** (Radford et al., 2019), and **Qwen2.5** (Team, 2024). We constrained our probing to models with hidden dimensions of 768 and 1024, which corresponded to the `bert-base-cased`, `roberta-base`, and `gpt2-small` for the smaller models and `bert-large-cased`, `roberta-large`, and `gpt2-medium` for the larger models. Per the suggestions of our reviewers, we also included two newer models: `qwen2.5-0.5` and `qwen2.5-1.5` (referred to in this paper as "Qwen2.5Baby" and "Qwen2.5Small", respectively). All models were accessed using the Huggingface Transformers library (Wolf et al., 2020), and the probe was developed using the parsing train/dev/test splits of the Penn Treebank (Marcus et al., 1993).

Following Hewitt and Manning (2019),[5] a probe was trained to convergence (maximum of 40 epochs) on each layer with a batch size of 20. Analysis was conducted on the best-performing layer.

Once the best-performing layer[6] was selected, we fed our novel dataset to that probe and obtained the predicted distances for our word-pairs of interest. Analysis was conducted on the predicted

---

[5] Hewitt and Manning (2019)'s original code can be found at `https://github.com/john-hewitt/structural-probes`, which includes the **BERT** models. A helpful starting point to modify the code for **RoBERTa** and **GPT2** can be found at `https://github.com/leoier/structural-probes`.

[6] See Appendix C, Figures 5, 6, and 7 for model performances.

distances for our word-pairs *if and **only** if* the probe properly established the necessary dependency relationships. That is to say, if the probe *misparsed* the tree in a relevant manner, that word-pair's predicted distance was excluded from analysis. Using Figure 3 as a gold-parse, if the probe's minimum spanning tree situated "him" as a dependent of "wanted," then we excluded the `subj-dobj` word-pair as the tree was misparsed in a critical way for that word-pair. We currently do not have strong reason to suspect that such a misparse would affect non-impacted word-pairs; therefore, the `subj-embed` and `subj-infin`'s predicted distances would still be used for analysis since the probe would have correctly parsed the subject and embedded verb as being dependents of "wanted" and parsed the infinitive as the dependent of "annoy."[7]

5 Results

As mentioned, we generated 33,120 sentences for which we gathered a total of 935,419 distances across our four word-pairs and all eight language models. Overall, this represents an 88.05% accuracy score for correctly parsed word-pairs. The accuracy for our four word-pairs can be found in Table 1 where we may observe that while the accuracy for the SC condition is slightly higher, both showed high accuracy with the lowest being attributable to the `subj-dobj` word-pair, which was due to the direct object not being tied to the embedded verb, hence the equivalent scores with `embed-dobj`.

Due to the large size of the data, we split the data by word-pair for statistical analyses. Mixed effect models were developed with the lmer function from lme4 (v. 1.1-31) (Bates et al., 2015) and lmerTest (v. 3.1-3) (Kuznetsova et al., 2017). Fixed effects were identified as the condition (SC or SR) and the linear distance (the number of intervening words plus 1 to avoid issues of 0 multiplication) as well as their interaction. The latter two only applied to word-pairs with the direct object as the direct object could be a single pronominal like "it" (in which case the linear distance would be 1) or a full nominal phrase like "the car" (in which case the linear distance would be 2). For the other word-pairs, there was no variation in linear distance, hence its exclusion as a fixed effect. Condition was contrast-coded with SC being -0.5 and SR being 0.5.

Model comparison via anova was used to determine the random effects of by-MatrixVerb, by-SubjectWord, by-EmbedVerb, by-ObjectWord, and by-LanguageModel as well as random slopes for Condition (and LinDist for `dobj`). For the full linear models, see Appendix B.

To recap, our hypothesis is that the probe's predicted distances between the matrix subject and elements in the embedded clause (i.e., the infinitive, the embedded verb, and the direct object) should be **longer** in the SC condition compared to the SR condition. Should this be the case, this effect should appear in all of our word-pairs (excepting our baseline of `embed-dobj`). In this regard, our study uses conjunction testing in that we require all tests be significant in order to reject the null hypothesis (Weber, 2007). We thus follow Rubin (2021) and do not adjust our alpha level.

Table 1 reports our results where we find a main effect for Condition in our `subj-embed`, `subj-infin`, and `subj-dobj` data ($p = 2.77e^{-5}$, $p = 0.035$, and $p = 1.90e^{-12}$). Thus we find evidence that the predicted distances from an SC construction are significantly longer than an SR construction when considering the distance between the subject and a word within the embedded clause.

Crucially, we do **not** find Condition to be a significant predictor for our baseline, suggesting the probe is not spuriously attributing higher distances to SCs than SRs in ways that are not predicted by the syntax. However, interaction between Linear Distance and Condition *is* found to be a main effect for `embed-dobj`. To conduct follow-up models to investigate this result, we split the data by linear distance, meaning sentences were grouped into those that took a pronominal direct object such as "it" (linear distance of 1) and those that took a nominal phrase object such as "the car" (linear distance of 2). In doing so, we do **not** find Condition to be a main effect in *either* group.

Analysis of the data further reveals that linear distance decreases the predicted distance for subject raising verbs only. When we control for linear distance by splitting up the data by direct object type, though, our follow-up analyses find that the predicted distance between the embedded verb and the direct object do not significantly vary between

---

[7] For our current study, the best-performing layer was selected as the probe with the highest UUAS. As such, we opted only to use word-pairs in which the minimum spanning tree established the correct necessary dependencies for the word-pair in question. However, future work may investigate selecting the probe based on the Spearman correlation, in which case, the motivation to reject data based on improper parses disappears as the Spearman metric does not utilize the minimum spanning tree and instead seeks to globally reduce the differences between the gold distances and the predicted distances.

				Fixed Effects					
WordPair	**Condition**	**Acc**	**PredDist** *(avg)*	Coefficient	$\hat{\beta}$	SE($\hat{\beta}$)	df	t	p
Subj-Embed	Cont	96.01%	2.04	(Intercept)	1.93655	0.10610	13.24239	18.253	8.99e-11
	Raise	94.35%	1.82	Condition	-0.22879	0.04147	18.53383	-5.517	**2.77e-05**
Subj-Infin	Cont	89.50%	2.86	(Intercept)	2.81437	0.07561	13.77291	37.221	3.21e-15
	Raise	84.75%	2.74	Condition	-0.13397	0.05517	10.25726	-2.428	**0.035**
Subj-Dobj	Cont	86.29%	2.98	(Intercept)	2.823	0.3260	24.68	8.659	6.01e-09
	Raise	84.45%	2.76	Condition	-4.805e-01	4.309e-02	31.38	-11.152	**1.90e-12**
				LinDist	2.572e-03	6.602e-02	22.62	0.039	0.969
				Interaction	6.023e-02	5.436e-03	2.260e+05	11.081	**<2e-16**
Embed-Dobj	Cont	86.29%	1.50	(Intercept)	1.54498	0.08508	75.04190	18.160	<2e-16
(baseline)	Raise	84.45%	1.50	Condition	-0.04512	0.02990	41.13811	-1.509	0.138948
				LinDist	-0.04085	0.05312	44.15045	-0.769	0.445915
				Interaction	0.03901	0.01095	48.82859	3.561	**0.000835**

Table 1: Results of the probes' predicted squared Euclidean distances between the word-pairs of interest. Accuracy records what percentage of the sentences properly established the necessary dependency relationships for that particular word-pair. The right side of the table reports the fixed effects findings for the linear mixed-effect models that were built for each word-pair. See Appendix C, Figures 8 and 9 for visuals.

the two conditions ($p = 0.816$ for pronominal direct objects and $p = 0.238$ for nominal phrase objects).

The significantly longer predicted distances of the SC condition in `subj-embed`, `subj-infin`, and `subj-dobj`, paired with Condition *not* being a significant predictor for our baseline comparison of `embed-dobj` (even when accounting for interaction effects), together show strong evidence to reject our null hypothesis.

5.1 Results by LLM

While our results indicate that the probe is sensitive on some level to syntactic hierarchies, this is not equally true for all word-pairs across all models. As can be seen in Table 2 in Appendix C, a significant main effect for the SC/SR Condition for word-pairs `subj-embed` and `subj-dobj` was found for the probes of all models *except* `gpt2-medium`, which revealed Condition to be marginal (p = 0.0818), though the reason for this is unclear. As for `subj-infin` however, Condition was significant *only* for the `Qwen2.5-1.5` model (p = 0.0419), and `roberta-large` (p = 0.00345).[8] For all other models, Condition was *not* a significant predictor.

If the structures proposed by generative syntax to account for Subject Raising and Subject Control are indeed captured by LLMs and subsequently by the probe, then we would anticipate that *all* three word-pairs of interest across *all* LLMs should show significantly longer predicted distances in the SCs compared to SRs while the baseline comparison of `embed-dobj` (which is *not* affected by an SC or SR construction) should not. While these predictions are largely borne out by `subj-embed` and `subj-dobj` along with our baseline of `embed-dobj`, it does not hold true for `subj-infin` for many models.

This finding is particularly puzzling. Should it be that the LLMs do not capture the SC/SR distinction, then *none* of the word-pairs should have significant differences in distances rather than just one (namely, `subj-infin`). Additionally, there is no theory in any school of syntax (generative or otherwise) we are aware of that suggests SC verbs take larger complements *below* the TP head of "to." We might then posit that the infinitive's seemingly imperviousness to the SC/SR distinction may arise from these LLMs somehow building a novel and alien structure in which the infinitival "to" sits in the matrix clause while the complement size distinctions are displayed beneath it. Again, however, we resist this notion as we know of no theory postulating such an arbitrary and alien structure.

It is evident this matter requires further investigation, but it is possible the aberrant behavior of the infinitive is due to the nature of infinitives themselves. Infinitival "to" is semantically vacuous: there is little to any rich semantic meaning to the word, which is entirely functional in nature—denoting either non-finiteness as an infinitive[9] or directionality or telicity as a preposition. For this reason, we suspect the lack of semantic-richness of purely functional words may impact how structure is captured by embedding vectors.[10]

---

[8] For `gpt2-small`, Condition was marginal (p = 0.0688).

[9] See (Satik, 2022) for discussion on the subtle semantic differences between different types of infinitives.

[10] We exempt pronouns from this hypothesis. Our dataset subjects were pronominal and our single-word direct objects were also pronouns. Unlike infinitival "to," pronouns pick out referents in the real world, and can furthermore carry

We also cannot attribute the lack of significant findings amid `subj-infin` to model complexity or novelty. While the model with the highest number of parameters (`qwen2.5-1.5` at `1.54B` parameters) did find Condition to be a significant predictor for `subj-infin`, so did the `340m` parameter `roberta-large`. Despite this, models with similar parameter sizes as `roberta-large` did *not* find the SC/SR distinction to significantly predict the distance for `subj-infin`. Nor can we suggest that it is the newer models whose embedding representations capture linguistic aspects that correlate to syntactic hierarchy; the 2024 `Qwen2.5-0.5` failed to find Condition to be significant for `subj-infin`. Further research is needed done to understand why only a select few models' (`Qwen2.5-1.5` and `roberta-large`) embedding representations for infinitives appear to capture linguistic aspects corresponding to syntactic hierarchies.

6 Discussion

While the matter of infinitives remains murky, our findings suggest that models are capable of encoding some linguistic aspects corresponding to the syntactic hierarchies as proposed in generative syntax. That SC verbs yield longer predicted Euclidean distances than SR (as opposed to the reverse) already aligns with generative theories that control verbs take larger, more complex complements (Chomsky and Lasnik, 1993; Landau, 2007, 2013, 2024) than raising verbs.

It may be asked whether these results are merely a product of semantics rather than syntax. However, we maintain that seeking to entirely divorce syntax from semantics should not be the main goal. As Leivada and Murphy (2021) comments, "syntax, semantics, and the other levels of linguistic analysis are not undecomposable modules that work autonomously," which makes it difficult to separate the two when researching the neural processing of the human mind, and, we argue, when researching the artificial neural processing of an LLM. Our findings may be due to hierarchical distance being larger in SC, or it may be due to the generative syntactic theory that SCs assign an extra theta role(Chomsky and Lasnik, 1993; Landau, 2024; Beockx and Hornstein, 2010). Both explanations speak to LLMs being able to encode deeper linguistic aspects that interface with syntactic structure.

In order to determine if these findings are in fact indicative of syntactic hierarchical distance or merely a quirk of the SC/SR constructions, future work should aim to test other syntactic structures. Preliminary work by Kennedy (2025) tests *wh*-extraction from different sized complements (e.g., "What did she see him eat" vs "What did she expect him to eat" vs "What did she think he ate") and finds that Hewitt and Manning (2019)'s probe's predicted distance between the extracted *wh*-word and its embedded verb (e.g., "What" and "eat") increases as the size of the complement increases. With continued research like this, should multiple different sentence structures all converge on larger syntactic hierarchical distances yielding longer predicted probe distances, then we can say with even greater confidence that LLMs are capable of encoding linguistic attributes that correspond to the structures propose by generative syntax.

7 Conclusion

The implications of this work have impact on both the field of NLP and the field of linguistics. Our work suggests that LLMs have learned to capture elements of deeper and more complex syntactic structures within their embeddings than previously realized and thus have the ability to capture the semantic nuances that result from sub-surface structural differences. Our findings therefore further the interpretability research of LLMs to discover what these models have actually learned regarding the features and structures of language. We also find evidence that neural networks trained using the dependency framework can still capture deeper syntactic features, suggesting these simpler representations may be adequate for downstream tasks as they appear to be capable of reaping the benefits of deep structure without needing to explicitly train on deep structure. As for linguists, the findings of our work warrant further investigation into the viability of using language models as a means to test syntactic structures. Our work begins to open up the possibility of utilizing LLMs as another source of data to help augment, build, and perhaps even test syntactic theories.

Taken together, we situate our work as a realization of Linzen (2019) and Futrell and Mahowald (2025)'s claim that the skillsets and knowledge of the fields of NLP and linguistics complement each other, and that the two stand primed to advance each other's respective fields through collaboration.

---

information such as Case, Gender, and Number as opposed to infinitival "to," which indicates non-finiteness only.

Limitations

Our work still faces limitations in that it does not enable a full reconstruction of the hierarchical syntactic tree. This is a limitation currently inherent to the data and format of LLMs. As can be seen in Figure 4, the generative syntax trees consist of branches and nodes that do not overtly appear in the final derivation. That is to say, trace nodes/moved elements are not surfaced, nor are all syntactic elements (such as tense) realized by a separate word. Because of this, an LLM's contextualized word embeddings cannot currently be used to directly derive the sub-surface syntactic trees. The methodology that we've deployed allows us to probe for behaviors that would indicate that LLMs have captured more complex, hierarchically-rich structural information within their embeddings, but this cannot be directly shown the way Hewitt and Manning (2019) did with the one-to-one mappings of dependency parses. Thus, our work is still largely in the tradition of much of linguistics. We cannot directly observe people's mental grammars, but we probe for their knowledge and structures using measurements that indicate how people process and produce language. Similarly, our use of Hewitt and Manning (2019)'s probe also provides an apparatus to measure behaviors that we can use to reverse-engineer the possible behaviors and mechanisms that would derive such results. The interpretability question of LLMs is not far at all from the research questions of linguistics.

Acknowledgments

I would like to thank Khalil Iskarous, Jon May, Jesse Thomason, Andrew Simpson, and Travis Major for their informative discussions that have helped to enrich this research. I would also like to acknowledge and thank the reviewers whose feedback has lead to experimentation with newer models and a deeper engagement with the results.

References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *The 5th International Conference on Learning Representations*.

Dang Anh, Limor Raviv, and Lukas Galke. 2024. Morphology matters: Probing the cross-linguistic morphological generalization abilities of large language models through a wug test. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 177–188, Bangkok, Thailand. Association for Computational Linguistics.

David Arps, Younes Samih, Laura Kallmeyer, and Hassan Sajjad. 2022. Probing for constituency structure in neural language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6738–6757, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Cedric Beockx and Norbert Hornstein. 2010. *Control as Movement*. Cambridge University Press.

Tommi Buder-Gröndahl. 2024. What does parameter-free probing really uncover? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 327–336, Bangkok, Thailand. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Noam Chomsky. 1957. *Syntactic Structures*. Mouton.

Noam Chomsky. 1973. Conditions on transformations. In *A Festschrift for Morris Halle*. Hole, Rinehard Winston.

Noam Chomsky. 1981. *Lectures on government and binding*. Foris Publications.

Noam Chomsky. 1986. *Knowledge of Language: Its Nature, Origin, and Use*. Praeger.

Noam Chomsky. 1993. A minimalist program for linguistic theory. In Kenneth Locke Hale and Samuel Jay Keyser, editors, *The View From Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*. MIT Press.

Noam Chomsky. 1995. *The Minimalist Program*. MIT Press, Cambridge, MA.

Noam Chomsky and Howard Lasnik. 1993. Syntax: An international handbook of contemporary research. In Joachim Jacobs, Arnim von Stechow, Wolfgang Sternefeld, and Theo Vennemann, editors, *The theory of principles and parameters*. De Gruyter.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.

Haley Coleman. 2020. This is a BERT. now there are several of them. can they generalize to novel words? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 333–341, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

N.F.M. Corver. 2007. From trace theory to copy theory. In *The Copy theory of movement*, pages 1–10, Netherlands. John Benjamins.

Mark Davies. 2008–. The corpus of contemporary american english (coca).

Christopher Davis, Christopher Bryant, Andrew Caines, Marek Rei, and Paula Buttery. 2022. Probing for targeted syntactic knowledge through grammatical error detection. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 360–373, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tiwalayo Eisape, Vineet Gangireddy, Roger Levy, and Yoon Kim. 2022. Probing for incremental parse states in autoregressive language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2801–2813, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Robert Fiengo. 1977. On trace theory. *Linguistic Inquiry*, 8:35–61.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *Preprint*, arXiv:2501.17047.

Linyang He, Peili Chen, Ercong Nie, Yuanning Li, and Jonathan R. Brennan. 2024. Decoding probing: Revealing internal linguistic structures in neural language models using minimal pairs. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4488–4497, Torino, Italia. ELRA and ICCL.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Gaurav Kamath, Sebastian Schuster, Sowmya Vajjala, and Siva Reddy. 2024. Scope ambiguities in large language models. *Transactions of the Association for Computational Linguistics*, 12:738–754.

Mary Kennedy. 2025. Evidence of hierarchically-complex syntactic structure within BERT's word representations. In *2025 Meeting of the Society for Computation in Linguistics*.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Alexandra Kuznetsova, Per B. Brockhoff, and Rune H. B. Christensen. 2017. lmerTest package: Tests in linear mixed effects models. *Journal of Statistical Software*, 82(13):1–26.

Idan Landau. 2007. *New Horizons in the Analysis of Control and Raising*, chapter Movement-Resistant Aspects of Control. Springer.

Idan Landau. 2013. *Control in Generative Grammar: A Research Companion*. Cambridge University.

Idan Landau. 2024. *Elements in Generative Syntax*, chapter Control. Cambridge University.

Evelina Leivada and Elliot Murphy. 2021. Mind the (terminological) gap: 10 misused, ambiguous, or polysemous terms in linguistics. *Ampersand*, 8:100073.

Tal Linzen. 2019. What can linguistics and deep learning contribute to each other? response to pater. *Language*, 95(1):e99–e108. Publisher Copyright: © 2019, Linguistic Society of America. All rights reserved.

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

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rowan Hall Maudslay and Ryan Cotterell. 2021. Do syntactic probes probe syntax? experiments with jabberwocky probing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 124–131, Online. Association for Computational Linguistics.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Stefan Müller. 2019. *Superseded: Grammatical theory*. Number 1 in Textbooks in Language Sciences. Language Science Press, Berlin.

Dmitry Nikolaev and Sebastian Padó. 2023. Investigating semantic subspaces of transformer sentence embeddings through linear structural probing. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 142–154, Singapore. Association for Computational Linguistics.

Maria Polinsky. 2013. *The Cambridge Handbook of Generative SYntax: Grammar and Syntax*, chapter Raising and Control. Cambridge University Press.

Paul M. Postal. 1974. *On raising: One rule of English and its theoretical implications*. MIT Press.

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

Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.

Peter Rosenbaum. 1967. *The grammar of English predicate complement constructions*. MIT Press.

Mark Rubin. 2021. When to adjust alpha during multiple testing: A consideration of disjunction, conjunction, and individual testing. *Synthese*, 199:10969–11000.

Deniz Satik. 2022. The semantics of infinitival tense. Under review.

M.E. Sánchez, Y. Sevilla, and A. Bachrach. 2016. Agreement processing in control and raising structures. evidence from sentence production in spanish. *Lingua*, 177:60–77.

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

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

Lucien Tesnière. 1959. *Eléments de Syntaxe Structurale*. Klincksieck, Paris.

Mycal Tucker, Tiwalayo Eisape, Peng Qian, Roger Levy, and Julie Shah. 2022. When does syntax mediate neural language model performance? evidence from dropout probes. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5393–5408, Seattle, United States. Association for Computational Linguistics.

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. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Rene Weber. 2007. Responses to matsunaga: To adjust or not to adjust alpha in multiple testing: That is the question. guidelines for alpha adjustment as response to o'keefe's and matsunaga's critiques. *Communication Methods and Measures*, 1:281–289.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Data Generation

Our data was generated through combinatorics of sets of words for each grammatical role. In short, our sentences followed the base structure of:

(3) [Subject] [past-tense matrix verb] [to] [embedded verb] [direct object].

In order to easily control linear distance, subject verbs were limited to pronominal subjects. Because control verbs are typically volitional, all subjects were prototypically [+HUMAN], but varied in Case, Gender, and Number (see List (4)).

We additionally selected 61 transitive verbs for our embedded verb (see List (5)). Of these verbs, 30 verbs implied human direct objects while 31 implied non-human direct objects. That is to say, a person can *flatter* the king, but it's nonsensical for them to *drink* the king. Conversely, they can drink sodas, but it would be hard to flatter an inanimate soda. This dichotomy was taken into account when selecting direct objects. Thus, when the direct object was a single-word pronominal, inanimate-coded verbs permutated through *it, that, this, stuff,* and *things* while animate-coded verbs permutated through *me, you, him, her, us, them, everyone,* and *someone*. The animate list is longer; however, the animates were truncated as we omitted direct objects that were the correspondent of the subject. That is to say, if the subject was "she", the direct object would *not* be *her*. Additionally, to avoid scope ambiguities, we excluded instances where the subject was "someone" and the direct object was "everyone".[11] Nominal direct objects ("the" + the noun) were more limited as we selected only one plausible noun to pair with the embedding verb.

(4) **Subjects**: You, He, She, We, They, Everyone, Someone

(5) **Embedded Verbs**
  a. *Inanimate-coded Verbs*: say, yell, whisper, shout, think, write, read, cook, eat, drink, buy, sell, rent, provide, offer, collect, grab, steal, bump, move, kick, break, destroy, build, wash, wear, sew, mend, fix, enjoy
  b. *Animate-coded Verbs*: kiss, hug, slap, wrestle, fight, bully, harass, intimidate, insult, slander, annoy, tease, seduce, flatter, comfort, compliment, question, interrogate, interview, meet, fire, hire, pay, reward, punish, scold, teach, train, serve, admire

(6) **Pronominal Direct Objects**
  a. *Inanimates*: it, that, this, stuff, things
  b. *Animates*: me, you, him, her, us, them, everyone, someone

(7) **Nominal Direct Objects and Their Corresponding Embedded Verb**: say the words, yell the answer, whisper the clues, shout the lyrics, think the worst, write the essay, read the book, cook the meal, eat the food, drink the sodas, buy the clothes, sell the toy, rent the apartment, provide the supplies, offer the bribes, collect the rocks, grab the keys, steal the gold, bump the table, move the chairs, kick the ball, break the glass, destroy the house, build the tower, wash the socks, wear the uniform, sew a shirt, mend the tears, fix the issue, enjoy the dessert, kiss the puppy, hug the baby, slap the clown, wrestle the children, fight the administration, bully the student, harass the reporter, intimidate the intern, insult the actress, slander the politician, annoy the teenagers, tease the toddlers, seduce the actor, flatter the king, comfort the victims, compliment the model, question the judge, interrogate the witness, interview the suspect, meet the manager, fire the employee, hire the applicant, pay the consultant, reward the winner, punish the cheaters, scold

---

[11] We did, however, include the distributive scopal alternative in which "someone" is the subject" of an "everyone" object. The two readings of this can either be there is some person X who [verbs] everyone, or it can be the distributive reading where for every person X, they are [verbed] by someone (not necessarily the same someone). The inclusion of a scopal ambiguity was due to an oversight on our part; however, because there were proportionally fewer of these pairings and because these pairings occurred in both conditions, the possible scopal ambiguity should not have an impact on our results.

the liars, teach the trainees, train the recruits, serve the queen, admire the hero

We utilized the following suite of diagnostics to select our condition matrix verbs:

1. SR predicates can be replaced by an expletive *it*; SCs cannot. (Polinsky, 2013; Landau, 2024)
    - Base: John seems/wants to annoy his brother.
    - SR: It seems John annoys his brother.
    - SC: *It wants John annoys his brother.
2. SR predicates can be replaced by an expletive *there*; SCs cannot. (Polinsky, 2013; Landau, 2024)
    - Base: A mouse seemed/wanted to be stuck in the house.
    - SR: There seemed to be a mouse stuck in the house.
    - SC: *There wanted to be a mouse stuck in the house.
3. SR predicates allow for idioms to retain their idiomatic meanings; SCs can only retrieve the literal meaning. (Polinsky, 2013; Landau, 2024)
    - Idiom: Every time my friend pet-sits, my fish *go belly up. (meaning: my fish die)*
    - SR: My fish seem to go belly up every time my friend pet-sits. *(Die meaning: still easily accessible)*
    - SC: My fish want to go belly up every time my friend pet-sits. *(Die meaning: less accessible if at all)*
4. When SR sentences are passivized, the meaning is equivalent. Passivization of the SC yields asymmetric meanings. (Sánchez et al., 2016)
    - SR: The teachers seemed to select the volunteers. = The volunteers seemed to be selected by the teachers.
    - SC: The teachers wanted to select the volunteers. $\neq$ The volunteers wanted to be selected by the teachers.
5. SRs allow for scope ambiguity, but SCs do not. (Polinsky, 2013; Landau, 2024)
    - SC: Someone from HR seems to win the office raffle every year.
        - *De re* reading: There is someone specific in HR who seems to win the raffle each year.
        - *De dicto* reading: It seems that the winner of the office raffle each year is someone from HR.
    - SR: Someone from HR wants to win the office raffle every year.
        - *De re* reading: There is someone specific in HR who wants to win the raffle each year.
        - *De dicto* reading: inaccessible.
6. Singular subjects of SC predicates can participate in plural-coded verbs,[12] but SRs cannot. (Landau, 2024). By plural-coded verbs, we mean those that necessitate multiple participants (e.g., it's ungrammatical to say "I met at midnight" as "meeting" requires two or more participants).
    - SR: *The student seemed to meet in the library.
    - SC: The student wanted to meet in the library.

From this, we selected 6 SC verbs—all of which met Landau (2024)'s criteria for logophoric control predicates—and 6 SR verbs, listed in List (8).[13]

(8) **Matrix Verbs**
  a. *Subject Control Verbs*: wanted, expected, wished, liked, hated, promised
  b. *Subject Raising Verbs*: appeared, seemed, happened, began, continued, tended

B Linear Mixed Effect Models

Below are the linear mixed effect models fit for results reported in Table 1. Random effects were identified via model comparison and included by-MatrixVerb, by-SubjectWord, by-EmbedVerb, by-ObjectWord, and by-LanguageModel random

---

[12] This is known as "partial control," and is a diagnostic for (Landau, 2024)'s logophoric control predicates.

[13] We acknowledge that three of our raising verbs are contentious: *begin* and *continue*, though they do appear as raising verbs in Postal (1974). There are instances of both appearing in the expletive construction (e.g., "It **continued** that the reserve would be 'a back-up solution only"' and "There **began** to be fewer men who paid taxes," both taken from Davies (2008–)).

slopes for our factor(s) of interest (Condition and LinDist). Word-pairs with direct objects made for more complicated linear models due to the addition of a by-ObjectWord grouping factor for random effects. Because of this, the linear model for `subj-dobj` included random intercepts for all grouping factors mentioned, but only warranted random slopes for the grouping factor of language model and linear distance. The linear model for `subj-dobj` included the same as well as a random slope for the group factor of the direct object noun/pronoun.

"Cond" refers to the Condition (SC vs SR), "CondVerb" refers to the matrix verb (6 in each condition); "Subjword" refers to the word used as the subject; "Objword" refers to the word used as the object; "Embed" refers to the embedded verb; and "Model" refers to the LLM.

- `subj-embed`: PredDist Cond + (1 | CondVerb) + (1 + Cond | SubjWord) + (1 + Cond | Embed) + (1 + Cond | Model)
- `subj-infin`: PredDist Cond + (1 | CondVerb) + (1 + Cond | SubjWord) + (1 + Cond | Embed) + (1 | Model)
- `subj-dobj`: PredDist Cond * LinDist + (1 | CondVerb) + (1 | ObjWord) + (1 | SubjWord) + (1 | Embed) + (1 + Cond + LinDist| Model)
- `embed-dobj`: PredDist Cond * LinDist + (1 | CondVerb) + (1 + Cond | ObjWord) + (1 | SubjWord) + (1 | Embed) + (1 + Cond + LinDist | Model)

C Figures

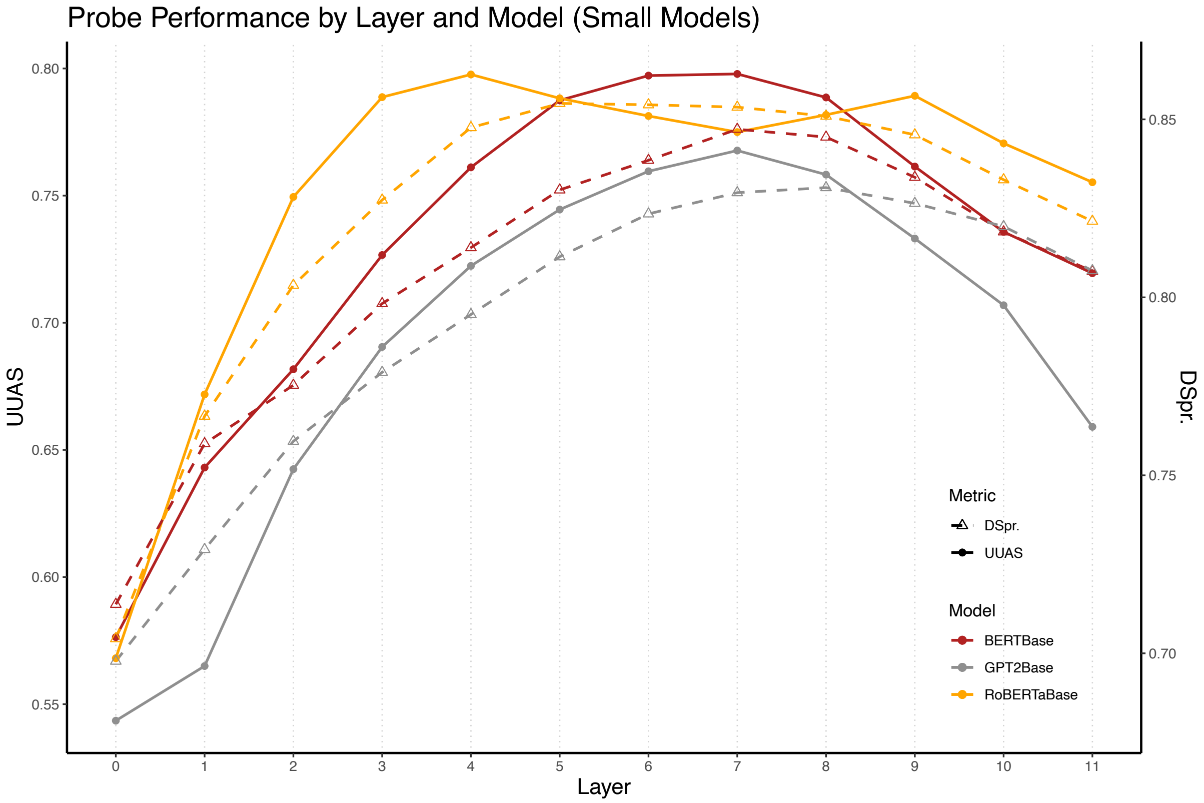


Figure 5: Probe performance for all small models. The solid lines are plotted against the left-hand y-axis and display the performance by Unlabeled Unattached Accuracy Score (UUAS) while the dotted lines plot the average Spearman correlation between the predicted and gold distances (DSpr.) along the right-hand y-axis. Highest-performing probes were `BERT-base-layer7`, `RoBERTa-base-layer4`, and `GPT2-layer7`.

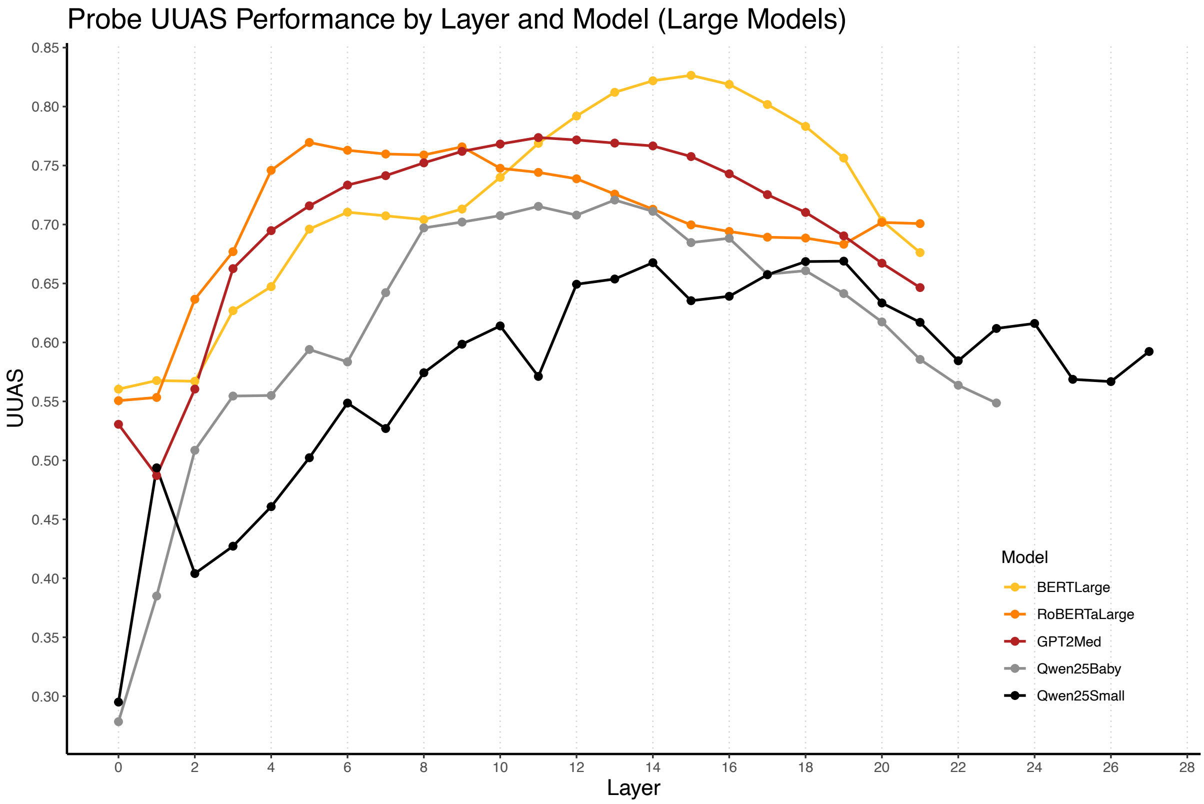


Figure 6: Probe Unlabeled Unattached Accuracy Score (UUAS) performance for all of the larger models. Highest-performing probes were `BERT-large-layer15`, `RoBERTa-large-layer5`, `GPT2-med-layer11`, `Qwen2.5-0.5-layer13` "Qwen25Baby", and `Qwen2.5-1.5-layer19` "Qwen25Small".

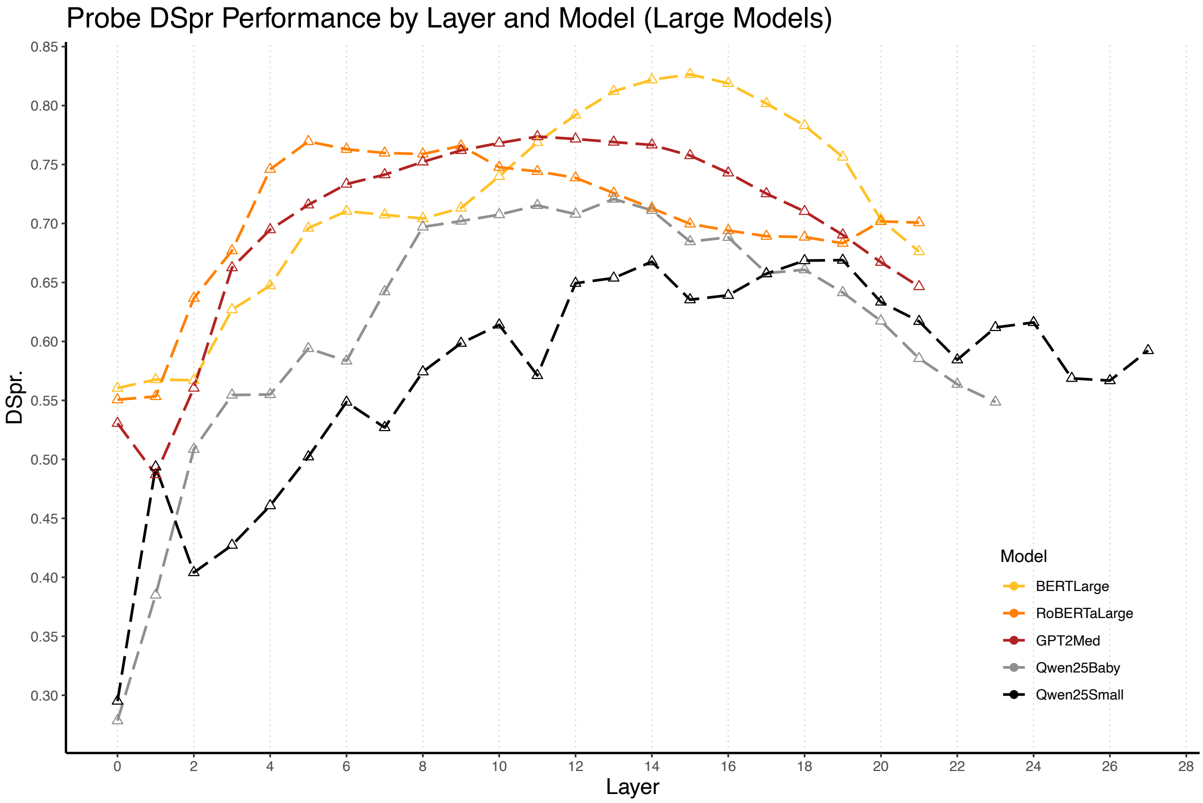


Figure 7: Probe average Spearman correlation (DSpr) performance for all of the larger models. Highest-performing probes were `BERT-large-layer15`, `RoBERTa-large-layer5`, `GPT2-med-layer11`, `Qwen2.5-0.5-layer13` "Qwen25Baby", and `Qwen2.5-1.5-layer19` "Qwen25Small".

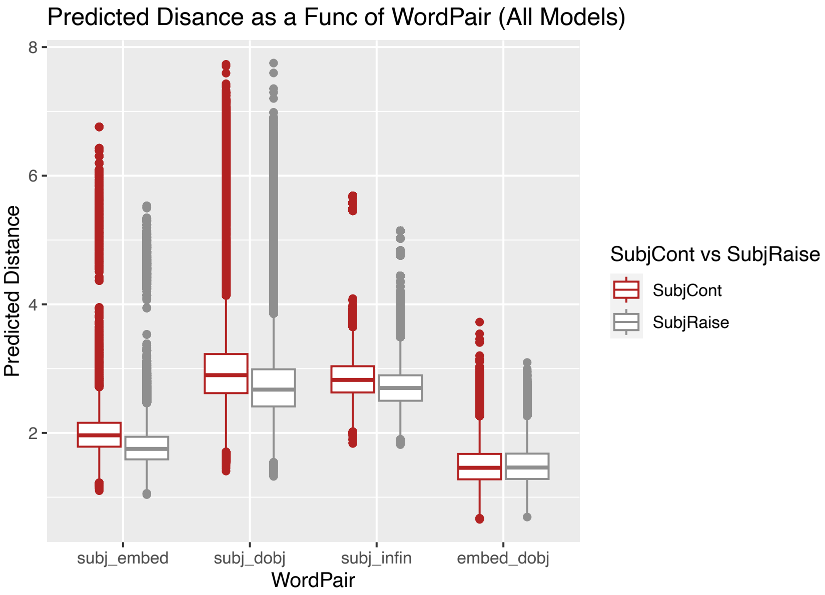


Figure 8: Predicted distances by WordPair for all LLMs. While the SC condition yields longer predicted distances than the SR condition, the baseline of `embed-dobj` shows no difference in the probes' predicted distance for the two conditions.

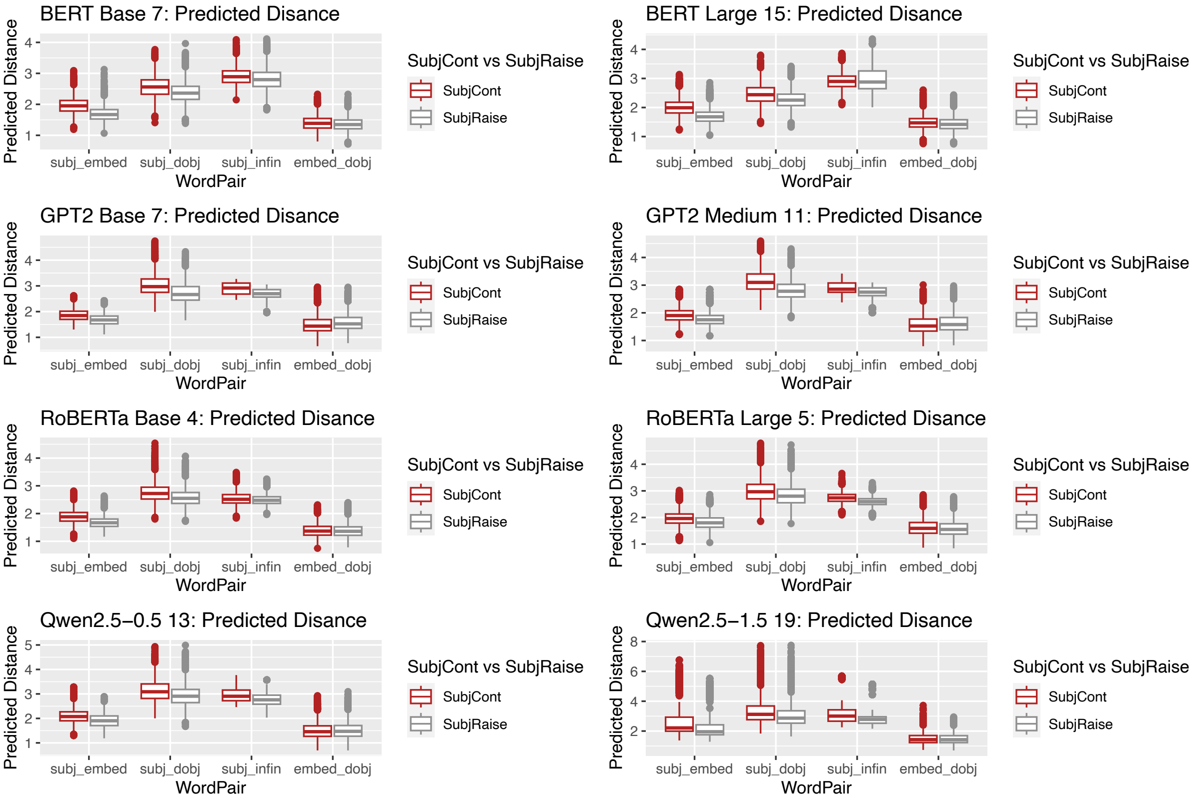


Figure 9: Predicted distances by WordPair and by LLM.

					Fixed Effects					
Model	**WordPair**	**Condition**	**Acc**	**PredDist** *(avg)*	Coefficient	$\hat{\beta}$	SE($\hat{\beta}$)	df	t	p
BB7	**Subj-Embed**	Cont	94.95%	1.96	(Intercept)	1.82245	0.04081	19.77873	44.655	<2e-16
		Raise	85.34%	1.69	Condition	-0.28385	0.04832	12.41598	-5.874	**6.58e-05**
	Subj-Infin	Cont	89.53%	2.56	(Intercept)	2.83493	0.08243	13.12998	34.39	2.97e-14
		Raise	81.48%	2.38	Condition	-0.11851	0.10040	12.28409	-1.18	0.26
	Subj-Dobj	Cont	68.54%	2.90	(Intercept)	2.77837	0.33444	69.00598	8.307	5.47e-12
		Raise	58.51%	2.81	Condition	-0.58313	0.13924	53.28816	-4.188	**0.000106**
					LinDist	-0.08603	0.06823	66.18457	-1.261	0.211758
					Interaction	0.09568	0.02659	43.99406	3.598	**0.000808**
BL15	**Subj-Embed**	Cont	96.51%	2.00	(Intercept)	1.84524	0.04304	24.16131	42.87	<2e-16
		Raise	90.34%	1.69	Condition	-0.31813	0.06931	13.65580	-4.59	**0.000448**
	Subj-Infin	Cont	95.22%	2.46	(Intercept)	2.92094	0.08590	12.97574	34.002	4.54e-14
		Raise	90.16%	2.28	Condition	0.02512	0.16398	11.02102	0.153	0.881
	Subj-Dobj	Cont	55.15%	2.91	(Intercept)	2.48195	0.26788	69.34548	9.265	9.42e-14
		Raise	58.58%	2.96	Condition	-0.38234	0.12774	39.42419	-2.993	**0.00475**
					LinDist	-0.03076	0.05465	66.48586	-0.563	0.57547
					Interaction	0.05318	0.02370	29.59535	2.243	**0.03251**
RB4	**Subj-Embed**	Cont	96.42%	1.89	(Intercept)	0.03251	0.02186	45.07316	81.24	<2e-16
		Raise	91.27%	1.68	Condition	-0.20958	0.01251	13.41216	-16.75	**2.23e-10**
	Subj-Infin	Cont	96.31%	2.54	(Intercept)	2.51492	0.04079	19.76975	61.652	<2e-16
		Raise	91.18%	2.49	Condition	-0.04289	0.06567	10.36167	-0.653	0.528
	Subj-Dobj	Cont	96.21%	2.75	(Intercept)	2.44086	0.46053	67.63892	5.300	1.37e-06
		Raise	90.68%	2.58	Condition	-0.55603	0.10135	53.91938	-5.486	**1.12e-06**
					LinDist	0.05054	0.09478	67.00031	0.533	0.596
					Interaction	0.09185	0.01952	45.36245	4.705	**2.41e-05**
RL5	**Subj-Embed**	Cont	98.61%	1.97	(Intercept)	1.88802	0.03756	71.54606	50.261	<2e-16
		Raise	96.61%	1.81	Condition	-0.15973	0.03398	11.28757	-4.701	**0.000605**
	Subj-Infin	Cont	96.91%	2.74	(Intercept)	0.000605	0.02767	24.94310	96.339	<2e-16
		Raise	93.80%	2.60	Condition	-0.14062	0.03783	10.90377	-3.717	**0.00345**
	Subj-Dobj	Cont	95.31%	2.99	(Intercept)	1.84384	0.66272	69.63534	2.782	0.00694
		Raise	91.96%	2.83	Condition	-0.62076	0.10333	52.19872	-6.008	**1.85e-07**
					LinDist	0.25798	0.13658	69.36638	1.889	*0.06308*
					Interaction	0.10951	0.02027	45.48663	5.402	**2.32e-06**
GS7	**Subj-Embed**	Cont	99.52%	1.86	(Intercept)	1.77275	0.03519	32.62122	50.379	<2e-16
		Raise	98.88%	1.69	Condition	-0.17882	0.05412	13.69496	-3.304	**0.00536**
	Subj-Infin	Cont	99.22%	2.90	(Intercept)	2.76690	0.06601	11.33192	41.91	8.7e-14
		Raise	87.23%	2.67	Condition	-0.26820	0.13411	11.95000	-2.00	*0.0688*
	Subj-Dobj	Cont	98.85%	3.04	(Intercept)	3.02957	0.44599	69.57371	6.793	3.05e-09
		Raise	96.32%	2.73	Condition	-0.66348	0.14587	54.34750	-4.548	**3.07e-05**
					LinDist	-0.03567	0.09176	68.84865	-0.389	0.69867
					Interaction	0.08601	0.02911	48.70634	2.954	**0.00481**
GM11	**Subj-Embed**	Cont	99.67%	1.92	(Intercept)	1.84576	0.05312	19.30353	34.745	<2e-16
		Raise	98.18%	1.77	Condition	-0.16218	0.08630	13.61600	-1.879	*0.0818*
	Subj-Infin	Cont	98.74%	2.89	(Intercept)	2.77503	0.07324	14.28224	37.891	9.76e-16
		Raise	93.16%	2.69	Condition	-0.18272	0.13783	12.19611	-1.326	0.209
	Subj-Dobj	Cont	97.49%	3.15	(Intercept)	2.93307	0.05039	54.90142	58.210	<2e-16
		Raise	90.68%	2.83	Condition	-0.33108	0.07295	17.11412	4.538	**0.000286**
QB13	**Subj-Embed**	Cont	95.14%	2.10	(Intercept)	2.01250	0.06700	11.27262	30.039	4.08e-12
		Raise	99.32%	1.91	Condition	-0.18818	0.06758	12.38379	-2.785	**0.0161**
	Subj-Infin	Cont	72.19%	2.95	(Intercept)	2.8575	0.1087	12.3465	26.276	3.27e-12
		Raise	72.32%	2.78	Condition	-0.2119	0.1363	11.6518	-1.555	0.147
	Subj-Dobj	Cont	92.73%	3.12	(Intercept)	2.39605	0.40124	68.76982	5.972	9.24e-08
		Raise	95.83%	2.93	Condition	-0.19786	0.06810	12.92618	-2.905	**0.0123**
					LinDist	0.15452	0.08148	65.01205	1.896	*0.0624*
QS19	**Subj-Embed**	Cont	87.29%	2.66	(Intercept)	2.55208	0.43185	6.06759	5.910	0.00100
		Raise	94.86%	2.30	Condition	-0.31453	0.08549	14.93464	-3.679	**0.00225**
	Subj-Infin	Cont	67.86%	3.23	(Intercept)	3.2211	0.3615	6.1781	8.911	9.44e-05
		Raise	68.65%	2.95	Condition	-0.2839	0.1270	14.2346	-2.236	**0.0419**
	Subj-Dobj	Cont	86.06%	3.50	(Intercept)	3.98534	0.53651	13.82113	7.428	3.47e-06
		Raise	93.00%	3.18	Condition	-0.27527	0.05825	11.01648	-4.726	**0.000621**
					LinDist	-0.14038	0.06518	66.57211	-2.154	**0.034893**

Table 2: From top to bottom, models are: `BERT-base-layer7`, `BERT-large-layer15`, `RoBERT-base-layer4`, `RoBERT-large-layer5`, `GPT2-small-layer7`, `GPT2-medium-layer11`, `Qwen2.5-0.5-layer13`, and `Qwen2.5-1.5-layer19`. Model comparison using anova revealed LinDist did not significantly improve the linear mixed-effects model for GPT2-medium; this method also showed the interaction between the Condition (SC vs SR) and LinDist was not a main effect for the **Qwen2.5** models.

Timestep Embeddings Trigger Collapse in Diffusion Text Generation

Ryota Nosaka
Tokyo University of Science
`1424519@ed.tus.ac.jp`

Takuya Matsuzaki
Tokyo University of Science
`matuzaki@rs.tus.ac.jp`

Abstract

Diffusion models have achieved remarkable success in various generative tasks, particularly in image and audio synthesis, which work by iteratively refining random noise into realistic data. Recent studies have highlighted the potential of diffusion models for text generation, but several challenges remain unresolved. One significant issue is that the model begins to degrade a previous sample rather than improve it after a certain timestep in the generation process, resulting in broken text. In this paper, we reveal that timestep embeddings are a principal cause of the collapse problem by analyzing their interactions with word embeddings. Further, we propose two key methods: (a) a simple lightweight word embedding technique that enhances model analyzability as well as learning efficiency; (b) a novel regularization on both word and timestep embeddings. Experimental results demonstrate that our approach effectively mitigates the collapse problem and can lead to a considerable improvement in the quality of generated text.

1 Introduction

Diffusion models are a class of generative models that have achieved state-of-the-art performance in continuous data generation, such as image and audio synthesis (Ho et al., 2020; Song et al., 2021; Kong et al., 2021). The generation process begins by sampling random noise at timestep T, and then progressively *denoising* it toward timestep 0, resulting in realistic data. Several studies have attempted to adapt diffusion models for text generation via word embeddings and have recently demonstrated performance comparable to earlier autoregressive models (Li et al., 2022; Gong et al., 2023; Yuan et al., 2024; Gao et al., 2024), such as GPT-2 (Radford et al., 2019).

However, diffusion-based text generation still faces challenges in ensuring high-quality output. For text generation, the one-step denoising task at timestep t is typically formulated as fully removing noise and reintroducing a smaller amount of noise corresponding to timestep $t-1$. One critical issue is that the model starts failing to perform the full denoising task after a certain timestep, leading to incoherent or grammatically incorrect output (Gao et al., 2024). We refer to it as the *collapse problem*. This phenomenon is counterintuitive, as the denoising task should gradually become easier as the generation process progresses.

To obtain a high-quality sample, a common approach is to generate multiple times and then select the best one using a re-ranking algorithm like Minimum Bayes-Risk (MBR) decoding (Kumar and Byrne, 2004). However, it compromises diversity, which is a key strength of diffusion models. Gao et al. (2024) reported that the collapse problem can be mitigated by their regularization for word embeddings and heuristically modified training and generation processes. This highlights that learning continuous representations of the vocabulary is a core problem, and motivates us to develop a solution that does not rely on heuristically tweaking the diffusion framework.

Usually, a single denoiser model is shared across all timesteps, and learned timestep embeddings are incorporated as signals of timesteps. Although it has long been common practice, the relation between the timestep embeddings and the collapse problem has not been adequately investigated. Meanwhile, it is also widely adopted to use low-dimensional word embeddings combined with additional projection layers. This approach enhances learning efficiency, but increases model complexity; it introduces non-linearity into the way timestep embeddings act on the model.

In this paper, in pursuit of a fundamental solution to preserve the expected behavior of diffusion models—progressively refining data quality—we investigate both word and timestep embeddings and their relationship. Firstly, we introduce a plug-and-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 397–406
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

play low-rank word embedding technique. It makes the effect of timestep embeddings more transparent while keeping computational cost low. Secondly, we propose a new metric that reveals that the timestep embedding disrupts word embeddings in addition to the noise. Moreover, based on our metric, we design a novel regularization method to counteract the adverse effects of timestep embeddings within the standard diffusion architecture. Experiments demonstrated the occurrence of the collapse problem and its mitigation through the combination of our two methods.

2 Related Work

2.1 Text Generation with Diffusion Models

Diffusion models are highly powerful generative models. They first intentionally diffuse real data step by step and learn to reverse this process. New realistic data are then generated by sampling from the stationary distribution and iteratively applying the learned denoising operation.

In recent years, research on diffusion-based text generation has been advancing. Broadly, two primary approaches have been proposed: continuous and discrete diffusion language models. Continuous methods generate word embeddings and discretize them to obtain words. Following diffusion image synthesis methods, such as DDPM (Ho et al., 2020) and DDIM (Song et al., 2021), they are based on Gaussian distributions (Li et al., 2022). Discrete methods work directly in word space and define the forward process as sampling from a categorical distribution, such as gradually replacing words with other words or mask tokens (Austin et al., 2021).

At present, diffusion language models are generally known to have limitations in generating high-quality text. In this paper, we focus on how continuous models work, which are built upon the standard diffusion architecture.

2.2 Word Embeddings

In continuous diffusion text generation, the key distinction from image generation lies in the necessity of constructing continuous representations of words in parallel with learning the denoiser model. Gong et al. (2023) reported a significant performance degradation when using fixed pretrained embeddings, implying that word embeddings need to be optimized for diffusion language models.

It is simultaneously essential to consider the discretization. Typically, a rounding distribution is defined and its likelihood is maximized, which facilitates the segregation of word embeddings. Regarding this, Gao et al. (2024) pointed out that the traditional rounding loss is insufficient to procure the desired distribution and proposed the anchor loss. Their method enhances the distinguishability of embeddings than the rounding loss, although the collapse problem still occurs around the final steps of generation.

2.3 Timestep Embeddings

Timestep embeddings play an important role by conditioning the denoiser model on the noise level of input. For continuous diffusion models, a timestep embedding is typically constructed by mapping a timestep to a vector using sinusoidal encoding and transforming it with a multi-layer perceptron (MLP). Particularly in text generation, research on the impact of timestep embeddings remains limited, whereas there are reports in discrete models. For instance, He et al. (2023) tested several embedding methods of timesteps and reported that they significantly impact on the performance. While this pertains to the discrete method, it emphasizes the importance and challenges of timestep embeddings.

3 Preliminaries

3.1 Diffusion Models

Denoising diffusion probabilistic models (DDPMs) (Ho et al., 2020) are the standard architecture of diffusion models. A DDPM consists of two processes: the forward process and the reverse process.

Given a training data sample $\mathbf{z}_0 \sim q(\mathbf{z}_0)$, the forward process gradually adds noise to $\mathbf{z}_0$, transforming it into random noise $\mathbf{z}_T$:

$$q(\mathbf{z}_t \mid \mathbf{z}_{t-1}) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\alpha_t}\mathbf{z}_{t-1}, \beta_t I\right)$$

where $0 < \beta_1 < \cdots < \beta_T < 1, \alpha_t = 1 - \beta_t$ are hyperparameters called *noise schedule*. Since Gaussian distributions are reproducible, the distribution of $\mathbf{z}_t$ conditioned on $\mathbf{z}_0$ has a closed form for any timestep t:

$$q(\mathbf{z}_t \mid \mathbf{z}_0) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\bar{\alpha}_t}\mathbf{z}_0, \bar{\beta}_t I\right) \tag{1}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s,\ \bar{\beta}_t = 1 - \bar{\alpha}_t$. A denoiser model $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$ is trained using $\{\mathbf{z}_t\}_{t=0}^{T}$.

The reverse process generates a new sample $\mathbf{z}_0$ by starting from random noise $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, I)$ and iteratively denoising via $p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t)$.

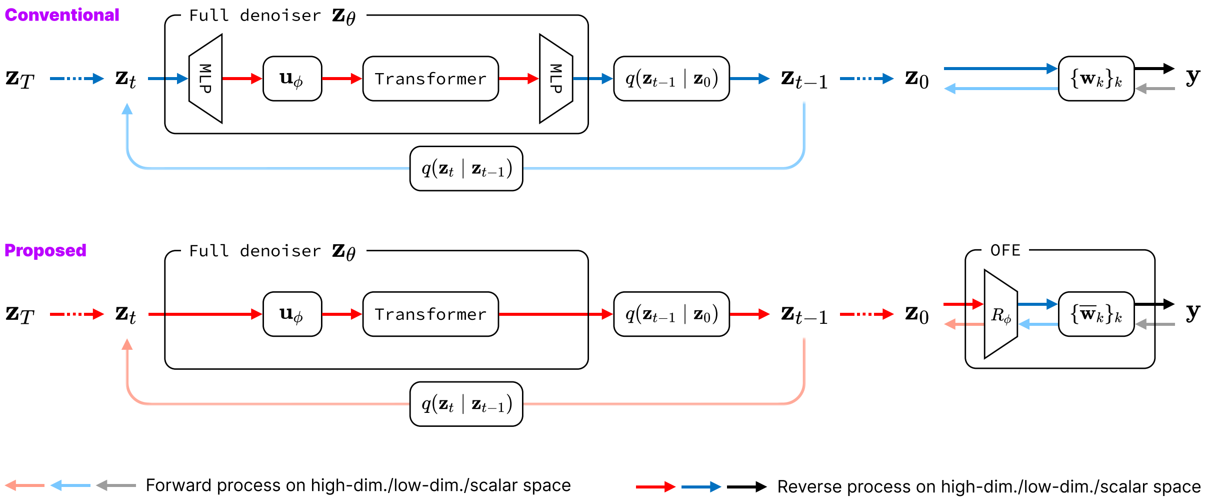

Conventional
Full denoiser $\mathbf{z}_\theta$
$\mathbf{z}_T$
$\mathbf{z}_t$
MLP
$\mathbf{u}_\phi$
Transformer
MLP
$q(\mathbf{z}_{t-1} \mid \mathbf{z}_0)$
$\mathbf{z}_{t-1}$
$\mathbf{z}_0$
$\{\mathbf{w}_k\}_k$
$\mathbf{y}$
$q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$
Proposed
Full denoiser $\mathbf{z}_\theta$
$\mathbf{z}_T$
$\mathbf{z}_t$
$\mathbf{u}_\phi$
Transformer
$q(\mathbf{z}_{t-1} \mid \mathbf{z}_0)$
$\mathbf{z}_{t-1}$
$\mathbf{z}_0$
OFE
R_ϕ
$\{\overline{\mathbf{w}}_k\}_k$
$\mathbf{y}$
$q(\mathbf{z}_t \mid \mathbf{z}_{t-1})$
Forward process on high-dim./low-dim./scalar space
Reverse process on high-dim./low-dim./scalar space

Figure 1: Architecture overview. In the conventional model, the diffusion model (i.e., noisy word embeddings) and Transformer work on spaces of different dimensions and MLPs are unavoidably inserted to bridge them. The proposed model eliminates this discrepancy by consolidating up-/down-projections within the OFE.

3.2 Text Generation with Diffusion Models

Diffusion text generation first generates a sequence of word embeddings $\mathbf{z}_0 = [\mathbf{z}_{0i}]_{i=1}^{L}$ through denoising and then discretize it into a sequence of words $\mathbf{y} = [y_i]_{i=1}^{L}$ (Yuan et al., 2024).

Let $\mathbf{w}_k$ be the embedding of the k-th word in the vocabulary ($k \in \{1, 2, ..., V\}$). The forward process begins by sampling each $\mathbf{z}_{0i}$ as follows:

$$q_\phi(\mathbf{z}_{0i} \mid y_i) = \mathcal{N}(\mathbf{z}_{0i}; \mathbf{w}_{y_i}, \beta_0 I)$$

where β_0 is a very small constant.

At timestep t, the one-step denoising is expressed as

$$\begin{aligned} p_\theta(\mathbf{z}_{t-1} \mid \mathbf{z}_t) &= q(\mathbf{z}_{t-1} \mid \mathbf{z}_0 = \mathbf{z}_\theta(\mathbf{z}_t, t)), \\ \mathbf{z}_\theta(\mathbf{z}_t, t) &= \mathrm{Transformer}_\theta(\mathbf{u}_\phi(\mathbf{z}_t, t)), \\ \mathbf{u}_\phi(\mathbf{z}_t, t) &= [\mathbf{z}_{ti} + \mathbf{u}_t]_{i=1}^{L}. \end{aligned}$$

The full denoiser $\mathbf{z}_\theta$ is a Transformer model (Vaswani et al., 2017). The function $\mathbf{u}_\phi$ fuses a cue of the timestep into noisy word embeddings by adding the timestep embedding. The timestep embedding $\mathbf{u}_t$ is usually parameterized by transforming sinusoidal encoding of the timestep t through an MLP and we follow it. The denoising loss $\mathcal{L}_{\text{denoise}}$ is

$$\mathcal{L}_{\text{denoise}} = \mathbb{E}_{\mathbf{y}, \mathbf{z}_{0:T}} \left[\sum_{t=2}^{T} \| \mathbf{z}_0 - \mathbf{z}_\theta(\mathbf{z}_t, t) \|^2 \right. \\ \left. + \| \mathbf{w}_{y_i} - \mathbf{z}_\theta(\mathbf{z}_1, 1) \|^2 + \| \sqrt{\bar{\alpha}_T} \mathbf{z}_0 \|^2 \right].$$

In practice, we sample $t \in \{1, 2, ..., T\}$ for each minibatch.

Minimizing only $\mathcal{L}_{\text{denoise}}$ would cause all word embeddings to collapse into a single point because $\mathcal{L}_{\text{denoise}}$ is mainly composed of the mean squared error between $\mathbf{z}_0$ and $\mathbf{z}_\theta(\mathbf{z}_t, t)$. Therefore, we need to promote appropriate segregation among word embeddings. Concretely, we define the distribution for rounding an embedding $\mathbf{w} \in \mathbb{R}^d$ to the k-th word as

$$p_\phi(k \mid \mathbf{w}) = \frac{\exp \mathbf{w}^\top \mathbf{w}_k}{\sum_{\ell=1}^{V} \exp \mathbf{w}^\top \mathbf{w}_\ell} \tag{2}$$

and the rounding loss $\mathcal{L}_{\text{round}}$ as

$$\mathcal{L}_{\text{round}} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{E}_{\mathbf{y}, \mathbf{z}_0} \left[-\log p_\phi(y_i \mid \mathbf{z}_{0i}) \right].$$

Another approach is the anchor loss (Gao et al., 2024) that uses the full denoising prediction instead of training data samples:

$$\mathcal{L}_{\text{anchor}} = \frac{1}{L} \sum_{i=1}^{L} \mathbb{E}_{\mathbf{y}, \mathbf{z}_t} \left[-\log p_\phi(y_i \mid \mathbf{z}_\theta(\mathbf{z}_t, t)_i) \right].$$

In this work, we choose the rounding loss, which is the standard way. The total loss $\mathcal{L}_{\text{total}}$ is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{denoise}} + \mathcal{L}_{\text{round}}.$$

Computing $\mathcal{L}_{\text{round}}$ is expensive because it is performed over the entire target sequence. On the other hand, using low-dimensional word embeddings and a smaller Transformer hurts prediction performance (see §5). Hence, in order to reduce

	d'	d & Hidden Dim.	Feed-Forward Dim.	Layers	Attention Heads
`Base`	-	768	2048	6	12
`Low-d`	-	132	2048	6	12
`OFE`	128	768	2048	6	12

Table 1: Hyperparameters in the exploratory experiments.

computational costs while keeping the capacity of Transformer, it is common practice to use low-dimensional word embeddings and append MLPs for up- and down-projection before $\mathbf{u}_\phi$ and after the Transformer (see Figure 1).

In reverse process, the prediction of $\mathbf{z}_{t-1}$ given the previous prediction of $\mathbf{z}_t$ is sampled by

$$\mathbf{z}_{t-1} = \sqrt{\bar{\alpha}_t}\mathbf{z}_\theta(\mathbf{z}_t, t) + \sqrt{\bar{\beta}_t}\varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, I). \tag{3}$$

Then the last output $\mathbf{z}_\theta(\mathbf{z}_1, 1)$ are rounded to words.

The simplest way to perform a sequence-to-sequence task is to use an encoder-decoder Transformer. The source text is input to the encoder, and the noisy target text $\mathbf{z}_t$ to the decoder with non-causal attention.

4 Lightweight Word Embeddings

The conventional MLP-based method for reducing the dimensionality of word embeddings is effective but complicates the interaction between word and timestep embeddings; the MLP first non-linearly transforms noisy word embeddings, and then the timestep embedding is applied. Besides, it may also extract information about timesteps according to the noise level of input, making it difficult to analyze the conditioning by timesteps.

This MLP is actually not required if the model has no such dimensional mismatch. To eliminate this entanglement, we propose the *Orthogonally Factorized Embedding* (OFE) technique that employs low-dimensional embeddings while letting both the diffusion model (i.e., noisy word embeddings) and Transformer work on high-dimensional space (see Figure 1). The OFE consists of low-dimensional word embeddings $\{\overline{\mathbf{w}}_k \in \mathbb{R}^{d'}\}_k$ along with a learned column-orthogonal matrix $R_\phi \in \mathbb{R}^{d \times d'}$. Every word embedding $\mathbf{w}_k$ is expressed as

$$\mathbf{w}_k = R_\phi \overline{\mathbf{w}}_k.$$

The rounding distribution can be rewritten as

$$p_\phi(k \mid \mathbf{w}) = \frac{\exp(R_\phi^\top \mathbf{w})^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^{V} \exp(R_\phi^\top \mathbf{w})^\top \overline{\mathbf{w}}_\ell}.$$

Notably, for every word embedding $\mathbf{w}_m = R_\phi \overline{\mathbf{w}}_m$,

$$\begin{aligned} p_\phi(k \mid \mathbf{w}_m) &= \frac{\exp(R_\phi^\top R_\phi \overline{\mathbf{w}}_m)^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^{V} \exp(R_\phi^\top R_\phi \overline{\mathbf{w}}_m)^\top \overline{\mathbf{w}}_\ell} \\ &= \frac{\exp \overline{\mathbf{w}}_m^\top \overline{\mathbf{w}}_k}{\sum_{\ell=1}^{V} \exp \overline{\mathbf{w}}_m^\top \overline{\mathbf{w}}_\ell} \end{aligned} \tag{4}$$

since $R_\phi^\top R_\phi = I$. Consequently, $p_\phi(k \mid \mathbf{w})$ reduces to the rounding in the low-dimensional embedding space.

Here, we provide the rationale behind the column-orthogonal constraint. Since the rounding loss promotes appropriate separation of word embeddings as noted in §3.2, the reconstruction of a low-dimensional embedding from a high-dimensional representation, i.e., $R_\phi^\top R_\phi \overline{\mathbf{w}}_m = \overline{\mathbf{w}}_m$ as can be seen in Eq. (4), seems to be naturally acquired due to the rounding loss even when using an unconstrained matrix for R_ϕ. However, we empirically found that this is not the case; in the course of training, $R_\phi^\top R_\phi$ gets close to a scaled identity matrix λI and λ gets larger and larger, and training becomes unstable. We conjecture that the reason of this phenomenon is that λ works similarity to the inverse temperature in a temperature softmax. Therefore, we attempt to remedy this problem by eliminating a scaling ambiguity between R_ϕ and low-dimensional embeddings $\overline{\mathbf{w}}_k$, i.e., the same high-dimensional embedding is obtained with γR_ϕ and $\gamma^{-1}\overline{\mathbf{w}}_k$ for any $\gamma \neq 0$: $\mathbf{w}_k = (\gamma R_\phi)\left(\gamma^{-1}\overline{\mathbf{w}}_k\right)$. For that, we choose to impose column-orthogonality on R_ϕ, thereby constraining its Frobenius norm. In addition, it enables the reconstruction by multiplying $R_\phi^\top$ from the left, which ensures that the high-dimensional rounding $p_\phi(k \mid \mathbf{w})$ is equivalent to the low-dimensional rounding. We expect that this property further contributes to stability.

Because a matrix can be parameterized under the column-orthogonal constraint,[1] we can employ the OFE by simply replacing the existing embedding and rounding functions in a model with those of the OFE.

---

[1] `https://pytorch.org/docs/stable/generated/torch.nn.utils.parametrizations.orthogonal.html`

5 Exploratory Experiments

In this section, we demonstrate the collapse problem and confirm that it is not attributable to the dimensionality reduction by comparing the three cases of denoisers shown in Table 1. `Base` does not use low-dimensional embeddings. In `Low-d`, $d = 132$ was chosen as the best feasible alternative to 128 ($= d'$ in `OFE`) conforming with 12 attention heads. We conducted experiments on Quora Question Pairs (DataCanary et al., 2017) and measured the BERTScore (Zhang et al., 2020) of the paraphrased questions generated by the models. Further details are described in the main experiments (§7), including the reason for setting $d' = 128$ in `OFE`.

Figure 2 illustrates the evaluation of intermediate samples during the reverse process, namely the full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t. The collapse problem was observed in all models, including `Base`. Thus, the collapse is not due to the dimensionality reduction techniques.

`Low-d` expectedly exhibited worse performance than `Base`. By contrast, `OFE` substantially outperformed `Low-d` and stood comparison with `Base`. Furthermore, it is noteworthy that `OFE` surpassed `Base` early in generation.

We found that models before convergence hardly exhibit collapse as shown in Figure 3. These findings suggest that continuing training only with the traditional loss makes the vectors $\mathbf{u}_\phi(\mathbf{z}_t, t)$, the input to the Transformer, more difficult to handle.

6 Study of Timestep Embeddings

The model has been simplified by the OFE, which allows word embeddings to naturally work on the same dimension as timestep embeddings (Figure 1). Thus, we are now able to analyze the relationship between these two embeddings directly. In this section, we first introduce a new metric that measures how timestep embeddings are implicated in the phenomenon that the vectors $\mathbf{u}_\phi(\mathbf{z}_t, t)$ tend to form an undesired structure. We then transform this score to a regularization method to mitigate the collapse problem.

6.1 Analysis of Ambiguity

A denoiser model learns the mapping from a hidden variable $\mathbf{z}_{0i}$ to a word y_i via the rounding loss. In addition, each vector in $\mathbf{u}_\phi(\mathbf{z}_t, t)$, the input of the denoiser, must be properly distinguished from one another, particularly at early diffusion steps.

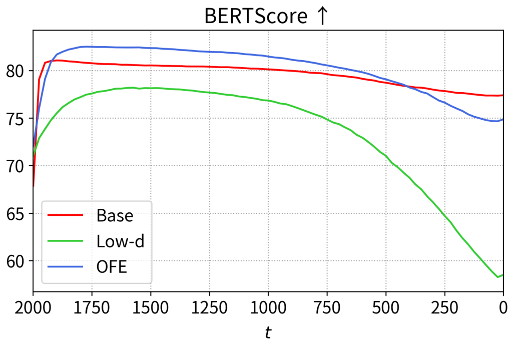


Figure 2: The evaluation of full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t in the exploratory experiments.

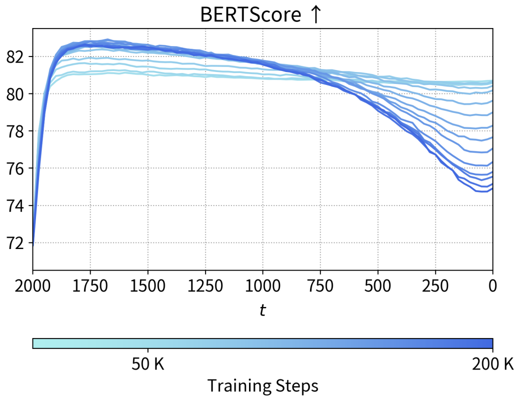


Figure 3: The progression of the collapse problem in `OFE` in the course of training.

The forward process is performed by shrinking word embeddings toward the origin and then adding Gaussian noise (Eq. (3); Figure 4). Since the model learns the inverse operation of the forward process, the variance of the full denoising prediction $\mathbf{z}_{0i} = \mathbf{z}_\theta(\mathbf{z}_t, t)_i$ for the input $\mathbf{z}_{ti}$ is expected to decrease as the generation progresses, and hence the sample is gradually determined. However, due to the timestep embedding, the actual input vector $\mathbf{z}_{ti} + \mathbf{u}_t$ may be confused with unrelated words *at different timesteps* (Figure 5).

This hypothesis motivates us to measure which words are now mistaken for which words due to timestep embeddings. Inspired by Kullback-Leibler divergence, we define the *Temporal Ambiguity Score* (TAS) between t and t' as follows:[2]

$$\mathrm{Ambig}\left(t, t'\right) := \frac{1}{V} \sum_{k=1}^{V} \sum_{\ell=1}^{V} \mathbb{E}_{\mathbf{w}_{tk}, \mathbf{w}_{t'k}} \Bigg[p_\phi\left(\ell \mid \mathbf{u}_\phi\left(\mathbf{w}_{t'k}, t'\right)\right) \log \frac{p_\phi\left(\ell \mid \mathbf{u}_\phi(\mathbf{w}_{t'k}, t')\right)}{p_\phi\left(\ell \mid \mathbf{u}_\phi(\mathbf{w}_{tk}, t)\right)} \Bigg]$$

---

[2] The rounding p_ϕ always denotes the softmax distribution over unmodified word embeddings, as shown in Eq. (2).

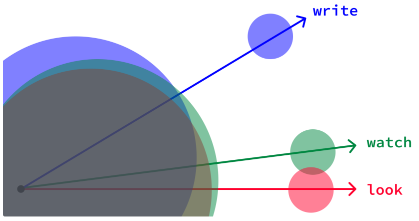


Figure 4: Noise addition at timesteps $0.2T$ and $0.8T$. The circles represent the regions from which $\mathbf{z}_t$ is mostly sampled. The regions move closer to the origin and expand their radii as the forward process progresses.

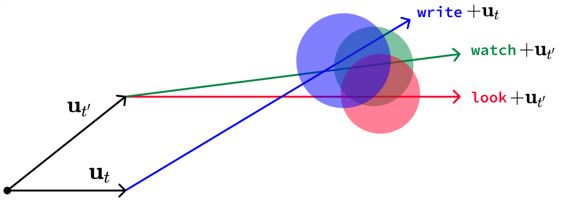


Figure 5: How the timestep embeddings disrupt noisy word embeddings. The noisy "write" at timestep t will be confused with the noisy "watch" and "look" at timestep t'.

where $\mathbf{w}_{tk}$ is a noisy embedding of the k-th word at timestep t. This metric represents how the rounding results of noisy word embeddings are altered by timestep embeddings. It is based on a fact that is peculiar to text generation: unlike in image synthesis, the effect of noise addition lies in whether the rounding yields different words before and after it.

Since exact calculation of $\mathrm{Ambig}\,(t, t')$ is challenging, we estimate it by sampling $(\mathbf{w}_{tk}, \mathbf{w}_{t'k})$ once for each k.

6.2 Regularization for Disambiguation

We consider using the ambiguity score as an objective function. In imitation of the rounding loss, we introduce the *Temporal Disambiguation Loss* using training data samples instead of entire vocabulary:

$$\mathcal{L}_{\text{disambig}} := \frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{L}\mathbb{E}_{\mathbf{y},\mathbf{z}_t,\mathbf{z}_{t'}\sim q}\Bigg[p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{t'i}, t'))\log\frac{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{t'i}, t'))}{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{ti}, t))}\Bigg].$$

Since directly minimizing it is difficult, we further simplify the loss. First, we fix t' at 0, because it is not meaningful to disambiguate embeddings at late diffusion steps, which are distributed almost randomly. Thus, we get

$$\mathcal{L}'_{\text{disambig}} = \frac{1}{L}\sum_{i=1}^{L}\sum_{j=1}^{L}\mathbb{E}_{\mathbf{y},\mathbf{z}_t\sim q}\Bigg[p_\phi(y_j \mid \mathbf{w}_{y_i})\log\frac{p_\phi(y_j \mid \mathbf{w}_{y_i})}{p_\phi(y_j \mid \mathbf{u}_\phi(\mathbf{z}_{ti}, t))}\Bigg].$$

Besides, the model easily learns so that $p_\phi(y_j \mid \mathbf{w}_{y_i}) \approx 1$ if $i = j$, and 0 otherwise. Ignoring noise for training stability, we arrive at the *Simplified Temporal Disambiguation Loss* (STDL):

$$\mathcal{L}^{\text{simple}}_{\text{disambig}} := \frac{1}{L}\sum_{i=1}^{L}\mathbb{E}_{\mathbf{y},t\sim q}\big[-\log p_\phi\left(y_i \mid \mathbf{u}_\phi\left(\sqrt{\bar{\alpha}_t}\mathbf{w}_{y_i}, t\right)\right)\big].$$

Note that, from Eq. (1), $\sqrt{\bar{\alpha}_t}\mathbf{w}_{y_i}$ is the expectation of $q(\mathbf{z}_{ti} \mid \mathbf{z}_{0i} = \mathbf{w}_{y_i})$. If we set $\bar{\alpha}_0 = 1$ and ignore the timestep embedding (i.e., $\mathbf{u}_0 = \mathbf{0}$), $\mathcal{L}^{\text{simple}}_{\text{disambig}}$ equals $\mathcal{L}_{\text{round}}$ when $t = 0$. Hence, this is a generalization of the conventional loss associated with rounding. Intuitively, it aims to obtain consistency across all timesteps in the space of $\mathbf{u}_\phi(\cdot, t)$ with respect to the noiseless word embedding space.

7 Main Experiments

7.1 Setup

We set the number of diffusion steps to $T = 2000$ and use the *sqrt* noise schedule (Li et al., 2022), which are widely used. We choose $d' = 128$ to align with baselines that utilize the MLP-based dimensionality reduction method. The output length is adjusted by generating padding tokens. The embedding of the padding token is learned as well as other word embeddings.

`Base` and `OFE` are identical to those used in §5. `OFE+STDL` refers to the model incorporating both OFE and STDL.

The generation proceeds step by step. The results were saved every 25 steps and the final step[3] to reduce the computational cost of evaluation.

Datasets We conduct experiments on two popular sequence-to-sequence tasks: Paraphrasing on Quora Question Pairs (DataCanary et al., 2017) and Text Simplification on Wiki-Auto (Jiang et al., 2020). The data split follows that used for DiffuSeq (Gong et al., 2023).

Metrics The quality of samples is evaluated using BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2020)[4]. The diversity among outputs generated from the same

---

[3] That is, $t = 2000, 1975, ..., 25, 1$.

[4] Following DiffuSeq, we use `microsoft/deberta-xlarge-mnli` for computing BERTScore.

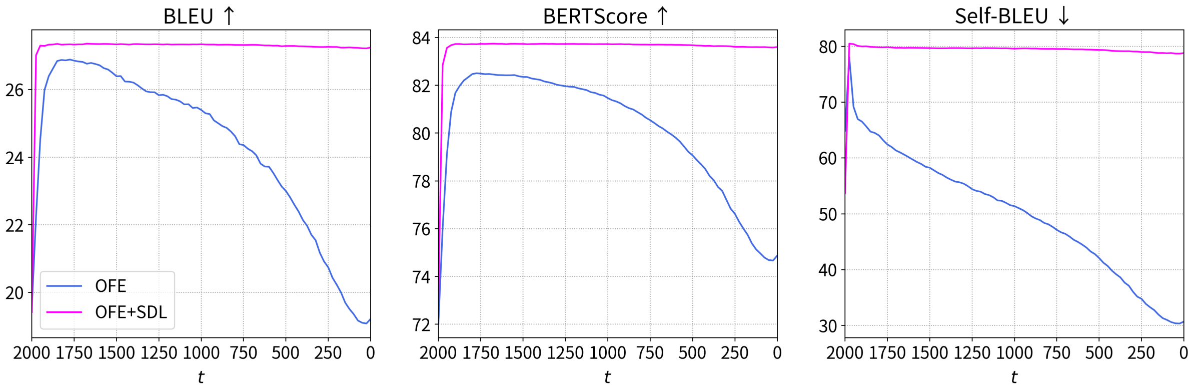
	MBR	BLEU	ROUGE-L	BERTScore	Self-BLEU
DiffuSeq	-	18.29†	52.99†	79.30†	27.32
DiffuSeq	10	24.13	58.80	83.65	-
SeqDiffuSeq	-	23.28	-	82.91	-
SeqDiffuSeq	10	24.34	-	84.00	-
Difformer	-	28.52	60.15	83.80	-
Difformer	10	30.43	61.25	85.02	-
Difformer	20	30.52	61.08	85.02	-
Base	-	22.19 (−2.38)	53.27 (−3.74)	77.39 (−3.66)	59.19 (+ 0.12)
OFE	-	19.21 (−7.68)	50.29 (−9.34)	74.86 (−7.64)	30.67 (+ 0.34)
OFE	10	25.85	57.73	80.64	-
OFE+STDL	-	27.24 (−0.12)	60.29 (−0.11)	83.60 (−0.14)	78.76 (+25.08)
OFE+STDL	10	27.90	60.90	84.20	-
OFE+STDL	20	28.05	61.02	84.30	-

(a) Quora Question Pairs

	MBR	BLEU	ROUGE-L	BERTScore	Self-BLEU
DiffuSeq	-	29.29†	53.13†	77.81†	46.42
DiffuSeq	10	36.22	58.49	81.26	-
SeqDiffuSeq	-	37.09	-	82.11	-
SeqDiffuSeq	10	37.12	-	82.14	-
Difformer	-	40.37	59.56	81.96	-
Difformer	10	40.77	59.86	82.21	-
Difformer	20	40.84	59.88	82.29	-
Base	-	27.85 (−11.68)	50.99 (− 8.78)	72.98 (−9.19)	58.32 (+ 1.43)
OFE	-	30.20 (−12.71)	49.78 (−10.03)	72.70 (−9.38)	48.51 (− 0.00)
OFE	10	39.82	58.15	80.25	-
OFE+STDL	-	41.45 (− 0.12)	59.02 (− 0.07)	81.75 (−0.09)	96.16 (+20.01)
OFE+STDL	10	41.49	59.12	81.84	-
OFE+STDL	20	41.49	62.35	81.84	-

(b) Wiki-Auto

Table 2: Generation qualities and diversities in the main experiments. Difference between the best sample through generation process and the final output are shown in parentheses. † indicates that we evaluated the samples released by the authors. Other baseline results are cited from their paper.



Quora Question Pairs

Figure 6: The evaluation of full denoising predictions $\mathbf{z}_\theta(\mathbf{z}_t, t)$ at each timestep t in the main experiments.

input is a unique strength of diffusion models. To measure this, we employ Self-BLEU (Zhu et al., 2018).

Baselines DiffuSeq (Gong et al., 2023) is a basic diffusion language model. SeqDiffuSeq (Yuan et al., 2024) is an advanced encoder-decoder model with adjusting noise schedule during training. Difformer (Gao et al., 2024) is a model designed to mitigate the collapse problem by improving its training objective and generation process.

7.2 Generation Quality and Diversity

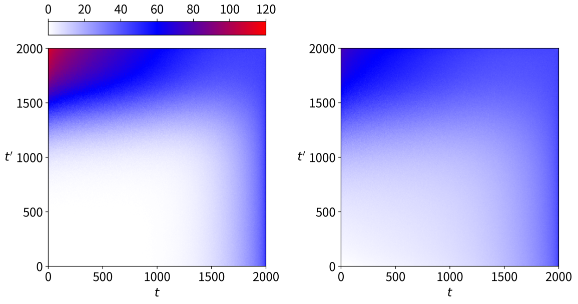
Table 2 shows the evaluation of the final outputs. As we saw in §5, the collapse problem occurred in `Base` and `OFE`, where BLEU, ROUGE-L and BERTScore deteriorated in the course of sampling. Their final results were roughly comparable performance to DiffuSeq as expected, since DiffuSeq is largely equivalent to the conventional model in Figure 1. However, `OFE`'s BERTScore is lower than DiffuSeq across all datasets. It suggests that the OFE may cause more severe collapse than the MLP-based method. `OFE+STDL` showed essentially comparable performance to Difformer, although it occasionally underperformed Difformer when combined with MBR.

`OFE+STDL` without MBR was substantially better than `OFE` with MBR. It is remarkable that the STDL achieves better performance without prolonged generation times, the drawback of MBR.

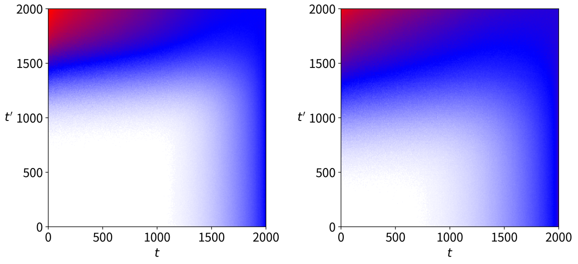
Figure 6 presents the assessment of intermediate outputs in the reverse process. In contrast to `OFE`, `OFE+STDL` consistently maintained generation quality throughout the reverse process. Moreover, on Quora Question Pairs, the STDL not only suppressed the collapse but also entirely improved sample quality.

The diversity unfortunately fell with STDL. However, as shown in Figure 6, the improvement in Self-BLEU of `OFE`'s output progresses in tandem with a decline in BLEU and BERTScore. This suggests that the diversity previously reported in diffusion language models may actually be an illusion arising from corrupted samples. Even if the collapse is partially tolerated, since Self-BLEU should ideally be comparable to BLEU, further improvements are required for text generation to achieve genuine diversity.

As a side note, employing MBR is also likely to reduce diversity. Comparing MBR = 10 and 20 of Difformer on Quora Question Pairs and `OFE+STDL` on Wiki-Auto, although the sequence-level similarity to the reference data measured by BLEU or ROUGE-L increased, the semantic similarity indicated by BERTScore remains unchanged.



(a) `OFE` (50 K)   (b) `OFE` (200 K)



(c) `OFE+STDL` (50 K)   (d) `OFE+STDL` (500 K)

Quora Question Pairs

Figure 7: Temporal Ambiguity Score of the checkpoints at the training steps indicated in parentheses.

	Checkpoint Step
`Base`	500 K
`OFE`	200 K
`OFE+STDL`	500 K

Quora Question Pairs

Table 3: The training step at which the evaluation checkpoint was saved.

7.3 Ambiguity across Timesteps

As observed in §5, the collapse problem in `OFE` becomes more severe as training progresses. To analyze this phenomenon, we compare the TAS between checkpoints that do and do not exhibit collapse. Figure 7 depicts the TAS for these checkpoints of `OFE` and `OFE+STDL`. In the early stage of training, the ambiguity is low for small timesteps and high for large timesteps as expected. However, as training of `OFE` advances, the ambiguity increases for small timesteps while decreasing for large timesteps. This suggests that the conventional loss function excessively focuses on constructing timestep embeddings for high noise levels while neglecting those for small noise levels. By contrast, in `OFE+STDL`, the TAS remains relatively stable throughout training, preserving the distinguishability of the non-collapsing embeddings space.

7.4 Training Efficiency

Table 3 presents the training steps corresponding to the evaluation checkpoints. `OFE` was trained faster than `Base`, even though the former achieved performance comparable to the latter. We also observed that the STDL led to slower convergence.

8 Conclusion

In this work, we investigated the collapse problem through careful observation of the emergence of the phenomenon and the lens of how timestep embeddings influence word embeddings. To address this challenge, we proposed a principled dimensionality reduction technique and a regularization method that acts on both embeddings. Our methods are simple and easy to employ, yet dramatically mitigate the collapse problem. In addition, we revisited generation diversity of diffusion language models, and suggested its intrinsic difficulty.

9 Limitations

One limitation of our study is that it remains unclear whether the STDL mitigates only the collapse problem or may also reduce the generation of desirable diversity. That said, our findings call for further investigation into timestep embeddings for diffusion models and diversity in text generation.

Acknowledgments

This work was partially supported by KIOXIA Corporation.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021. Structured Denoising Diffusion Models in Discrete State-Spaces. In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc.

DataCanary, hilfialkaff, Lili Jiang, Meg Risdal, Nikhil Dandekar, and tomtung. 2017. Quora Question Pairs. Kaggle.

Zhujin Gao, Junliang Guo, Xu Tan, Yongxin Zhu, Fang Zhang, Jiang Bian, and Linli Xu. 2024. Empowering Diffusion Models on the Embedding Space for Text Generation. 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 4664–4683. Association for Computational Linguistics.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *International Conference on Learning Representations*.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. 2023. DiffusionBERT: Improving Generative Masked Language Models with Diffusion Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4521–4534. Association for Computational Linguistics.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF Model for Sentence Alignment in Text Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960. Association for Computational Linguistics.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. DiffWave: A Versatile Diffusion Model for Audio Synthesis. In *International Conference on Learning Representations*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 169–176. Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. 2022. Diffusion-LM Improves Controllable Text Generation. In *Advances in Neural Information Processing Systems*, volume 35, pages 4328–4343. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Fei Huang, and Songfang Huang. 2024. Text Diffusion Model with Encoder-Decoder Transformers for Sequence-to-Sequence Generation. 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 22–39. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A Benchmarking Platform for Text Generation Models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100. Association for Computing Machinery.

Investigating Psychometric Predictive Power of Syntactic Attention

Ryo Yoshida
The University of Tokyo
yoshiryo0617@g.ecc.u-tokyo.ac.jp

Yushi Sugimoto
The University of Osaka
sugimoto.yushi.hmt@osaka-u.ac.jp

Yohei Oseki
The University of Tokyo
oseki@g.ecc.u-tokyo.ac.jp

Abstract

In computational psycholinguistics, Merkx and Frank (2021) demonstrated that surprisal values from Transformers exhibit a closer fit to measures of human reading effort than those from Recurrent Neural Networks (RNNs), suggesting that Transformers' attention mechanisms may capture cue-based retrieval-like operations in human sentence processing. Meanwhile, explicit integration of syntactic structures has been shown to improve language models' ability to model human sentence processing—for example, Hale et al. (2018) demonstrated that Recurrent Neural Network Grammars (RNNGs), which integrate RNNs with explicit syntactic structures, account for human brain activities that vanilla RNNs cannot capture. In this paper, we investigate the psychometric predictive power of Composition Attention Grammars (CAGs), which integrate Transformers with explicit syntactic structures, to test whether they provide a better fit to human reading times than both vanilla Transformers and RNNGs. We hypothesized that CAGs' syntactic attention mechanisms capture cue-based retrieval-like operations over syntactic memory representations—operations that may be involved in human sentence processing. The results of our strictly controlled experiments demonstrate that CAGs outperformed vanilla Transformers and RNNGs, suggesting that syntactic attention mechanisms of CAGs may serve as a mechanistic implementation of cue-based retrieval from syntactic memory.

1 Introduction

In computational psycholinguistics, language models (LMs) developed in Natural Language Processing (NLP) have been evaluated for their ability to model human sentence processing. Recurrent Neural Networks (RNNs; Elman, 1990), which process sequential representations recurrently, have traditionally been considered a practical implementation that demonstrates strong correspondence with human sentence processing, with their surprisal values successfully correlating with human reading times (Goodkind and Bicknell, 2018) and brain activities (Frank et al., 2015). Recently, Transformers (Vaswani et al., 2017), which have achieved state-of-the-art results on various downstream tasks, have also been tested for their power to predict human reading effort. Merkx and Frank (2021) demonstrated that Transformers outperformed RNNs in predicting human reading times and brain activities, suggesting that Transformers' attention mechanisms may provide a computational parallel to cue-based retrieval (Van Dyke and Lewis, 2003), a theory of human memory retrieval proposed in psycholinguistics.

While RNNs and Transformers primarily process sequential representations, the previous literature on computational psycholinguistics has empirically shown that explicit integration of syntactic structures can significantly improve LMs' ability to model human sentence processing. For instance, Hale et al. (2018) showed that Recurrent Neural Network Grammars (RNNGs; Dyer et al., 2016), which integrate RNNs with explicit syntactic structures, capture variance in human brain activities that cannot be accounted for by vanilla RNNs.[1]

Given that (i) Transformers may capture cue-based retrieval-like operations in human sentence processing and (ii) LMs integrated with explicit syntactic structures may capture variance in human syntactic processing, we investigate whether the integration of these two approaches might provide a better fit to measures of human reading effort

---

[1] More recently, Wolfman et al. (2024) showed that surprisal values from Transformer Grammars (TGs; Sartran et al., 2022), which integrate Transformers with explicit syntactic structures, also explain human brain activities that vanilla Transformers cannot predict. While their work and ours are similar in that both investigate the advantage of explicit integration of syntactic structures on Transformers, we additionally investigate the advantage of syntactic attention over syntactic recurrence, a research question not addressed in Wolfman et al. (2024).

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 407–418
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

than LMs employing either approach in isolation. Specifically, we investigate the psychometric predictive power of Composition Attention Grammars (CAGs; Yoshida and Oseki, 2022), which integrate Transformers with explicit syntactic structures, to test whether they provide a better fit to human reading times than both vanilla Transformers and RNNGs. We hypothesize that CAGs' syntactic attention mechanisms capture cue-based retrieval-like operations over syntactic memory representations—operations that may be involved in human sentence processing. The results of our controlled experiments demonstrate that CAGs outperformed vanilla Transformers and RNNGs, suggesting that syntactic attention mechanisms of CAGs may serve as a mechanistic implementation of cue-based retrieval from syntactic memory.[2]

2 Background

2.1 Psychometric predictive power

In psycholinguistics, it is well established that humans predict the next word during sentence processing (i.e., expectation-based theories), and the less predictable the next word is, the more effort is required to process it. The computational psycholinguistics literature (Hale, 2001; Levy, 2008) quantifies this predictability as *surprisal*, the negative log probability of a word given the context:

$$\text{surprisal} = -\log p(\text{word}|\text{context}). \quad (1)$$

Previous work has employed this information-theoretic complexity metric to link LMs' probability estimates with human reading effort (Smith and Levy, 2013; Goodkind and Bicknell, 2018). Building upon this paradigm, the computational psycholinguistics community has investigated LMs with high psychometric predictive power—i.e., LMs that can compute surprisal values with trends similar to measures of human reading effort—by comparing surprisal from various models with reading times or brain activities from humans (Frank and Bod, 2011; Fossum and Levy, 2012; Frank et al., 2015; Hale et al., 2018; Brennan and Hale, 2019; Wilcox et al., 2020; Brennan et al., 2020; Merkx and Frank, 2021; Kuribayashi et al., 2022; Wolfman et al., 2024, *inter alia*).

2.2 Sequential recurrence vs. sequential attention

RNNs (Elman, 1990) process sequential information (i.e., word embeddings) in a recurrent manner; they maintain a single vector representing a "context" and, at each time step, update this context vector with the embedding of the current input word (implementing sequential recurrence; Figure 1a). In contrast, recently introduced Transformers (Vaswani et al., 2017) employ an attention mechanism; they maintain all previous word embeddings and, at each time step, generate a context vector by selectively attending to them (implementing sequential attention; Figure 1b). Taking advantage of direct access to previous information, Transformers have been shown to outperform RNNs in various NLP tasks (cf. Wang et al., 2018, 2020).

Recently, the computational psycholinguistics community has also investigated whether Transformers have an advantage over RNNs in psychometric predictive power. Merkx and Frank (2021) compared Transformers and RNNs on their predictive power for human reading times and brain activities. The results showed that Transformers generally outperformed RNNs, suggesting that sequential attention, implemented by Transformers, captures aspects of human reading effort that sequential recurrence, implemented by RNNs, cannot account for.

Based on these findings, Merkx and Frank (2021) argued that the explained effort might be attributed to cue-based retrieval-like operations during human sentence processing (Van Dyke and Lewis, 2003). The cue-based retrieval theory posits that human sentence processing involves memory retrieval, where elements are retrieved from working memory based on cues provided by the current input word. Merkx and Frank's (2021) argument was that Transformers' attention mechanism—selective attention to previous word embeddings based on Queries from current input and Keys from previous words—might serve as a mechanistic implementation of this cue-based memory retrieval. Consequently, surprisal values from the attention mechanism would show similar trends to human reading effort, serving as the *causal bottleneck* (Levy, 2008).[3]

---

[2] Code for reproducing our results is available at `https://github.com/osekilab/CAG-EyeTrack`.

[3] Complementary research has examined the relationship between attention-based metrics (such as attention entropy) and human reading effort to assess the validity of attention mechanisms as a mechanistic implementation of cue-based retrieval (Ryu and Lewis, 2021; Oh and Schuler, 2022).

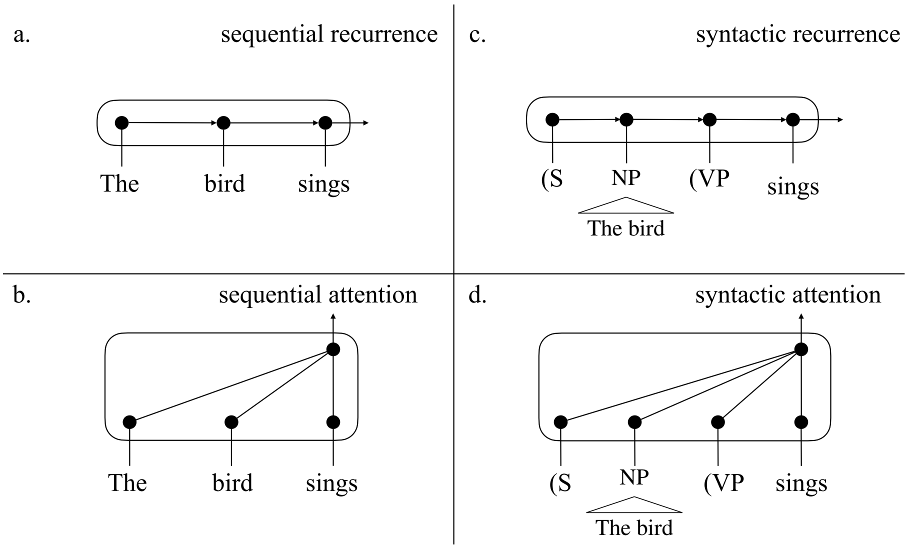


Figure 1: Four types of architectures. Previous work has investigated three types of architectural comparisons: (i) recurrence vs. attention in sequential architectures (a vs. b), (ii) sequential vs. syntactic in recurrent architectures (a vs. c), and (iii) sequential vs. syntactic in attention architectures (b vs. d). In this paper, we complete this comparison framework by directly comparing recurrence vs. attention in syntactic architectures (c vs. d).

More recently, Michaelov et al. (2021) replicated Merkx and Frank's (2021) results and presented additional analysis suggesting that Transformers can better capture human semantic facilitation effects than RNNs.

2.3 Sequential vs. syntactic

Although RNNs and Transformers have shown non-negligible results in psychometric predictive power, these architectures are fundamentally "sequential" models that process sequential information—without explicitly modeling the hierarchical syntactic structures of natural languages. The distinction between vanilla LMs and *syntactic LMs* such as RNNGs lies in this structural aspect—syntactic LMs not only generate a word sequence but also explicitly construct its underlying syntactic structure. Specifically, syntactic LMs jointly generate sentences and their syntactic structures through next-action prediction for the following three actions:

- `(X`: Generate a non-terminal symbol `(X`, where `X` represents a phrasal tag (e.g., `NP`). The vector representing the phrasal tag is placed on top of the *stack*, which maintains a list of vectors corresponding to the current context in syntactic LMs.

- `w`: Generate a terminal symbol `w`, where `w` represents a word (e.g., `bird`). The vector representing the word is placed on top of the stack.

- `)`: Close the most recent open non-terminal symbol. The vectors that constitute the closed phrase (i.e., the closed phrasal tag and its constituent vectors) are typically combined into a single vector representation using a *composition function* and placed on top of the stack. However, some syntactic LMs omit this composition step and simply place a vector representing the phrase closure on top of the stack (henceforth, we denote this type of syntactic LM with the subscript $_{-\text{comp}}$).

Computational psycholinguistics studies have shown that syntactic LMs outperform their vanilla LM counterparts in psychometric predictive power, suggesting that syntactic LMs can capture non-trivial variance in human syntactic processing. For instance, RNNGs, which recurrently summarize the stack state using RNNs (Dyer et al., 2015) (implementing syntactic recurrence; Figure 1c), can predict patterns in human brain activity (Hale et al., 2018) and human reading time (Yoshida et al., 2021) that vanilla RNNs cannot. Hale et al. (2018) also showed the advantage of the composition function, demonstrating that RNNGs$_{-\text{comp}}$ cannot explain the brain activity that RNNGs can.

More recently, Wolfman et al. (2024) demonstrated that Transformer Grammars (TGs; Sartran et al., 2022), which summarize the stack state by selectively attending to previous vectors using Transformers (implementing syntactic attention; Figure 1d), also explain human brain activities more successfully than vanilla Transformers.[4]

3 Syntactic recurrence vs. syntactic attention

As reviewed in Section 2, previous work has investigated three types of architectural comparisons: (i) recurrence vs. attention in sequential architectures (Merkx and Frank, 2021; Michaelov et al., 2021) (Figure 1a vs. 1b), (ii) sequential vs. syntactic in recurrent architectures (Hale et al., 2018; Yoshida et al., 2021) (Figure 1a vs. 1c), and (iii) sequential vs. syntactic in attention architectures (Wolfman et al., 2024) (Figure 1b vs. 1d). In this paper, we complete this comparison framework by directly comparing recurrence vs. attention in syntactic architectures (Figure 1c vs. 1d).

We hypothesize that syntactic attention—where previous vectors "in the stack" are selectively attended to based on Queries from current input and Keys from previous vectors—might show superior psychometric predictive power over syntactic recurrence by capturing cue-based retrieval-like operations over "syntactic memory representations"—operations that may be involved in human sentence processing. This hypothesis extends Merkx and Frank's (2021) argument that sequential attention (implemented by vanilla Transformers) outperforms sequential recurrence (implemented by RNNs), capturing cue-based retrieval-like operations over sequential memory representations.

LMs that implement syntactic attention include Transformer Grammars (TGs; Sartran et al., 2022) and Composition Attention Grammars (CAGs; Yoshida and Oseki, 2022). Both TGs and CAGs are syntactic LMs based on Transformers and employ composition functions. For our investigation, we employ CAGs for three reasons. First, CAGs' implementation includes word-synchronous beam search (Stern et al., 2017), an inference technique commonly used in computational psycholinguistics to model human local ambiguity resolution through parallel parsing (Hale et al., 2018; Sugimoto et al., 2024) (see Section 4.3 for details), whereas TGs lack this capability. Second, CAGs' probability estimation aligns more closely with human offline grammaticality judgments than TGs (Yoshida and Oseki, 2022). Third, CAGs employ bidirectional LSTMs for the composition function, which is the same implementation used in RNNGs, while TGs implement the composition function via attention masks. This design choice enables a more controlled comparison between syntactic recurrence and syntactic attention, as the architectures differ only in their stack summarization process.

4 Method

We evaluate four LMs that employ either selective attention or recurrent processing on word sequences or syntactic structures, comparing their psychometric predictive power for human reading times using the Zurich Cognitive Language Processing Corpus (ZuCo; Hollenstein et al., 2018). Following Hale et al. (2018), we also include degraded versions of syntactic LMs that lack the composition function. The following subsections describe our experimental settings in detail.

4.1 Language models

In our experiment, we trained LMs with strictly controlled hyperparameters following Yoshida and Oseki (2022), as their model sizes were made maximally comparable.

LSTM (sequential recurrence) Long Short-Term Memories (LSTMs; Hochreiter and Schmidhuber, 1997) are LMs that perform recurrent processing on word sequences. We used 2-layer LSTMs with 301 hidden and input dimensions (model size: 16.59M).[5]

RNNG (syntactic recurrence) Recurrent Neural Network Grammars (RNNGs; Dyer et al., 2016) are LMs that perform recurrent processing on syntactic structures. RNNGs are equipped with a composition function based on bidirectional LSTMs. We used stack-only RNNGs (Kuncoro et al., 2018; Noji and Oseki, 2021) with 2-layer stack LSTMs with 276 hidden and input dimensions (model size: 16.61M).[6]

RNNG$_{-\text{comp}}$ (degraded syntactic recurrence) RNNGs$_{-\text{comp}}$ (Choe and Charniak, 2016; Hale et al., 2018) are a degraded version of RNNGs without the composition function. We used

---

[4] Yoshida et al. (2025) also demonstrated that attention entropy derived from TGs can predict human reading times more successfully than vanilla Transformers.

[5] We implemented LSTMs using the PyTorch package (`https://github.com/pytorch/pytorch`).

[6] `https://github.com/aistairc/rnng-pytorch`

RNNGs$_{-\text{comp}}$ with 2-layer LSTMs with 301 hidden and input dimensions (model size: 16.58M).

Transformer (sequential attention) Transformers (Radford et al., 2018) are LMs that perform selective attention on word sequences. We used 3-layer 4-head Transformers with 272 hidden and input dimensions (model size: 16.62M).[7]

CAG (syntactic attention) Composition Attention Grammars (CAGs; Yoshida and Oseki, 2022) are LMs that perform selective attention on syntactic structures. CAGs are equipped with a composition function based on bidirectional LSTMs. We used 3-layer 4-head CAGs with 256 hidden and input dimensions (model size: 16.57M).[8]

CAG$_{-\text{comp}}$ (degraded syntactic attention) CAGs$_{-\text{comp}}$ (Qian et al., 2021) are a degraded version of CAGs without the composition function. We used 3-layer 4-head CAGs$_{-\text{comp}}$ with 272 hidden and input dimensions (model size: 16.63M).[9]

4.2 Training data

All LMs were trained using BLLIP-LG, which comprises 1.8M sentences and 42M tokens sampled from the Brown Laboratory for Linguistic Information Processing 1987-89 Corpus Release 1 (BLLIP; Charniak et al., 2000). The train-dev-test split followed Hu et al. (2020). Following Qian et al. (2021), sentences were tokenized into subwords using a Byte Pair Encoding tokenizer (Sennrich et al., 2016) from the Huggingface Transformers package (Wolf et al., 2020).

All LMs were trained at the sentence level: LSTMs and Transformers were trained on terminal subwords, whereas RNNGs, RNNG$_{-\text{comp}}$, CAGs, and CAG$_{-\text{comp}}$ were trained on both terminal subwords and syntactic structures, which were parsed by Hu et al. (2020) using a state-of-the-art constituency parser (Kitaev and Klein, 2018). All LMs shared the same training hyperparameters: a learning rate of 10^{-3}, a dropout rate of 0.1, the Adam optimizer (Kingma and Ba, 2015), and a minibatch size of 256. Training was conducted for 15 epochs. We selected the checkpoint with the lowest loss on the development set for evaluation and conducted experiments three times with different random seeds.

4.3 Eye tracking data

We used reading times from the Zurich Cognitive Language Processing Corpus (ZuCo; Hollenstein et al., 2018) to evaluate whether LMs can successfully predict human reading effort. ZuCo is a collection of single sentences from the Stanford Sentiment Treebank and the Wikipedia relation extraction corpus, accompanied by simultaneous eye-tracking and electroencephalography (EEG) recordings from 12 native English speakers. Although ZuCo comprises data from both normal reading and task-specific reading tasks, we used only 700 sentences from the natural reading task, following previous work (e.g., Hollenstein et al., 2021). During the natural reading task, sentences were displayed one by one, and participants read them at their own pace. During preprocessing by Hollenstein et al. (2018), fixations that were (i) shorter than 100 ms or (ii) recorded when EEG amplitude exceeded ±90 µV were removed due to irrelevance to reading activity or data quality concerns.

In this paper, first-pass reading time (the sum of all fixation times on a word before the eye moves away from it) was used as the prediction target.[10] Following the convention of psycholinguistic studies, we excluded words with missing values (e.g., non-fixations) or at sentence-initial and sentence-final positions from our statistical analysis. We further removed words that were out of vocabulary (OOV) in the large corpus (Wikitext-2; Merity et al., 2017) or words following OOV words, as frequency values are required for our baseline regression model. Consequently, 80,853 data points were included in the statistical analysis out of 161,597 total data points. The high proportion of deleted data points during preprocessing was mainly due to the large number of missing values (52,240 data points).

In previous computational psycholinguistic research, there was often a mismatch between LMs' processing level and human data collection procedures—for instance, LMs trained at the sentence level were evaluated against human data col-

---

[7] We implemented Transformers using the Huggingface Transformers package (`https://github.com/huggingface/transformers`).

[8] `https://github.com/osekilab/CAG`

[9] `https://github.com/IBM/transformers-struct-guidance`

[10] We first conduct validation using reading time as the most accessible and interpretable human data source, given that the specific event-related potential (ERP) components of EEG that would best reflect cue-based retrieval-like operations over syntactic memory representations remain to be determined.

lected during document-level reading (cf. Wilcox et al., 2020). In this paper, we address this gap by conducting more strictly controlled experiments using ZuCo, a corpus where eye-tracking data was recorded during sentence-level reading.[11]

Since only word sequences were input during surprisal calculation, we employed word-synchronous beam search (Stern et al., 2017) to infer syntactic structures for CAGs and RNNGs. Word-synchronous beam search retains a collection of the most likely syntactic structures given a partial word sequence and marginalizes their probabilities to approximate next-word probabilities. Hale et al. (2018) argued that the combination of syntactic LMs and word-synchronous beam search successfully captured human local ambiguity resolution during online sentence processing.[12]

4.4 Statistical analysis

We analyzed how well surprisal from each LM predicts human reading time, measuring improvements in regression model fit when adding surprisal values as predictors. For each LM, we included both the surprisal of the current word and the previous word to account for spillover effects (Mitchell, 1984).[13] As a measure of psychometric predictive power, we evaluated the per-token increase in log-likelihood (ΔLogLik) on the entire dataset. This evaluation was conducted for each random seed, and we report the mean psychometric predictive power with standard deviation.

Following previous studies such as Merkx and Frank (2021), the baseline regression model controlled for several predictors relevant to reading activity:

- `order` (integer): sentence display order during the reading task;
- `position` (integer): word position in the sentence;
- `length` and `prev_length` (integer): number of characters in the current and previous word;
- `freq` and `prev_freq` (continuous): log-transformed frequencies of the current and previous word.

Previous words' values were included for modeling the spillover effect. All numeric factors were z-transformed.

The baseline regression model was a linear mixed-effects model (Baayen et al., 2008) with these fixed effects and a by-subject random intercept:[14]

$$\begin{aligned}\log(\mathtt{RT}) \sim\ & \mathtt{order} + \mathtt{position} + \\ & \mathtt{length} + \mathtt{prev\_length} + \\ & \mathtt{freq} + \mathtt{prev\_freq} + \\ & (\mathtt{1|subj}). \end{aligned} \tag{2}$$

Before evaluating psychometric predictive power, we conducted baseline regression model-based data omission, removing data points beyond three standard deviations. This removed 559 data points, leaving 80,294 data points for the final statistical analysis.

4.5 Nested model comparison

We conducted nested model comparisons (Wurm and Fisicaro, 2014) to evaluate whether the differences in ΔLogLik are statistically significant. Specifically, we extended Equation 2 by adding surprisal values from two LMs versus adding surprisal values from only one LM, and tested the statistical significance of the deviance using the χ^2 test ($p \leq 0.05$). Following Aurnhammer and Frank (2019), we used surprisal values averaged across different random seeds for these nested model comparisons.

5 Results

5.1 Overall

The Psychometric Predictive Power (PPP, per-token ΔLogLik) of each LM is summarized in Figure 2. The psychometric predictive power averaged across different random seeds (the vertical axis) is plotted against the LMs investigated in this paper (the horizontal axis). Error bars denote standard deviations across random seeds. We confirmed that the psychometric predictive power was statistically

---

[11] An alternative approach would be to train LMs at the document level and evaluate them on document-level reading data. However, we adopt the sentence-level setting because syntactic LMs are conventionally trained on sentences, and RNNGs and CAGs lack implementations applicable to document-level training.

[12] We set the action beam size to 100, word beam size to 10, and fast-track to 1. Word beam size corresponds to the number of syntactic structures to be marginalized.

[13] Following the convention of previous studies (e.g., Wilcox et al., 2020; Kuribayashi et al., 2021), the word-level surprisal was calculated as the cumulative surprisal of its constituent subwords.

[14] We implemented the regression model using the lme4 package (Bates et al., 2015) in R (R Core Team, 2024).

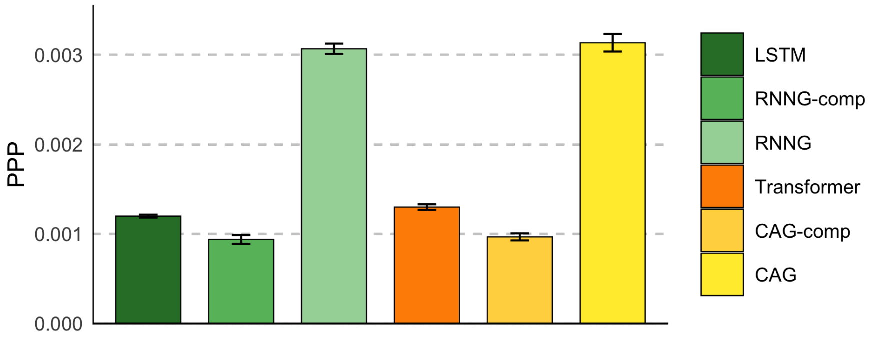


Figure 2: Psychometric Predictive Power (PPP, per-token ΔLogLik) of each LM. The psychometric predictive power averaged across different random seeds (vertical axis) is plotted against the LMs investigated in this paper (horizontal axis). Error bars denote standard deviations across random seeds.

significant for all LMs under nested model comparisons against the baseline regression model, and the direction was appropriate for reading time—that is, higher surprisal values corresponded to longer reading times. The results demonstrated that CAGs achieved the highest psychometric predictive power: CAG > RNNG > Transformer > LSTM > CAG$_{-\text{comp}}$ > RNNG$_{-\text{comp}}$, showing that the architecture performing syntactic attention captures the most variance in human reading time.

Reproduction of sequential recurrence vs. sequential attention In our experiment, Transformers outperformed LSTMs in psychometric predictive power. To confirm that this difference is statistically significant, the result of the nested model comparison is shown in the top block of Table 1. The nested model comparison revealed that Transformers significantly outperformed LSTMs, corroborating Merkx and Frank's (2021) finding that Transformers, which implement sequential attention, capture variance in human reading effort that RNNs, which implement sequential recurrence, cannot.[15]

Reproduction of sequential vs. syntactic In our experiment, RNNGs and CAGs outperformed LSTMs and Transformers, respectively. To confirm that these differences are statistically significant, the results of nested model comparisons are shown in the middle block of Table 1. The nested model comparisons revealed that RNNGs and CAGs significantly outperformed LSTMs and Transformers, respectively, supporting the findings of Hale et al. (2018) and Wolfman et al. (2024) that syntactic LMs can account for human reading effort that vanilla LMs cannot predict.

In addition, RNNGs and CAGs also significantly outperformed RNNGs$_{-\text{comp}}$ and CAGs$_{-\text{comp}}$, respectively, corroborating Hale et al.'s (2018) argument that the composition function is crucial for syntactic LMs to capture human syntactic processing. As a side note, RNNGs$_{-\text{comp}}$ and CAGs$_{-\text{comp}}$ underperformed LSTMs and Transformers, respectively. This implies that stack representations without the composition function not only harm the ability to account for syntactic processing but also cause a loss in simulating general human predictive processing. Hale et al. (2018) also showed a null result when comparing the psychometric predictive power of RNNGs$_{-\text{comp}}$ to that of LSTMs.

Syntactic recurrence vs. syntactic attention In our experiment, CAGs outperformed RNNGs in the absolute value of psychometric predictive power. To confirm that the difference between CAGs and RNNGs is statistically significant, the result of the nested model comparison is shown in the bottom block of Table 1. The nested model comparison revealed that CAGs significantly outperformed RNNGs, suggesting that CAGs, which implement syntactic attention, can successfully capture variance in human reading time that RNNGs, which implement syntactic recurrence, cannot account for.

---

[15] Incidentally, Merkx and Frank (2021) found the advantage of Transformers on self-paced reading times and EEG but obtained mixed results on first-pass reading time. Our more definitive findings may be attributed to our strictly controlled experimental settings, where Transformer advantages could become more consistently observable.

	χ^2	df	p
Sequential recurrence vs. sequential attention			
LSTM < TF	16.75	2	**0.00023**
Sequential vs. syntactic			
LSTM < RNNG	315.7	2	**<0.0001**
TF < CAG	308.5	2	**<0.0001**
RNNG$_{-\text{c.}}$ < RNNG	369.8	2	**<0.0001**
CAG$_{-\text{c.}}$ < CAG	372.0	2	**<0.0001**
Syntactic recurrence vs. syntactic attention			
RNNG < CAG	11.42	2	**0.00331**

Table 1: Results of nested model comparisons from three perspectives: (i) reproduction of sequential recurrence vs. sequential attention, (ii) reproduction of sequential vs. syntactic, and (iii) syntactic recurrence vs. syntactic attention. TF and $_{-\text{c.}}$ indicate Transformer and $_{-\text{comp}}$, respectively.

5.2 Longer and shorter sentences

To investigate under what conditions syntactic attention has an advantage over syntactic recurrence, we split the data points in ZuCo into two subsets based on sentences longer or shorter than the average sentence length, following Merkx and Frank (2021). Merkx and Frank (2021) conducted this analysis expecting that longer sentences could accentuate Transformers' advantage of direct access to previous information. The longer and shorter subsets include 37,578 and 43,275 data points, respectively. We removed 601 and 703 data points that were beyond three standard deviations, leaving 37,307 and 42,997 data points for the final statistical analysis, respectively.

The psychometric predictive power of CAGs and RNNGs on longer and shorter sentences is shown in Figure 3. The results show that CAGs and RNNGs achieve comparable psychometric predictive power on shorter sentences, but CAGs outperformed RNNGs on longer sentences. To confirm that these differences are statistically significant, the results of nested model comparisons are shown in Table 2. The nested model comparisons revealed that CAGs significantly outperformed RNNGs only on longer sentences, consistent with their performance on the complete dataset.

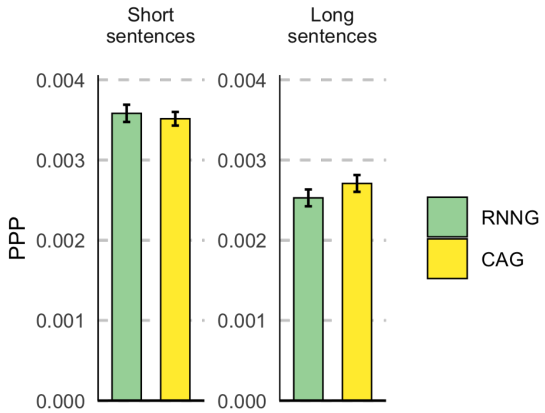


Figure 3: Psychometric predictive power (PPP, per-token ΔLogLik) of CAGs and RNNGs on longer and shorter sentences. The psychometric predictive power averaged across different random seeds (vertical axis) is plotted against the LMs (horizontal axis). Error bars denote standard deviations across random seeds.

	χ^2	df	p
Short sentences			
RNNG < CAG	0.8359	2	0.6584
Long sentences			
RNNG < CAG	14.793	2	**0.0006133**

Table 2: Results of nested model comparisons on longer and shorter subsets of ZuCo

6 Discussion

In this paper, we reproduced the results of (i) sequential recurrence vs. sequential attention (cf. Merkx and Frank, 2021), (ii) sequential vs. syntactic (cf. Hale et al., 2018; Wolfman et al., 2024), and (iii) demonstrated that CAGs, which implement syntactic attention, achieve higher psychometric predictive power than both vanilla Transformers and RNNGs. Given that Merkx and Frank (2021) and Hale et al. (2018) suggest that attention mechanisms and syntactic LMs can serve as mechanistic implementations of human cue-based retrieval and syntactic processing, respectively, our results suggest that syntactic attention in CAGs may serve as a mechanistic implementation of cue-based retrieval from syntactic memory. This interpretation is also consistent with psycholinguistic studies demonstrating that memory costs derived from syntactic structures successfully predict reading times and brain activities (Isono, 2024; Shain et al., 2022)—findings that support the cognitive plausibility of syntactic memory representations and operations over them.

Furthermore, the analyses of longer versus shorter sentences suggest that cue-based retrieval-like operations over syntactic memory representations may become more prominent when processing longer sentences. Merkx and Frank (2021) demonstrated that Transformers' superior psychometric predictive power over RNNs was particularly pronounced on longer sentences, suggesting that retrieval operations may be especially important when accessing information from linearly distant words. While both CAGs and RNNGs can maintain information from linearly distant words through their composition functions, the direct access afforded by attention mechanisms nevertheless provides additional advantages as sentences get longer.

Interestingly, Wilcox et al. (2018) and Oh et al. (2021) found that syntactic LMs (i.e., RNNGs) underperformed LSTMs or Transformers in modeling human reading times and brain activities, contradicting the advantages observed by Hale et al. (2018), Wolfman et al. (2024), and our sequential vs. syntactic results. One potential explanation for these discrepancies lies in experimental control: while Wilcox et al. (2018) and Oh et al. (2021) compared LMs with varying model sizes, Hale et al. (2018), Wolfman et al. (2024) and our experiment all employed LMs with maximally comparable model sizes. Our approach further extends this methodology by aligning LMs' processing units with human data collection procedures at the sentence level. These results highlight the critical role of controlled experimental design, especially when comparing minimally different architectures.

7 Conclusion

In this paper, we investigated the psychometric predictive power of Composition Attention Grammars (CAGs) through strictly controlled experiments. Our results demonstrated that CAGs outperformed both vanilla Transformers and RNNGs, suggesting that syntactic attention may serve as a mechanistic implementation of cue-based retrieval from syntactic memory. Further analyses revealed that this result is primarily driven by improved performance on longer sentences, indicating that cue-based retrieval-like operations over syntactic memory representations became increasingly important as sentences got longer.

Limitations

There are several limitations to this study. First, although we utilized CAGs as a model of syntactic attention, TGs could also serve as an alternative. While our choice of CAGs was motivated by (i) their word-synchronous beam search capability, (ii) better alignment to human offline grammaticality judgments, and (iii) their use of bidirectional LSTMs for the composition function (see Section 3), whether our positive results for syntactic attention generalize to TGs remains an open question.

Second, our experiments were based solely on reading time data from ZuCo. As noted earlier, we chose reading time as the most accessible and interpretable human data source, given that the specific event-related potential (ERP) components of EEG that would best reflect cue-based retrieval-like operations over syntactic memory representations remain to be determined. Future research should explore which ERP components might be most sensitive to these operations and extend the evaluation to additional measures of human sentence processing.

Third, while our sentence-level analysis provided technical advantages for controlled comparisons, extending these syntactic LMs to document-level processing would be valuable for future research, as this would enable controlled experiments on additional datasets (e.g., the Natural Stories corpus; Futrell et al., 2018).

Acknowledgments

We would like to thank Shinnosuke Isono for his helpful suggestions on the manuscript. We also thank anonymous CoNLL 2025 reviewers for their insightful reviews. This work was supported by JSPS KAKENHI Grant Number 24H00087, Grant-in-Aid for JSPS Fellows JP24KJ0800, JST PRESTO Grant Number JPMJPR21C2, and JST SPRING Grant Number JPMJSP2108.

References

Christoph Aurnhammer and Stefan L. Frank. 2019. Comparing Gated and Simple Recurrent Neural Network Architectures as Modelsof Human Sentence Processing. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 41(0).

R. H. Baayen, D. J. Davidson, and D. M. Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of Memory and Language*, 59(4):390–412.

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.

Jonathan R. Brennan, Chris Dyer, Adhiguna Kuncoro, and John T. Hale. 2020. Localizing syntactic predictions using recurrent neural network grammars. *Neuropsychologia*, 146:107479.

Jonathan R. Brennan and John T. Hale. 2019. Hierarchical structure guides rapid linguistic predictions during naturalistic listening. *PLOS ONE*, 14(1):e0207741.

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. BLLIP 1987-89 WSJ Corpus Release 1.

Do Kook Choe and Eugene Charniak. 2016. Parsing as Language Modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas. Association for Computational Linguistics.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-Based Dependency Parsing with Stack Long Short-Term Memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 334–343, Beijing, China. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

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

Victoria Fossum and Roger Levy. 2012. Sequential vs. Hierarchical Syntactic Models of Human Incremental Sentence Processing. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, pages 61–69, Montréal, Canada. Association for Computational Linguistics.

Stefan L. Frank and Rens Bod. 2011. Insensitivity of the Human Sentence-Processing System to Hierarchical Structure. *Psychological Science*, 22(6):829–834.

Stefan L. Frank, Leun J. Otten, Giulia Galli, and Gabriella Vigliocco. 2015. The ERP response to the amount of information conveyed by words in sentences. *Brain and Language*, 140:1–11.

Richard Futrell, Edward Gibson, Harry J. Tily, Idan Blank, Anastasia Vishnevetsky, Steven Piantadosi, and Evelina Fedorenko. 2018. The Natural Stories Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

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, Salt Lake City, Utah. Association for Computational Linguistics.

John Hale. 2001. A Probabilistic Earley Parser as a Psycholinguistic Model. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation*, 9(8):1735–80.

Nora Hollenstein, Federico Pirovano, Ce Zhang, Lena Jäger, and Lisa Beinborn. 2021. Multilingual Language Models Predict Human Reading Behavior. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 106–123, Online. Association for Computational Linguistics.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. ZuCo, a simultaneous EEG and eye-tracking resource for natural sentence reading. *Scientific Data*, 5(1):180291.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A Systematic Assessment of Syntactic Generalization in Neural Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Shinnosuke Isono. 2024. Category Locality Theory: A unified account of locality effects in sentence comprehension. *Cognition*, 247:105766.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436, Melbourne, Australia. Association for Computational Linguistics.

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

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower Perplexity is Not Always Human-Like. 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 5203–5217, Online. Association for Computational Linguistics.

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

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*.

Danny Merkx and Stefan L. Frank. 2021. Human Sentence Processing: Recurrence or Attention? In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 12–22, Online. Association for Computational Linguistics.

James A. Michaelov, Megan D. Bardolph, Seana Coulson, and Benjamin Bergen. 2021. Different kinds of cognitive plausibility: Why are transformers better than RNNs at predicting N400 amplitude? *Proceedings of the Annual Meeting of the Cognitive Science Society*, 43(43).

D. C. Mitchell. 1984. An Evaluation of Subject-Paced Reading Tasks and Other Methods for Investigating Immediate Processes in Reading 1. In *New Methods in Reading Comprehension Research*. Routledge.

Hiroshi Noji and Yohei Oseki. 2021. Effective Batching for Recurrent Neural Network Grammars. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.

Byung-Doh Oh, Christian Clark, and William Schuler. 2021. Surprisal Estimators for Human Reading Times Need Character Models. 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 3746–3757, Online. Association for Computational Linguistics.

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.

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural Guidance for Transformer Language Models. 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 3735–3745, Online. Association for Computational Linguistics.

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving Language Understanding by Generative Pre-Training. page 12.

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.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Cory Shain, Idan A. Blank, Evelina Fedorenko, Edward Gibson, and William Schuler. 2022. Robust effects of working memory demand during naturalistic language comprehension in language-selective cortex. *Journal of Neuroscience*, 42(39):7412–7430.

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

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective Inference for Generative Neural Parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Yushi Sugimoto, Ryo Yoshida, Hyeonjeong Jeong, Masatoshi Koizumi, Jonathan R. Brennan, and Yohei Oseki. 2024. Localizing Syntactic Composition with Left-Corner Recurrent Neural Network Grammars. *Neurobiology of Language*, 5(1):201–224.

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

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

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2020. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *Preprint*, arXiv:1905.00537.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

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.

Ethan G. Wilcox, Jon Gauthier, Jennifer Hu, Peng Qian, and Roger P. Levy. 2020. On the Predictive Power of Neural Language Models for Human Real-TimeComprehension Behavior. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 42.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michael Wolfman, Donald Dunagan, Jonathan Brennan, and John Hale. 2024. Hierarchical syntactic structure in human-like language models. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 72–80, Bangkok, Thailand. Association for Computational Linguistics.

Lee H. Wurm and Sebastiano A. Fisicaro. 2014. What residualizing predictors in regression analyses does (and what it does not do). *Journal of Memory and Language*, 72:37–48.

Ryo Yoshida, Shinnosuke Isono, Kohei Kajikawa, Taiga Someya, Yushi Sugimoto, and Yohei Oseki. 2025. If Attention Serves as a Cognitive Model of Human Memory Retrieval, What is the Plausible Memory Representation? *Preprint*, arXiv:2502.11469.

Ryo Yoshida, Hiroshi Noji, and Yohei Oseki. 2021. Modeling Human Sentence Processing with Left-Corner Recurrent Neural Network Grammars. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2964–2973, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ryo Yoshida and Yohei Oseki. 2022. Composition, Attention, or Both? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5822–5834, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

A Continuous Approach to Metaphorically Motivated Regular Polysemy in Language Models

Anna Temerko and **Marcos Garcia** and **Pablo Gamallo**
CiTIUS – Research Center in Intelligent Technologies
Universidade de Santiago de Compostela
`{a.temerko,marcos.garcia.gonzalez,pablo.gamallo}@usc.gal`

Abstract

Linguistic accounts show that a word's polysemy structure is largely governed by systematic sense alternations that form overarching patterns across the vocabulary. While psycholinguistic studies confirm the psychological validity of regularity in human language processing, in the research on large language models (LLMs) this phenomenon remains largely unaddressed. Revealing models' sensitivity to systematic sense alternations of polysemous words can give us a better understanding of how LLMs process ambiguity and to what extent they emulate representations in the human mind. For this, we employ the measures of surprisal and semantic similarity as proxies of human judgment on the acceptability of novel senses. We focus on two aspects that have not received much attention previously – metaphorically motivated patterns and the continuous nature of regularity. We find evidence that surprisal from language models represents regularity of polysemic extensions in a human-like way, discriminating between different types of senses and varying regularity degrees, and overall strongly correlating with human acceptability scores.

1 Introduction

Polysemy, a linguistic phenomenon whereby a word is associated with multiple related senses, is fundamental to language. As most lexical words are polysemes to varying degrees (Zipf, 1945; Durkin and Manning, 1989; Haber and Poesio, 2024), this form of ambiguity remains a challenge for NLP. However, recent studies show that current language models (LMs) based on Transformers are able to reveal the degree of a word's polysemy, meaningfully cluster word senses, distinguish homonymy from polysemy or perform superior word sense disambiguation (see Garí Soler and Apidianaki, 2021; Li and Joanisse, 2021; Nair et al., 2020; Wiedemann et al., 2019 for each of the above).

We focus on the topic that received less attention in LM research – the regularity dimension of polysemy and its continuous nature. The definition and scope of regular polysemy vary depending on the linguistic theory. The widely cited definition has been proposed by Apresjan (1974, p. 16) and states that "Polysemy of the word A with the meanings a_i and a_j is called regular if [...] there exists at least one other word B with the meanings b_i and b_j, which are semantically distinguished from each other in exactly the same way as a_i and a_j [...]". Pustejovsky's (1991) approach, also frequently adopted, frames regular polysemy as an ability of words that belong to one semantic type to act as members of another, behaving predictably, unlike irregular (accidental) polysemes.

To exemplify, the word *star* shows irregularity in its sense structure referring to a celestial body or a highly celebrated, famous person. To our knowledge, such meaning extension is not shared with any other lexical units of English. In contrast, regular polysemy forms patterns of meaning structure across the vocabulary. Some of the widely used examples of such patterns are ANIMAL - MEAT pattern (instantiated by *chicken* or *salmon*) or CONTAINER - CONTENT (e.g. *cup*, *glass*), as exemplified with sentences (1) and (2) below.

(1) a. We keep our chicken in the backyard.
b. Chicken is a great source of protein.

(2) a. Glass is made of liquid sand.
b. He drank the whole glass.

These are an instance of metonymy – a sense extension device that is based on contiguity (association, referential co-existence) of two concepts. The theoretical approaches mentioned above largely attribute regular polysemy to this figure, and so do the researchers in computational linguistics who adopt these theories (see Section 2 for their overview).

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 419–436
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

There is, however, another cognitive tool that structures polysemy – metaphor. Unlike metonymy, it is based on analogy, or referential disjunction (Lombard et al., 2023). Regular polysemy by metaphor can be exemplified by such polysemes as *antenna* (insect's organ, signal transmission device) or *leg* (limb, table support) instantiating the pattern BODY PART - OBJECT PART. The two figures are based on different cognitive mechanisms, have different processing profiles in our brain (Klepousniotou et al., 2012), but, as recent psycholinguistic studies show, they equally govern polysemous sense extensions (Lombard et al., 2023, 2024).

Another important aspect of regular polysemy is its continuous nature. In a recent study, Lombard et al. (2024) introduce a method to extract regular polysemes (including metaphors) from WordNet and suggest metrics to measure the degree of regularity of the patterns they are governed by (Table 1). Their findings are in contrast with the widely applied categorical approach to polysemy, where a sense extension of a polyseme is labeled in a binary way, i.e. as either regular or irregular.

Here we adopt this continuous view, aligning with recent work that argues that word meaning, polysemy regularity, and productivity form a continuum rather than discrete representations (Trott and Bergen, 2023; Li, 2024). To the best of our knowledge, no experimental design has previously targeted the graded aspect of regularity in LLMs, although researchers have noted that some patterns seemed more regular or productive than others (Li and Armstrong, 2024). We also contribute by focusing on metaphorically motivated regular polysemy. Only a handful of works in computational linguistics include regular metaphor in their experiments, and even less in the experiments with LMs in particular.

In order to investigate the effect of graded regularity on models' representation of metaphorically motivated polysemes, we rely on datasets compiled for psycholinguistic studies on human polysemy processing in French and English (Lombard et al., 2023, 2024). The datasets feature semantic neologisms – novel senses of existing words created using polysemy patterns of varying regularity degrees. These are compared against attested, existing polysemes and nonsensical derivations (refer to Table 2 for the examples). Human acceptability assessment confirmed the psychological validity of graded regularity for human processing: the more regular the polysemy pattern, the more acceptable its novel senses. Using surprisal and semantic similarity measures, we aim to find out how closely language model processing of semantic neologisms aligns with human processing, and whether the degree of regularity plays a role in it. With this in mind, we outline the following research questions:

RQ1. Which of the two measures (surprisal or semantic similarity) would be a better proxy for human behaviour in our task? As discussed in Methods section (§3), both proved to have psycholinguistic predictive power, despite operating at different levels of language structure.

RQ2. Are the results consistent across model types and sizes? Oh and Schuler (2022) show, e.g., that larger models do not necessarily deliver more human-like linguistic representations.

RQ3. Do models distinguish between the novel senses based on existing regular polysemy patterns and the senses created using the patterns that do not exist? To match human behaviour, models should be able to discriminate between these groups.

RQ4. Are LMs sensitive to the varying degrees of regularity of polysemy patterns? If their processing matches human ratings, we should expect the models to be less surprised by neologisms from highly regular patterns and vice versa.

RQ5. What type of regularity metrics (as defined in Table 1) are models more sensitive to: count-based or consistency-based? Do word frequency and word length play a role, and how does this compare with data from human evaluators?

In the case of LLMs, evaluating novel senses allows us to test their ability to generalize beyond previously seen material and avoid data contamination. Additionally, on a higher level, we can assess their sensitivity to the polysemy patterns abstracted from concrete, previously seen words.

Our results show that LLMs could discriminate between different sense types and regularity gradations in a human-like way, and overall correlated well with human sense plausibility judgment.

In the following sections we will briefly discuss the existing work on regular polysemy (§2), justify our methodology (§3), present the experiments (§4) and discuss their results (§5).

2 Related Work

Aside from the theoretical frameworks cited in the Introduction (§1), regular polysemy is studied in several areas dealing with language processing.

Psycholinguistics. In psycholinguistics, regular polysemy is addressed in the discussion about the meaning representation in human mind and the nature of restrictions that govern polysemy patterns in language. Many authors defend hybrid approaches to these problems. Rabagliati and Snedeker (2013) suggest that irregular senses are stored separately, while senses that follow regular patterns form core meanings. Analyzing co-predication acceptability and sense similarity of polysemes and homonyms, Haber and Poesio (2020) suggest that senses form groups according to their similarity (in line with Ortega-Andrés and Vicente, 2019), and reject the idea of a fully underspecified representation. In contrast, Vicente (2024) analyses regular and irregular polysemy along several dimensions and defends the one-representation hypothesis.

In the discussion on whether linguistic conventions or an underlying conceptual structure restrict polysemy patterns, Srinivasan and Rabagliati (2015) propose the "conventions-constrained-by-concepts" model. Their study across 15 languages suggests that while the conceptual structure governs the patterns, the language-specific conventions define senses that instantiate them. A hybrid approach is also supported by the investigations in language learning: Zhu (2021) studies how preschoolers acquire regular metonymies, highlighting their ability to quickly grasp semantic generalizations without extensive prior exposure. Children rely on an early-emerging conceptual structure, although at later stages linguistic generalizations also play a crucial role in word learning.

Mental processing of ambiguous words is affected by the degree of relatedness of meanings in memory. This is demonstrated by Brocher (2016; 2018), who report increased processing effort associated with disambiguation of unrelated meanings.

Computational Linguistics. In this field, regular polysemy is addressed in a variety of works, such as Boleda et al. (2012a,b); Lopukhina and Lopukhin (2016), who model systematic polysemy, or Del Tredici and Bel (2015), exploring the representations of polysemous and monosemous words in static word embeddings. A number of researchers propose methods of sense annotation for regular polysemy (Nimb and Pedersen, 2000; Freihat et al., 2013; Martinez Alonso, 2013), while other authors use WordNet to automatically extract regular polysemes (Peters and Peters, 2000; Barque and Chaumartin, 2009; Lombard et al., 2024). Interestingly, the latter authors recognize metaphoric extensions as types of systematic polysemy patterns, in contrast to most of the previously mentioned studies. Peters and Peters (2000) depart from an assumption that metaphoric alternations are irregular, but after applying their extraction method, "stumble upon" the instances of metaphoric sense extensions that can only be described as regular. Only a few more works mentioned in this section fully recognize that regular polysemy by metaphor is possible: Nimb and Pedersen, 2000; Freihat et al., 2013; Lopukhina and Lopukhin, 2016; Lombard et al., 2023 and Lombard et al., 2024.

Language models. Regarding regular polysemy and neural language models, Haber and Poesio (2021) test BERT's ability to predict human assessment of sense similarity degree. They report that BERT$_{\text{LARGE}}$ captures distinctions between polysemic, homonymic and same-sense samples in a human-like way. BERT delivers sensible results in sense clustering, suggesting that this model is sensitive to polysemy patterns. Sørensen et al. (2023) explore BERT sense clustering as a guidance tool for annotation of systematic polysemy in lexical resources. Similarly to Haber and Poesio (2021), they got mixed results but see potential: for one of the patterns, BERT discovered a sense that the authors overlooked when creating the dataset. Finally, Li and Armstrong (2024) use sense analogy questions to investigate how regular polysemy is represented in BERT embeddings. The authors observe that the pattern of BERT's sense similarity score distribution reflects differences not only in the processing of regular polysemes and irregular/homonymous controls, but also of distinct polysemy patterns. They also note on the scalar nature of regularity, an observation that contributes to Li's (2024) comprehensive approach to polysemy as continuous in its sense individuation, regularity, and productivity.

The present paper adopts the recent insights about the graded nature and metaphoric motivation of regular patterns and incorporates them in the experimental design.

3 Materials and Methods

3.1 Data

To answer our research questions, we evaluated two datasets compiled by Lombard et al. (2023) and Lombard et al. (2024)[1]. Both data sets were created for psycholinguistic experiments investigating the effect of graded regularity on the human

---

[1] Licensed under Creative Commons Attribution 4.0 International (CC-BY-4.0)

Metr.	Definition	Formula
R1	Number of words having SENSE$_1$ and SENSE$_2$ in a given pattern.	$R_1 = N_{S2}$
R2	Ratio of R1 and the number of words with SENSE$_1$, whether or not they have SENSE$_2$.	$R_2 = \frac{N_{S2}}{N_{S1}}$
R3	R1 weighted by the log-frequency of occurrence of the word.	$R_3 = \sum_{w=1}^{N_{S2}} \log(f_w)$
R4	R2 weighted by the log-frequency of occurrence of the word	$R_4 = \frac{\sum_{w=1}^{N_{S2}} \log(f_w)}{\sum_{w=1}^{N_{S1}} \log(f_w)}$

Table 1: Regularity metrics as proposed by (Lombard et al., 2024, pp. 4–5). While R1 and R3 capture the number of pattern instantiations, R2 and R4 reflect the consistency with which words having a base sense (SENSE$_1$) also have a derived sense (SENSE$_2$) within a pattern.

Type	Pattern	Example	W.	S.
new	ANIMAL - ARTIFACT	My sister cleaned the **porcupine** of the brush.	35	70
	ANIMAL - PERSON	The chessplayer is always a cruel **spider** with his opponents.		
	ARTIFACT - MESSAGE	A mean **spear** slipped through her lips in an angry tone.		
	BODY PART - OBJECT PART	We can see the **knee** of the chair getting damaged.		
	NATURAL EVENT - HAPPENING	There was a huge **tornado** of claps at the final of the challenge.		
	PERSON - ANIMAL	Some zoos are trying to protect the **doctor** from extinction.		
	PHYS. PROP. - PSYCHOL. PROP.	She said that the **density** of the project was an issue.		
illegal		My brother painted the **curry** of the controller in blue.	40	80
existing		My dog chewed the **tongue** of my new shoes	40	40
all			115	**190**

Table 2: Sentence examples of each sense type, labeled in the original dataset as *new*, *illegal*, and *existing*. *New* senses include 7 polysemy patterns (5 words per pattern). *Illegal* and *existing* senses are not annotated with patterns in the original dataset. The column *W.* lists the number of words per sense type, while *S.* – the number of sentences.

perception of neology. The more recent study is in English and focuses solely on regular metaphor, whereas the earlier one is in French and involves both metaphor and metonymy. Since the present research focuses on metaphoric polysemy, we only evaluate the part of the French dataset containing metaphors. Having removed the metonymies, we were left with only 42 sentences to evaluate, which limited our ability to derive meaningful results for French (see §4.2 and §5). The English dataset comprises 190 sentences.

The stimuli. The datasets contain sentences with target words of three types:

1. Semantic neology: words used in a novel, unattested sense. The derived metaphoric sense, together with the base sense, represent a polysemy pattern that a given word has never developed, unlike other words from its semantic field. To exemplify, the word *knee* represents a pattern BODY PART-OBJECT PART and is used in the sentence *We can see the knee of the chair getting damaged.* For comparison, some of the words that actually developed both senses are *leg, heart, artery, vein, antenna, wing, head, skeleton, brow, tongue* etc.

2. Non-sensical derivation: semantic neologisms that follow a non-existent pattern in each language. For instance, *curry* in *My brother painted the curry of the controller in blue* represents an unattested pattern FOOD-OBJECT PART.

3. Existing polysemy: words used in an attested sense of a valid, existing polysemy pattern. For example, *tongue* in *My dog chewed the tongue of my new shoes* is used in an attested sense of an OBJECT PART. An overview of the English dataset with sentence examples is presented in Table 2.

The dataset is annotated with human acceptability scores, regularity degree of polysemy patterns, word frequency and word length.

Human acceptability rating. Human acceptability scores are derived from the initial psycholinguistic experiment. They reflect how plausible the annotators found each sentence on a scale from 'no sense at all' (0) to 'completely acceptable' (100).

Regularity. Each target word is annotated with a score reflecting the degree of regularity of a polysemy pattern it instantiates. For the two languages, this metric has been calculated using different procedures. For English, the authors developed an automatic extraction technique using WordNet and proposed several formulas to calculate the regularity degree of a pattern based on the extracted data. These regularity metrics are summarized in Table 1. For French, the authors relied on the judgment of experts in French lexicology to assess the degree of regularity for each pattern. The methodological differences in the compilation of both datasets seem to affect our results, which will be discussed in more detail in Section 4.2.

3.2 Methods

To answer our research questions, we explore two common methods in NLP and computational psycholinguistics – surprisal and semantic similarity from large language models.

Surprisal. Surprisal is the negative log-probability of a token given its immediate context. Surprisal theory (Hale, 2001; Levy, 2008) assumes that the processing difficulty of the word is based on its predictability. This information-theoretic measure is typically used in the studies on human reading, where it proved to predict reading times and, consequently, cognitive processing difficulty in multiple languages (for recent work, see de Varda and Marelli, 2022; Nair and Resnik, 2023; Wilcox et al., 2023; Xu et al., 2023). It is also used to assess the models' ability to predict linguistic acceptability (grammaticality) of sentences (Noh et al., 2024). In our study, we use surprisal from language models as a proxy of human acceptability judgment of novel word senses: we assume that higher surprisal values assigned to a target word by an LM correspond to lower acceptability scores obtained from human evaluation.

Semantic relatedness. Semantic similarity between a word and its context is used along with surprisal to predict reading times, assess processing difficulty and explain brain activity during language processing (Leal et al., 2021; Salicchi et al., 2021; Kun et al., 2023). Specifically, we apply the cosine similarity between the vector of the target word and the vector of the sentence obtained by mean-pooling. Additionally, since a few rogue dimensions often dominate similarity measures in transformer models (Timkey and van Schijndel, 2021), we compare the original and normalized vectors (z-scoring) to assess their impact. We also use Spearman's ρ as a similarity metric, another technique suggested by Timkey and van Schijndel (2021) and replicated by Lyu et al. (2023) and Salicchi et al. (2023).

In reading experiments, low similarity between a word and its context is associated with increased human reading difficulty. In our study, we expect to associate low similarity rating from LMs with low human acceptability of semantic neologisms.

As shown by Salicchi et al. (2023), both surprisal and semantic relatedness equally contribute to the prediction of reading difficulty, despite operating at different levels of language structure. While surprisal operates at the syntagmatic level and reflects how predictable the word is from its context, semantic relatedness reflects coherence of a word with its context modeling paradigmatic dimension. Both surprisal and semantic relatedness proved to predict brain activity during language comprehension and are associated with signals from distinct brain areas (Frank and Willems, 2017; Michaelov et al., 2023; Salicchi and Hsu, 2025).

3.3 Models

We used a set of masked language models and compared them with an autoregressive Llama.

For English, we use the monolingual BERT as well as RoBERTa. For French, we took the BERT-based FlauBERT, and the RoBERTa-based CamemBERTv2. We also evaluate multilingual models on both languages: mBERT and XLM-RoBERTa.

Surprisal experiments typically use unidirectional decoder models (e.g., GPT), as they rely only on left-context to emulate human reading, avoiding access to future words. In our case, the experimental settings of the initial psycholinguistic study entail the choice of a masked model: the evaluators were first presented with the context on both sides before seeing the full sentence. We still include an autoregressive LM to compare the results and challenge our assumption about masked language modeling being more suitable for our task. For this, we chose Llama 3.1 8B and Llama 3.2 3B, which we oppose to BERT as more recent and significantly larger multilingual models that include English and French. For all models, weights were taken off HuggingFace. Additional information on these models is presented in Table 5 of Appendix A.

4 Experiments and Results

4.1 Experiments

We feed the sentences into each of the language models and compute[2] the surprisal and semantic relatedness scores as described in Methods section (§3). For this, we use the `minicons` library provided by Misra (2022). For bi-directional models, we rely on the 'pseudo-log-likelihood' proposed by Kauf and Ivanova (2023), which takes into account multi-token and out-of-vocabulary words.[3]

We then compute the Spearman correlation between the human acceptability scores and each of the measures (target word surprisal and the similarity between the target word and its context).

4.2 Results

Models	Default	K & I	PF
BERT$_{\text{BASE}}$	**-0.65**	-0.63	-0.61
BERT$_{\text{LARGE}}$	**-0.68**	-0.65	-0.64
RoBERTa$_{\text{BASE}}$	-0.67	**-0.76**	-0.68
RoBERTa$_{\text{LARGE}}$	-0.72	$\boxed{\mathbf{-0.78}}$	-0.70
XLM-RoBERTa$_{\text{BASE}}$	-0.38	**-0.56**	-0.39
XLM-RoBERTa$_{\text{LARGE}}$	-0.44	**-0.63**	-0.43
mBERT$_{\text{BASE}}$	-0.26	**-0.47**	**-0.47**
Llama 3.1 8B	**-0.65**	-	-
Llama 3.2 3B	**-0.65**	-	-

Table 3: Results of the surprisal experiment in English. The column *Default* reports results obtained from the default implementation of `minicons` (Misra, 2022), the column *K & I* reports the results from the method by Kauf and Ivanova (2023), and *PF* – from the PsychFormers application (Michaelov et al., 2023). All results are statistically significant (p<.05). Bold formatting points to the strongest correlation achieved by each model.

Surprisal. Across models, we observe moderate to strong correlation with human judgment. As expected, models correlate negatively, showing that more acceptable senses elicit lower surprisal.

Among masked models, the strongest correlation was achieved by RoBERTa$_{\text{LARGE}}$ at -0.78, p<.001. It is followed by BERT$_{\text{LARGE}}$ showing moderate negative correlation of -0.68, p<.001. Multilingual models demonstrated poorer results with correlation coefficients of -0.63 for XLM-RoBERTa$_{\text{LARGE}}$ (p<.001), as well as -0.47 for mBERT (p<.001). As mentioned previously, the method of Kauf and Ivanova (2023) yielded the best results, except for the BERT models which performed slightly better using the standard metric. See Table 3 for a complete overview of the different models and metrics.

As for autoregressive models, Llama 3.1 8B and Llama 3.2 3B achieved correlation of -0.65 (p<.001 for both), yielding the best results among the multilingual models but exhibiting a lower correlation than the smaller monolingual encoders.

In French, none of the models gave statistically significant correlation at the word level. We attribute this to the much smaller dataset size (42 sentences). However, we could still obtain usable results by changing the experimental settings: we checked correlation of sentence-wise surprisal with human judgment (obtained by sum and mean) and received statistically significant results for XLM-RoBERTa$_{\text{LARGE}}$, at -0.32, p=.039 (sum). We compared this result with the sentence surprisal of the English version from XLM-RoBERTa, and curiously, for English, this was the only model that showed stronger correlation when computing sentence surprisal instead of the target word surprisal (-0.65 vs. -0.63, p<.001 in both cases). Table 6 in Appendix C presents all scores obtained from the sentence-wise correlation experiment. Additionally, it reports correlation of sentence surprisal with the acceptability of polysemy patterns, where XLM-RoBERTa$_{\text{LARGE}}$ achieved moderate significant correlation.

Semantic relatedness. The results of the experiment with semantic relatedness are more difficult to summarize, as the data does not allow to discern clear trends. In different models, the highest correlation was achieved across varying layers, model sizes and normalization approaches. Moreover, some models show positive correlation with human judgment, while others correlate negatively. This is not expected, as usually we assume a better word/context coherence to elicit higher acceptability scores. Tables 7 to 10 of Appendix D offer a full overview of the correlation scores distribution within several selected models: masked RoBERTa and FlauBERT for English and French, as well as a significantly bigger multilingual autoregressive Llama 3.1 8B. Here, we will only highlight the best results achieved by the models to give an idea of how inconsistent they are across experimental settings.

The strongest correlation was reached by Llama

---

[2] The information on GPU use and computation time is reported in Appendix B.

[3] We also tested the standard scoring based on Salazar et al., 2020, and that of PsychoFormers (Michaelov and Bergen, 2022), obtaining generally lower results, as shown in Table 3.

3.1 8B (32 layers) in the layer 4 using Spearman's ρ instead of cosine, the correlation being positive (0.66, p<.001). RoBERTa$_{\text{BASE}}$ (12 layers) follows with coefficient of -0.58, p<.001 in the ninth layer without applying any normalization techniques. Finally, BERT$_{\text{BASE}}$ (12 layers) achieved the correlation of 0.52 in the last layer when applying Spearman's ρ instead of the cosine (p<.001). Multilingual models score 0.5 and below, their best achieved correlation coefficients being scattered across different experimental settings.

For French, the scores lie in the same range, but with the strongest correlation achieved by a smaller Llama 3.2 3B (-0.53, p<.001, in the last 28th layer, non-normalized).

Lyu et al. (2023) report similar outcome of their study of lexical stylistic features in language models: although normalization generally improves the results (especially for the multilingual models), it is hard to single out the best technique for all models and experimental settings. As for Salicchi et al. (2023), they do not notice any effect of BERT's embedding anisotropy on reading times prediction.

Overall, semantic relatedness results show no clear interpretable trend across models and settings.

5 Discussion

In this section, we will address the research questions presented in the Introduction (§1).

RQ1. Regarding the choice of measure (surprisal or semantic relatedness), the results suggest that surprisal is preferable. Not only because it achieved strong correlation (-0.78 for RoBERTa$_{\text{LARGE}}$ in surprisal vs. -0.61 for Llama 3.2 3B in similarity setting), but also because it is consistent, more interpretable, and easier to obtain. While we can confirm the assumption that surprisal is in an inverse relationship with sense plausibility as assessed by humans, the semantic similarity scores correlated both negatively and positively depending on the model, its layer and the embedding normalization technique (see Tables in Appendix D). Finding the most suitable configuration thus demands running a considerable number of trials.

RQ2. In surprisal setting, masked LMs performed better, confirming our assumption that masked model scoring with its access to the bi-directional context would be more suitable for our task. Previous research has repeatedly shown that larger model size delivers a poorer prediction of processing difficulty (Oh and Schuler, 2022; Salicchi et al., 2023; Liu et al., 2024; Shain et al., 2024). In contrast, in our experiments, large varieties of the same models always performed above the base ones (see Table 3). Interestingly, much larger Llama 3.1 8B and 3.2 3B did not outperform masked monolingual BERT and RoBERTa (330M and 355M respectively for large varieties). We attribute this to differences in model architecture, although it requires further investigation. The Limitations section (Appendix E) elaborates more on this issue. In the case of semantic similarity, results are not consistent enough to draw conclusions on this topic, as explained in the Results section.

We further analyse the results to test the models' sensitivity to such features as sense types, regularity degrees, word frequency and word length.

RQ3. We run a series of tests to confirm whether the models discriminate between the senses derived using the existing and non-existing patterns, as well as to see if they are sensitive to the varying pattern regularity degrees. We took our best-performing masked model RoBERTa$_{\text{LARGE}}$ and an autoregressive Llama 3.1 8B, for comparison. For French, we picked the same Llama model and FlauBERT$_{\text{LARGE}}$. A Mann-Whitney U test on two independent samples for the two sense types (two-sided, p<0.05) shows that the difference is significant for RoBERTa, Llama and human evaluators. They could distinguish between all three groups of senses (Figure 1). For French, FlauBERT and Llama 3.1 8B did not yield significancy (see Figure 4 in Appendix E for score distribution).

RQ4. In the same way, we established that language models were sensitive to the degrees of regularity of the polysemy patterns the senses instantiated, although not as fine-grained as humans: while the Mann-Whitney U test shows significancy in the difference between low, medium and high regularity of patterns for humans, the models only discriminate between high/low and medium/low groups (Figure 2). Again, neither Llama nor FlauBERT reached statistical significance in French. The distribution plot can be found in Appendix E, Figure 5.

RQ5. We also establish whether there is a relation between the model scoring and such factors as the degree of polysemy pattern regularity, word frequency and word length. The latter two factors contribute to the cognitive processing load in humans since less frequent and longer words require more time to process (Pollatsek et al., 2008). Figure 3 illustrates the pattern of correlation (Pearson) between four regularity metrics and the measures

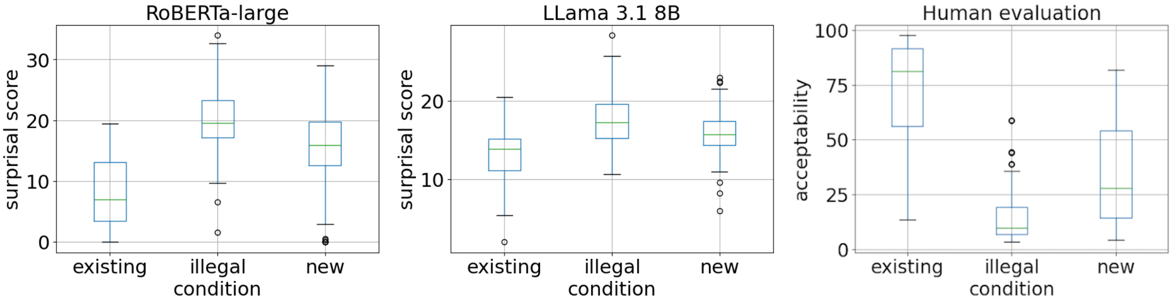


Figure 1: Distribution of English surprisal scores by condition labeled in the original dataset as *new*, *illegal* and *existing*. These correspond to the groups (1), (2) and (3) respectively, as described in the Section 3.1. High model surprisal is expected to correspond to the low acceptability scores in the human rating.

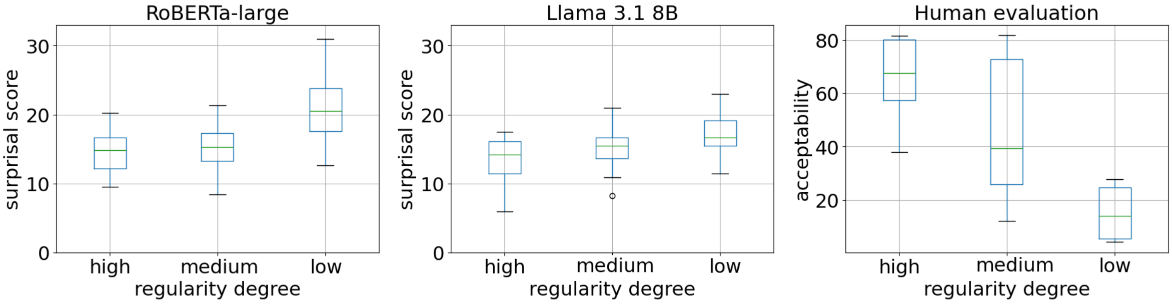


Figure 2: Distribution of surprisal scores by regularity degrees labeled as *high*, *medium* and *low* for the English data. High model surprisal is expected to correspond to the low acceptability scores in the human rating.

of semantic similarity and surprisal from RoBERTa and Llama, as compared to human evaluators. As described in Table 1, we consider a count-based metric R1, consistency-based metric R2 and two metrics that weight them by the log-frequency of the occurrence of the word – R3 and R4. The measure of surprisal is more aligned with the human judgment correlated with regularity, Llama showing almost identical coefficients for most metrics. The same as for human evaluators, for both language models, the regularity metrics that reflect how consistently words instantiate a polysemy pattern appeared to be more relevant than the sheer number of words having the SENSE$_1$ and the SENSE$_2$. Weighting R1 and R2 by word frequency generally did not improve the correlation coefficients (except for RoBERTa in R4 where it gains one point). Again, the correlation scores for the measure of semantic relatedness are generally low, with apparent preference for consistency-based and frequency-weighted metrics. All correlation coefficients are listed in Table 11, Appendix E.

As for the effect of the word frequency and word length, the models generally show a low correlation, although it is higher than the one computed with human scoring. The exceptions are RoBERTa in the similarity setting and FlauBERT in the surprisal setting relying on these features more and correlating moderately (see Table 4 for correlation scores and Figure 6 in Appendix E for visualization).

English		
Models	**W. freq.**	**W. length**
RoBERTa$_{\text{LARGE}}$ SURP	-0.21*	0.24*
EN Llama 3.1 8B SURP	-0.27*	0.16*
RoBERTa$_{\text{LARGE}}$ SIMIL	-0.34*	0.33*
Llama 3.1. 8B SIMIL	0.07	0.1
Human acceptability rating	0.04*	-0.09*
French		
FlauBERT$_{\text{LARGE}}$ SURP	0.38*	0.41*
Llama 3.1 8B SURP	0.00	0.29
FlauBERT$_{\text{LARGE}}$ SIMIL	0.06	-0.07
Llama 3.1 8B SIMIL	-0.16	0.16
Human acceptability rating	-0.15	0.17

Table 4: Correlation (Pearson) of word length and word frequency with model scoring and human evaluation. Asterisk (*) indicates a statistically significant correlation (p<.05).

6 Conclusions

In this paper, we investigated the effect of the graded regularity of polysemy patterns on the pro-

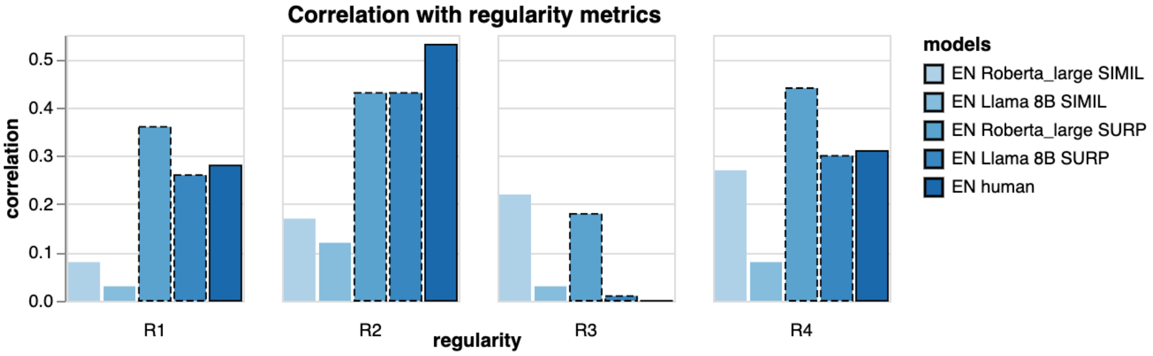


Figure 3: Pearson's r for regularity metrics as described in Section 5, in absolute numbers. The dashed line marks the measure of surprisal from RoBERTa and Llama, while word/context similarity remains unmarked.

cessing of novel metaphorical word senses by large language models. Using surprisal and semantic relatedness as proxies, we found evidence that models represent regularity of polysemy extensions in a human-like way. Especially surprisal proved to adequately model sense plausibility, showing a strong correlation with human judgment. Among models, RoBERTa delivered the best results. Furthermore, the distributions of model scores suggest sensitivity to different types of sense extensions and regularity degrees. Similarly to humans, LLMs could discriminate between attested polysemes, novel senses derived from regular polysemy patterns and nonsensical derivations. They were, however, less responsive to the gradations in regularity, only differentiating very regular and weakly regular patterns. These observations allow us to better understand how LLMs model lexical ambiguity and to what extent such factors as regularity, continuity and sense relatedness affect model representations.

Acknowledgments

This work was funded by MCIU/AEI/10.13039/501100011033 (grants with references PID 2021-128811OA-I00, PRE2022-102762, CNS2024-154902, and TED 2021-130295B-C33, the latter also funded by "European Union Next Generation EU/PRTR"), by the Galician Government (ERDF 2024-2027: Call ED431G 2023/04, and ED431F 2021/01), and by a Ramón y Cajal grant (RYC2019-028473-I).

References

Wissam Antoun, Francis Kulumba, Rian Touchent, Éric de la Clergerie, Benoît Sagot, and Djamé Seddah. 2024. Camembert 2.0: A smarter french language model aged to perfection. *Preprint*, arXiv:2411.08868.

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

Lucie Barque and François-Régis Chaumartin. 2009. Regular polysemy in WordNet. *Journal for language technology and computational linguistics*, 24(2):5–18.

Gemma Boleda, Sebastian Padó, and Jason Utt. 2012a. Regular polysemy: A distributional model. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 151–160, Montréal, Canada. Association for Computational Linguistics.

Gemma Boleda, Sabine Schulte im Walde, and Toni Badia. 2012b. Modeling regular polysemy: A study on the semantic classification of Catalan adjectives. *Computational Linguistics*, 38(3):575–616.

Andreas Brocher, Stephani Foraker, and Jean-Pierre Koenig. 2016. Processing of irregular polysemes in sentence reading. *Journal of experimental psychology. Learning, memory, and cognition*, 42 11:1798–1813.

Andreas Brocher, Jean-Pierre Koenig, Gail Mauner, and Stephani Foraker. 2018. About sharing and commitment: the retrieval of biased and balanced irregular polysemes. *Language, Cognition and Neuroscience*, 33:443 – 466.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Andrea de Varda and Marco Marelli. 2022. The effects of surprisal across languages: Results from native and non-native reading. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP*

2022, pages 138–144, Online only. Association for Computational Linguistics.

Marco Del Tredici and Núria Bel. 2015. A word-embedding-based sense index for regular polysemy representation. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 70–78, Denver, Colorado. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Kevin Durkin and Jocelyn Manning. 1989. Polysemy and the subjective lexicon: Semantic relatedness and the salience of intraword senses. *Journal of Psycholinguistic Research*, 18:577–612.

S. Frank and Roel M. Willems. 2017. Word predictability and semantic similarity show distinct patterns of brain activity during language comprehension. *Language, Cognition and Neuroscience*, 32:1192 – 1203.

Abed Alhakim Freihat, Fausto Giunchiglia, and Biswanath Dutta. 2013. Regular polysemy in wordnet and pattern based approach. *International Journal On Advances in Intelligent Systems*, 6:199–212.

Aina Garí Soler and Marianna Apidianaki. 2021. Let‘s play mono-poly: BERT can reveal words' polysemy level and partitionability into senses. *Transactions of the Association for Computational Linguistics*, 9:825–844.

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.

Janosch Haber and Massimo Poesio. 2020. Assessing polyseme sense similarity through co-predication acceptability and contextualised embedding distance. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 114–124, Barcelona, Spain (Online). Association for Computational Linguistics.

Janosch Haber and Massimo Poesio. 2021. Patterns of polysemy and homonymy in contextualised language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2663–2676, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Janosch Haber and Massimo Poesio. 2024. Polysemy—Evidence from linguistics, behavioral science, and contextualized language models. *Computational Linguistics*, 50(1):351–417.

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*.

Carina Kauf and Anna Ivanova. 2023. A better way to do masked language model scoring. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 925–935, Toronto, Canada. Association for Computational Linguistics.

Ekaterini Klepousniotou, G. Bruce Pike, Karsten Steinhauer, and Vincent L. Gracco. 2012. Not all ambiguous words are created equal: An eeg investigation of homonymy and polysemy. *Brain and Language*, 123:11–21.

Sun Kun, Qiuying Wang, and Xiaofei Lu. 2023. An interpretable measure of semantic similarity for predicting eye movements in reading. *Psychonomic Bulletin & Review*, 30:1227 – 1242.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Sidney Leal, Edresson Casanova, Gustavo Paetzold, and Sandra Aluísio. 2021. Evaluating semantic similarity methods to build semantic predictability norms of reading data. In *Text, Speech, and Dialogue*, pages 35–47, Cham. Springer International Publishing.

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

Jiangtian Li. 2024. Semantic minimalism and the continuous nature of polysemy. *Mind & Language*, 39(5):680–705.

Jiangtian Li and Blair C Armstrong. 2024. Probing the representational structure of regular polysemy via sense analogy questions: Insights from contextual word vectors. *Cognitive Science*, 48(3):e13416.

Jiangtian Li and Marc Joanisse. 2021. Word senses as clusters of meaning modulations: A computational model of polysemy. *Cognitive Science*, 45.

Tong Liu, Iza Škrjanec, and Vera Demberg. 2024. Temperature-scaling surprisal estimates improve fit to human reading times – but does it do so for the "right reasons"? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9598–9619, Bangkok, Thailand. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Alizée Lombard, Richard Huyghe, Lucie Barque, and Doriane Gras. 2023. Regular polysemy and novel word-sense identification. *The Mental Lexicon*, 18(1):94–119.

Alizée Lombard, Anastasia Ulicheva, Maria Korochkina, and Kathy Rastle. 2024. The regularity of polysemy patterns in the mind: Computational and experimental data. *Glossa Psycholinguistics*, 3(1).

Anastasiya Lopukhina and Konstantin Lopukhin. 2016. Regular polysemy: from sense vectors to sense patterns. In *Proceedings of the 5th Workshop on Cognitive Aspects of the Lexicon (CogALex-V)*, pages 19–23.

Qing Lyu, Marianna Apidianaki, and Chris Callison-burch. 2023. Representation of lexical stylistic features in language models' embedding space. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 370–387, Toronto, Canada. Association for Computational Linguistics.

Hector Martinez Alonso. 2013. *Annotation of Regular Polysemy: An empirical assessment of the underspecified sense*. Ph.D. thesis, Universitat Pompeu Fabra and University of Copenhagen, Barcelona, Spain.

James A. Michaelov, Megan D. Bardolph, Cyma K. Van Petten, Benjamin K. Bergen, and Seana Coulson. 2023. Strong prediction: Language model surprisal explains multiple n400 effects. *Neurobiology of Language*, 5:107 – 135.

James A. Michaelov and Benjamin K. Bergen. 2022. Do language models make human-like predictions about the coreferents of Italian anaphoric zero pronouns? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1–14, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Sathvik Nair and Philip Resnik. 2023. Words, subwords, and morphemes: What really matters in the surprisal-reading time relationship? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11251–11260, Singapore. Association for Computational Linguistics.

Sathvik Nair, Mahesh Srinivasan, and Stephan Meylan. 2020. Contextualized word embeddings encode aspects of human-like word sense knowledge. In *Proceedings of the Workshop on the Cognitive Aspects of the Lexicon*, pages 129–141, Online. Association for Computational Linguistics.

Sanni Nimb and Bolette Sanford Pedersen. 2000. Treating metaphoric senses in a danish computational lexicon –different cases of regular polysemy. In *Proceedings of the 9th EURALEX International Congress*, pages 679–691, Stuttgart, Germany. Institut für Maschinelle Sprachverarbeitung.

Kangsan Noh, Eunjeong Oh, and Sanghoun Song. 2024. Testing language models' syntactic sensitivity to grammatical constraints: a case study of wanna contraction. *Frontiers in Communication*, 9.

Byung-Doh Oh and William Schuler. 2022. 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.

Marina Ortega-Andrés and Agustín Vicente. 2019. Polysemy and co-predication. *Glossa: a journal of general linguistics*, 4(1).

Wim Peters and Ivonne Peters. 2000. Lexicalised systematic polysemy in WordNet. In *Proceedings of the*

Second International Conference on Language Resources and Evaluation (LREC'00), Athens, Greece. European Language Resources Association (ELRA).

Alexander Pollatsek, Barbara Jean Juhasz, Erik D. Reichle, Debra Machacek, and Keith Rayner. 2008. Immediate and delayed effects of word frequency and word length on eye movements in reading: a reversed delayed effect of word length. *Journal of experimental psychology. Human perception and performance*, 34 3:726–50.

James Pustejovsky. 1991. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441.

Hugh Rabagliati and Jesse Snedeker. 2013. The truth about chickens and bats: Ambiguity avoidance distinguishes types of polysemy. *Psychological science*, 24(7):1354–1360.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Lavinia Salicchi, Emmanuele Chersoni, and Alessandro Lenci. 2023. A study on surprisal and semantic relatedness for eye-tracking data prediction. *Frontiers in Psychology*, 14.

Lavinia Salicchi and Yu-Yin Hsu. 2025. Not every metric is equal: Cognitive models for predicting n400 and p600 components during reading comprehension. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3648–3654, Abu Dhabi, UAE. Association for Computational Linguistics.

Lavinia Salicchi, Alessandro Lenci, and Emmanuele Chersoni. 2021. Looking for a role for word embeddings in eye-tracking features prediction: Does semantic similarity help? In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 87–92, Groningen, The Netherlands (online). Association for Computational Linguistics.

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

Nathalie Sørensen, Sanni Nimb, and Bolette Sandford Pedersen. 2023. How do we treat systematic polysemy in wordnets and similar resources?–using human intuition and contextualized embeddings as guidance. In *Proceedings of the 12th Global Wordnet Conference*, pages 117–126.

Mahesh Srinivasan and Hugh Rabagliati. 2015. How concepts and conventions structure the lexicon: Cross-linguistic evidence from polysemy. *Lingua*, 157:124–152.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sean Trott and Benjamin Bergen. 2023. Word meaning is both categorical and continuous. *Psychological Review*, 130(5):1239.

Agustín Vicente. 2024. Polysemies and the one representation hypothesis. *The Mental Lexicon*.

Gregor Wiedemann, Steffen Remus, Avi Chawla, and Chris Biemann. 2019. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 161–170, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

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.

Weijie Xu, Jason Chon, Tianran Liu, and Richard Futrell. 2023. The linearity of the effect of surprisal on reading times across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15711–15721, Singapore. Association for Computational Linguistics.

Rebecca Zhu. 2021. Preschoolers' acquisition of producer-product metonymy. *Cognitive Development*, 59:101075.

George Kingsley Zipf. 1945. The meaning-frequency relationship of words. *The Journal of general psychology*, 33:251–6.

A Model Information

Table 5 offers a model description in terms of size and languages, HuggingFace names, as well as paper references.

Model	**Reference**	**Layers**	**Lang.**	**Size**
`google-bert/bert-base-uncased`	Devlin et al., 2019	12	EN	110M
`google-bert/bert-large-uncased`	Devlin et al., 2019	24	EN	330M
`FacebookAI/roberta-base`	Liu et al., 2019	12	EN	125M
`FacebookAI/roberta-large`	Liu et al., 2019	24	EN	355M
`almanach/camembertv2-base`	Antoun et al., 2024	12	FR	112M
`FlauBERT/flaubert_base_cased`	Le et al., 2020	12	FR	138M
`FlauBERT/flaubert_large_cased`	Le et al., 2020	24	FR	373M
`google-bert/bert-base-multilingual-uncased`	Devlin et al., 2019	12	multi	179M
`FacebookAI/xlm-roberta-base`	Conneau et al., 2019	12	multi	279M
`FacebookAI/xlm-roberta-large`	Conneau et al., 2019	24	multi	560M
`meta-llama/Llama-3.1-8B`	Grattafiori et al., 2024	32	multi	8B
`meta-llama/Llama-3.2-3B`	Grattafiori et al., 2024	28	multi	3B

Table 5: Models used in this experiment can be obtained from `https://huggingface.co`.

B GPU Use

We ran the experiments with Llama 3.1 8B and LLama 3.2 3B on an Nvidia Ampere A100 80GB GPU node. We estimate the overall execution time to be 200 hours.

C Sentence-wise Surprisal

Sentence surprisal for EN models, word acceptability			**Sentence surprisal for FR models, word acceptability**			**Sent. surprisal for FR models, pattern acceptability**	
Model	**Default**	**K & I**	**Model**	**Default**	**K & I**	**Default**	**K & I**
BERT$_{\text{BASE}}$	-0.46*	-0.54*	CamemBERTv2$_{\text{BASE}}$	0.02	-0.06	-0.2	-0.11
BERT$_{\text{LARGE}}$	-0.46*	-0.55*	FlauBERT$_{\text{BASE}}$	-0.18	-	-0.11	-
RoBERTa$_{\text{BASE}}$	-0.57*	**-0.63***	FlauBERT$_{\text{LARGE}}$	-0.17	-	-0.06	-
RoBERTa$_{\text{LARGE}}$	-0.54*	-0.61*	XLM-RoBERTa$_{\text{BASE}}$	-0.24	-0.15	-0.3	0.01
XLM-RoBERTa$_{\text{BASE}}$	-0.36*	-0.54*	XLM-RoBERTa$_{\text{LARGE}}$	**-0.32***	-0.23	**-0.52***	-0.21
XLM-RoBERTa$_{\text{LARGE}}$	-0.42*	-0.65*	mBERT$_{\text{BASE}}$	0.02	0.02	-0.12	-0.02
mBERT$_{\text{BASE}}$	-0.16	-0.41*	Llama 3.2 3B	-0.2	-	-0.23	-
Llama 3.2 3B	-0.62*	-	Llama 3.1 8B	-0.19	-	-0.27	-
Llama 3.1 8B	-0.62*	-					

Table 6: Spearman correlaiton coefficients for sentence surprisal and human acceptability scores in English and French. For French, we additionally include the correlation with human acceptability as per polysemy pattern (averaged over acceptability of each word instance of a pattern). Asterisk (*) indicates a statistically significant result at $p < .05$.

D Semantic Similarity vs. Human Acceptability Judgment

We show mixed results obtained from the experiments with semantic similarity for selected models in Tables 7 to 10. Results are presented for RoBERTa, FlauBERT and Llama 3.1 8B covering both languages.

	RoBERTa$_{\text{BASE}}$		
L.	**Cosine**	**Cosine norm.**	**Spearman**
0	-0.2, p=.034	0.08, p=.422	0.0, p=.964
1	-0.18, p=.058	0.27, p=.003	0.2, p=.031
2	-0.27, p=.003	0.37, p<.001	0.25, p=.006
3	-0.28, p=.002	0.44, p<.001	0.34, p<.001
4	-0.31, p<.001	**0.56, p<.001**	0.38, p<.001
5	-0.32, p<.001	0.54, p<.001	**0.42, p<.001**
6	-0.39, p<.001	0.42, p<.001	0.32, p<.001
7	-0.52, p<.001	0.25, p=.008	0.15, p=.105
8	-0.55, p<.001	0.1, p=.301	0.06, p=.494
9	**-0.58, p<.001**	-0.14, p=.129	-0.14, p=.144
10	-0.38, p<.001	-0.26, p=.006	-0.27, p=.003
11	-0.15, p=.102	-0.33, p<.001	-0.26, p=.005
12	-0.19, p=.04	-0.3, p=.001	-0.23, p=.013
	RoBERTa$_{\text{LARGE}}$		
0	-0.02, p=.808	-0.0, p=.959	-0.05, p=.6
1	-0.15, p=.121	0.15, p=.104	0.0, p=.966
2	-0.32, p<.001	0.27, p=.004	-0.07, p=.485
3	-0.36, p<.001	0.29, p=.001	0.0, p=.974
4	-0.29, p=.002	0.32, p<.001	0.22, p=.017
5	-0.14, p=.131	0.36, p<.001	0.17, p=.062
6	-0.16, p=.086	0.43, p<.001	0.2, p=.034
7	-0.21, p=.022	0.48, p<.001	0.33, p<.001
8	-0.2, p=.029	**0.5, p<.001**	0.31, p<.001
9	-0.26, p=.005	0.47, p<.001	0.29, p=.002
10	-0.17, p=.078	0.41, p<.001	0.16, p=.083
11	0.01, p=.923	0.36, p<.001	0.09, p=.325
12	-0.25, p=.007	0.42, p<.001	0.14, p=.123
13	-0.17, p=.078	0.32, p<.001	0.06, p=.508
14	-0.21, p=.022	0.18, p=.048	-0.0, p=.99
15	-0.23, p=.016	0.11, p=.238	-0.09, p=.316
16	-0.23, p=.014	0.13, p=.166	-0.06, p=.549
17	-0.21, p=.023	0.11, p=.254	-0.08, p=.376
18	-0.2, p=.029	-0.05, p=.589	-0.2, p=.033
19	-0.37, p<.001	-0.18, p=.059	-0.26, p=.006
20	**-0.48, p<.001**	-0.29, p=.002	-0.18, p=.057
21	-0.48, p<.001	-0.36, p<.001	-0.12, p=.188
22	-0.34, p<.001	-0.46, p<.001	-0.2, p=.028
23	-0.0, p=.974	-0.49, p<.001	**-0.34, p<.001**
24	-0.4, p<.001	-0.26, p=.005	-0.21, p=.025

Table 7: Layerwise correlation of human acceptability scores with semantic relatedness of a word with its context, as computed by RoBERTa$_{\text{BASE}}$ and RoBERTa$_{\text{LARGE}}$ models.

	FlauBERT$_{\text{BASE}}$		
L.	**Cosine**	**Cosine norm.**	**Spearman**
0	-0.01, p=.937	0.15, p=.356	-0.05, p=.764
1	0.05, p=.763	0.19, p=.22	0.08, p=.623
2	0.03, p=.848	0.19, p=.221	0.05, p=.773
3	0.02, p=.901	0.22, p=.17	0.04, p=.796
4	0.02, p=.882	0.24, p=.126	0.07, p=.677
5	0.04, p=.825	0.24, p=.125	0.07, p=.673
6	0.01, p=.935	0.21, p=.187	0.05, p=.737
7	0.05, p=.736	0.2, p=.205	0.04, p=.793
8	0.06, p=.696	0.21, p=.181	0.1, p=.51
9	0.11, p=.491	0.15, p=.331	0.09, p=.562
10	0.09, p=.55	0.15, p=.347	0.09, p=.578
11	0.13, p=.404	0.21, p=.182	0.13, p=.401
12	**0.31, p=.049**	**0.35, p=.023**	**0.33, p=.031**
	FlauBERT$_{\text{LARGE}}$		
0	-0.05, p=.765	0.17, p=.293	-0.06, p=.719
1	-0.08, p=.628	0.18, p=.241	0.04, p=.81
2	-0.11, p=.471	0.14, p=.375	-0.07, p=.647
3	-0.15, p=.352	0.09, p=.553	-0.05, p=.76
4	-0.32, p=.041	0.04, p=.79	-0.15, p=.359
5	-0.34, p=.026	0.1, p=.513	-0.12, p=.464
6	-0.35, p=.022	0.2, p=.207	-0.08, p=.634
7	-0.31, p=.045	0.4, p=.01	-0.01, p=.944
8	-0.32, p=.036	0.45, p=.003	0.09, p=.577
9	-0.41, p=.008	0.45, p=.002	0.06, p=.684
10	-0.39, p=.011	0.45, p=.003	0.12, p=.436
11	-0.4, p=.009	0.45, p=.003	0.16, p=.316
12	-0.38, p=.014	0.46, p=.002	0.15, p=.329
13	-0.38, p=.014	**0.5, p<.001**	0.14, p=.369
14	**-0.42, p=.006**	0.42, p=.005	0.12, p=.43
15	-0.39, p=.01	0.37, p=.016	0.13, p=.42
16	-0.41, p=.007	0.38, p=.014	0.16, p=.324
17	-0.35, p=.023	0.42, p=.006	0.19, p=.232
18	-0.31, p=.044	0.43, p=.005	0.14, p=.38
19	-0.35, p=.023	0.41, p=.008	0.15, p=.332
20	-0.32, p=.036	0.39, p=.011	0.13, p=.414
21	-0.24, p=.121	0.38, p=.013	0.12, p=.438
22	-0.23, p=.136	0.39, p=.01	0.13, p=.395
23	-0.21, p=.18	0.4, p=.01	0.11, p=.489
24	-0.25, p=.106	0.44, p=.003	0.15, p=.336

Table 8: Layerwise correlation of human acceptability scores with semantic relatedness of a word with its context, as computed by FlauBERT$_{\text{BASE}}$ and FlauBERT$_{\text{LARGE}}$.

Llama 3.1 8B (on English)			
L.	**Cosine**	**Cosine norm.**	**Spearman**
0	-0.01, p=.886	-0.16, p=.095	0.07, p=.464
1	0.02, p=.814	-0.01, p=.938	0.13, p=.164
2	-0.03, p=.749	0.19, p=.039	0.39, p<.001
3	-0.13, p=.157	0.34, p<.001	0.49, p<.001
4	0.06, p=.522	**0.54, p<.001**	**0.66, p<.001**
5	0.07, p=.449	0.5, p<.001	0.61, p<.001
6	0.13, p=.164	0.41, p<.001	0.51, p<.001
7	0.21, p=.026	0.43, p<.001	0.54, p<.001
8	0.28, p=.002	0.43, p<.001	0.54, p<.001
9	0.42, p<.001	0.4, p<.001	0.5, p<.001
10	0.42, p<.001	0.41, p<.001	0.52, p<.001
11	0.48, p<.001	0.4, p<.001	0.48, p<.001
12	0.51, p<.001	0.44, p<.001	0.51, p<.001
13	0.58, p<.001	0.48, p<.001	0.54, p<.001
14	**0.58, p<.001**	0.49, p<.001	0.54, p<.001
15	0.54, p<.001	0.48, p<.001	0.5, p<.001
16	0.52, p<.001	0.43, p<.001	0.47, p<.001
17	0.47, p<.001	0.42, p<.001	0.49, p<.001
18	0.38, p<.001	0.38, p<.001	0.45, p<.001
19	0.36, p<.001	0.38, p<.001	0.46, p<.001
20	0.32, p<.001	0.36, p<.001	0.47, p<.001
21	0.31, p<.001	0.35, p<.001	0.47, p<.001
22	0.29, p=.002	0.26, p=.005	0.41, p<.001
23	0.25, p=.007	0.19, p=.041	0.35, p<.001
24	0.22, p=.016	0.16, p=.08	0.33, p<.001
25	0.2, p=.03	0.13, p=.152	0.3, p=.001
26	0.19, p=.042	0.11, p=.222	0.29, p=.001
27	0.21, p=.022	0.1, p=.28	0.27, p=.003
28	0.15, p=.109	0.09, p=.328	0.29, p=.002
29	0.12, p=.194	0.07, p=.486	0.26, p=.005
30	0.11, p=.228	0.05, p=.563	0.25, p=.007
31	0.04, p=.648	0.06, p=.54	0.21, p=.023
32	-0.26, p=.004	0.07, p=.428	-0.2, p=.032

Table 9: Layerwise correlation of human acceptability scores with semantic relatedness of a word with its context, as computed by Llama 3.1 8B, on English.

Llama 3.1 8B (on French)			
L.	**Cosine**	**Cosine norm.**	**Spearman**
0	-0.04, p=.812	0.12, p=.478	-0.02, p=.885
1	0.04, p=.816	0.05, p=.779	-0.01, p=.943
2	-0.13, p=.444	0.29, p=.076	0.03, p=.867
3	-0.23, p=.17	0.06, p=.74	-0.01, p=.947
4	-0.29, p=.073	0.11, p=.525	0.08, p=.651
5	-0.21, p=.214	0.31, p=.059	0.16, p=.348
6	-0.06, p=.741	0.4, p=.013	0.19, p=.263
7	0.09, p=.596	**0.41, p=.01**	0.2, p=.228
8	-0.06, p=.703	0.37, p=.022	0.2, p=.239
9	-0.03, p=.859	0.3, p=.07	0.19, p=.263
10	0.06, p=.723	0.27, p=.099	0.18, p=.27
11	0.17, p=.31	0.3, p=.066	0.2, p=.234
12	0.12, p=.455	0.26, p=.121	0.14, p=.388
13	0.06, p=.725	0.25, p=.128	0.2, p=.238
14	0.12, p=.492	0.24, p=.141	0.2, p=.226
15	0.09, p=.608	0.15, p=.385	0.12, p=.472
16	0.04, p=.805	0.09, p=.609	0.01, p=.94
17	-0.01, p=.968	0.12, p=.478	0.11, p=.518
18	-0.09, p=.591	0.1, p=.569	0.14, p=.411
19	-0.14, p=.392	0.09, p=.605	0.13, p=.431
20	-0.22, p=.185	0.1, p=.555	0.1, p=.55
21	-0.23, p=.156	0.09, p=.581	0.04, p=.811
22	-0.24, p=.14	0.1, p=.541	0.02, p=.895
23	-0.25, p=.133	0.07, p=.655	-0.0, p=.996
24	-0.27, p=.107	0.15, p=.364	0.0, p=.98
25	-0.27, p=.097	0.1, p=.537	0.04, p=.806
26	-0.3, p=.065	0.17, p=.316	0.07, p=.695
27	-0.24, p=.139	0.16, p=.345	0.09, p=.611
28	-0.26, p=.117	0.16, p=.34	0.1, p=.559
29	-0.26, p=.113	0.17, p=.316	0.11, p=.524
30	-0.33, p=.04	0.14, p=.393	0.02, p=.888
31	-0.29, p=.074	0.1, p=.549	-0.1, p=.541
32	**-0.33, p=.046**	0.09, p=.609	-0.21, p=.21

Table 10: Layerwise correlation of human acceptability scores with semantic relatedness of a word with its context, as computed by Llama 3.1 8B, on French.

E Analysis of Model Scoring

Figure 4 demonstrates the surprisal score distribution in French for three types of senses as described in Section 3.1. The data on human evaluation is absent from this figure, since it was not obtained by the authors of the initial psycholinguistic experiment (Lombard et al., 2023).

Figure 5 shows surprisal score distribution in French as per regularity degree, along with human acceptability judgment.

Table 11 reveals the relationships of the models with the degree of pattern regularity, while Figure 4 shows correlation with word frequency and word length, as compared to the human evaluators.

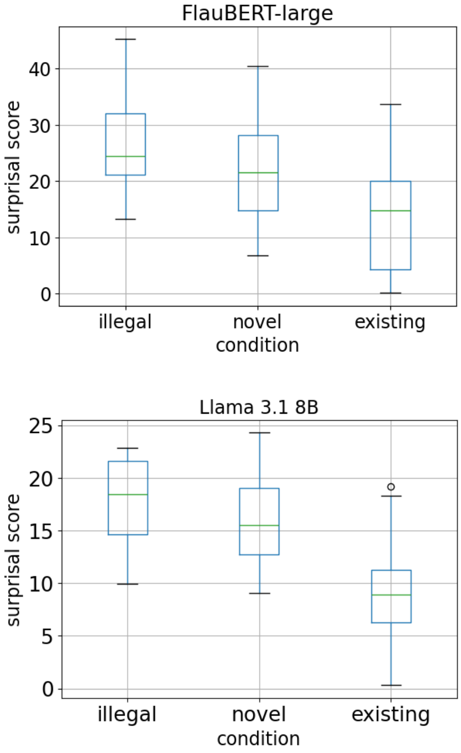


Figure 4: Distribution of French surprisal scores by condition labeled in the original dataset as *new*, *illegal* and *existing*. These correspond to the groups (1), (2) and (3) described in the Section 3.1

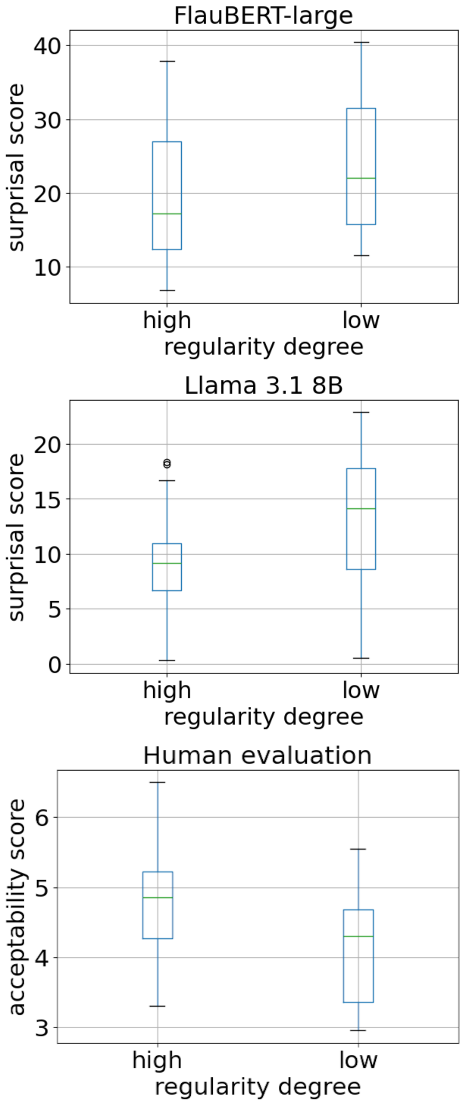


Figure 5: Distribution of surprisal scores by regularity degrees labeled as *high* and *low* for the French data.

Models	R1	R2	R3	R4
RoBERTa$_{\text{LARGE}}$ SURP	-0.36*	-0.43*	-0.18	**-0.44***
Llama 3.1 8B SURP	-0.26*	**-0.43***	-0.01	-0.3*
RoBERTa$_{\text{LARGE}}$ SIMIL	-0.08	-0.17	-0.22	**-0.27***
Llama 3.1 8B SIMIL	0.03	-0.12	-0.03	-0.08
Human acceptability rating	0.28*	**0.53***	0.00	0.31*

Table 11: Correlation (Pearson) between the regularity metrics and scores from surprisal, similarity and human evaluation. The regularity metrics are defined in Table 1. Asterisk (*) indicates a statistically significant correlation ($p < .05$). Since the high human acceptability is associated with low model surprisal, the negative correlation with model-derived metrics is expected.

Limitations

Languages. In our experiments, we mainly focus on English data. The size of the dataset in French often did not allow us to make meaningful comparisons or confirm the validity of results received from English. As shown by Srinivasan and Rabagliati (2015), polysemy patterns and their degree of regularity overlap only partially across languages. Hence, studying the phenomenon of continuous regular polysemy in one language has limited generalization potential. More languages from diverse language families need to be involved in such investigations.

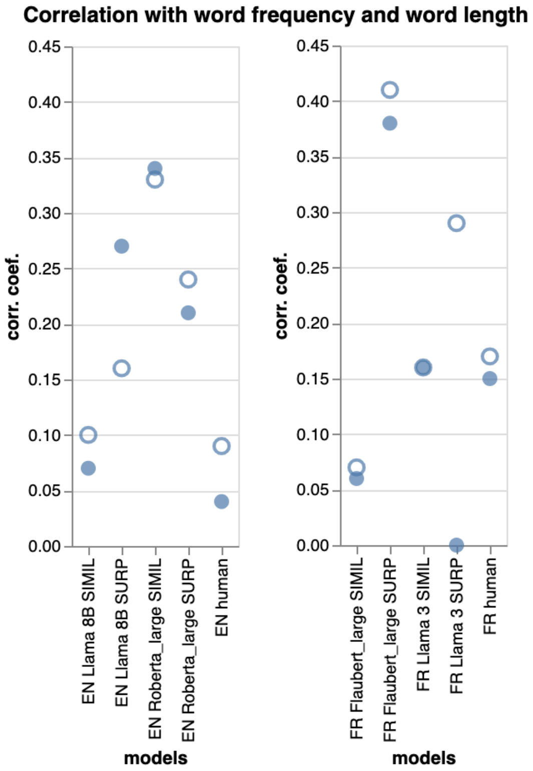


Figure 6: Correlation (Pearson) of word frequency and word length with the measures of semantic relatedness, surprisal and human rating. The empty circles represent word length, while the filled circles represent word frequency.

Model size. Regarding the models represented in our study, it would be interesting to compare masked BERT-based varieties with an autoregressive model that actually matches their size (such as GPT-2). Certain differences in polysemy processing were difficult to interpret since both varying architectures and wide-ranging model sizes were involved.

Model architecture. It is possible that the weaker correlation with human judgment achieved by the autoregressive language models is due to the initial psycholinguistic experiment design that favors bidirectional language models over unidirectional ones. Precisely, the authors first introduce the participants to a sentence with the target word masked. Afterwards, the participants see the full sentence and are asked to judge its plausibility on a scale from 1 to 100 by moving the cursor. This way, the human evaluators have a chance to consider the context on both sides before deciding on plausibility. In the surprisal setting, having access to the left context only, the autoregressive Llama does not mimic this behavior and does not align as closely with human estimation. Another argument in favor of this idea is that the subsequent analysis of the correlation of surprisal scores with the different regularity metrics showed that Llama delivers identical results to human evaluators for almost all regularity metrics. At the same time, RoBERTa, while generally following the correlation pattern, does not align as closely (see Figure 3 and Table 11, Appendix E). This suggests that our experimental design might poorly accommodate this model type, and we need other means of establishing whether the sensitivity to scalar regularity in models of different architectures varies.

Methodology. In order to understand the effect of such variables as word length, word frequency, and pattern regularity degree on the model scoring, we conducted correlation analyses and were able to gain insights from them. However, a deeper understanding of each variable's contribution to the results requires a regression study, which we plan to conduct as part of future work.

Is Incremental Structure Prediction Process Universal across Languages?: Revisiting Parsing Strategy through Speculation

Taiga Ishii and **Yusuke Miyao**
The University of Tokyo
`{taigarana,yusuke}@is.s.u-tokyo.ac.jp`

Abstract

While natural language is processed incrementally, it is unclear whether the syntactic structure prediction process is universal across languages or language-specific. This study investigates this question by revisiting parsing strategies of syntactic language models that incrementally predict both the next token and the associated syntactic structure. Unlike previous studies that have focused on a few strategies, we examine a wide range of strategies by introducing different parameterizations of "speculation", which quantifies the degree to which a model predicts syntactic structure before encountering the corresponding tokens. The experiments with 10 typologically diverse languages reveal that the optimal strategy differs depending on the language and the beam size.

1 Introduction

Understanding how syntactic structure is incrementally processed during language comprehension is a fundamental challenge in computational linguistics and cognitive science. Syntactic language modeling (SLM), also known as syntax-aware language modeling, provides a direct approach to addressing this question (Choe and Charniak, 2016; Dyer et al., 2016; Qian et al., 2021; Sartran et al., 2022). SLM is a task that jointly performs parsing and next-token prediction, thereby explicitly modeling the interplay between syntactic structure and incremental sequence processing. This approach has proven valuable for offering insights into the cognitive mechanisms of human language processing (Hale et al., 2018; Yoshida et al., 2021; Sugimoto et al., 2024).

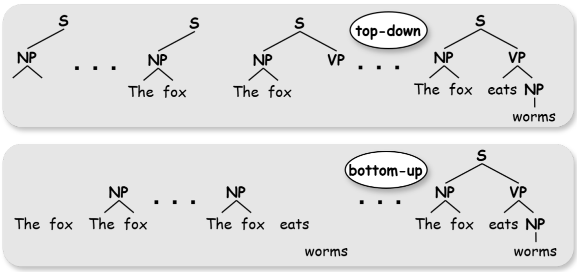


Figure 1: Example of incremental structure prediction process.

While SLM provides a framework for modeling syntactic processing, there exist multiple ways to incrementally process the same sequence of tokens and syntactic structures depending on the timing of structure prediction (Figure 1). These differences in processing are captured by the concept of "parsing strategy," (Abney and Johnson, 1991). For example, Figure 1 illustrates the two most commonly used strategies in parsing: top-down and bottom-up. Top-down is a strategy that predicts structure before tokens, while bottom-up is a strategy that predicts structure after tokens. Previous studies in SLM, however, have primarily focused on a limited set of strategies, such as top-down, bottom-up, and left-corner, and, moreover, lack cross-linguistic comparisons (Kuncoro et al., 2018; Yoshida et al., 2021), leaving it unclear whether optimal strategies are universal or language-specific.

This paper aims to address this gap in the literature by conducting a comprehensive analysis of parsing strategies for SLM across a diverse set of languages. To this end, we explore a wide range of parsing strategies from the perspective of "speculation", which quantifies the degree to which a model predicts syntactic structure before encountering the corresponding tokens. For example, the top-down strategy is highly speculative because it cannot use token information for structure prediction, and the predicted structure may be incorrect depending on subsequent tokens. In this work, we consider strategies based on 4 different parameterizations of speculation and evaluate a total of 15 distinct strategies on SLM tasks in 10 typologically diverse languages. While less speculative strategies might intuitively seem more advantageous, our experiment demon-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 437–451
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

strates that it is not always the case: the optimal strategy can vary across languages and depends on the beam size. Furthermore, we also analyze the fundamental question: does syntactic structure contribute to token prediction? By comparing strategies with different degrees of speculation, we show that syntactic structure indeed captures information about tokens, while also suggesting that exact parsing might not be necessary for token prediction. The implementation code is available at `https://github.com/mynlp/optimal-strategy.git`.

2 Background

Early studies argued that the left-corner strategy is more efficient and cognitively plausible than top-down or bottom-up strategies (Abney and Johnson, 1991; Resnik, 1992).[1] These arguments relied primarily on analyzing the maximum stack size required by shift-reduce parsers (Abney and Johnson, 1991; Resnik, 1992; Noji and Miyao, 2014). However, as Resnik (1992) points out, the difference in stack efficiency between strategies depends on the specific implementation of the parser. For instance, implementations of Recurrent Neural Network Grammar (RNNG) (Dyer et al., 2016; Noji and Oseki, 2021) require $O(n)$ stack size for right-branching structures even with the top-down strategy unlike claimed to be $O(1)$ in (Abney and Johnson, 1991). Therefore, it is unclear to what extent stack efficiency influences the choice of incremental processing strategies.

In the context of SLM, recent studies have explored the impact of different parsing strategies on downstream tasks such as language modeling and parsing. For example, Kuncoro et al. (2018) compared top-down, bottom-up, and left-corner strategies for English number agreement, finding that top-down parsing yielded better performance. Yoshida et al. (2021) compared top-down and left-corner strategies for Japanese language modeling, demonstrating the effectiveness of the left-corner strategy. Kuribayashi et al. (2024) compared top-down and left-corner strategies using an artificial language dataset with varying word order. However, these studies are limited in several aspects. First, they focus on a limited set of parsing strategies, e.g., top-down, bottom-up, and left-corner, due to the ease of implementation. Second, there is a lack of comprehensive cross-linguistic comparisons using real-world natural language data, leaving it unclear whether optimal parsing strategies are universal or language-specific.

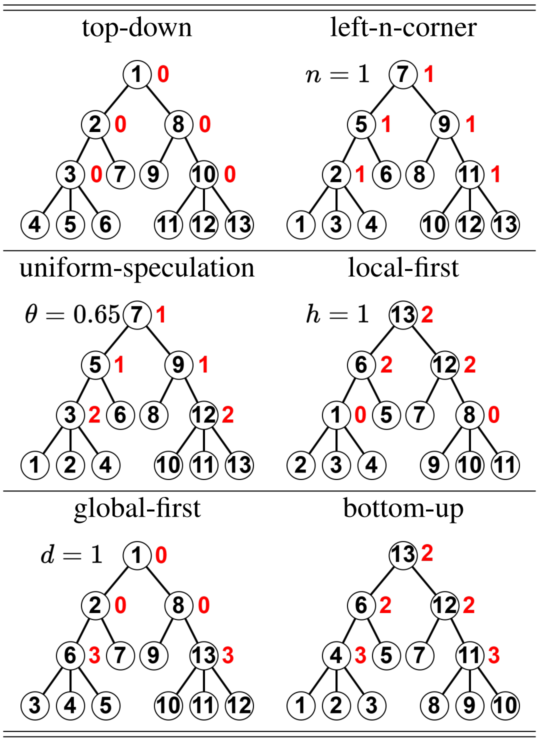


Table 1: Examples of parsing strategies. The numbers inside the circles indicate the order of node enumeration, and the numbers to the right of each nonterminal node represent its i_v.

To this end, this study conducts a more comprehensive analysis of parsing strategies for SLM, both in terms of strategies and languages.

3 Formulating Various Strategies

Following the general formulation of Abney and Johnson (1991), we formalize various parsing strategies. The difference between parsing strategies is defined by the timing at which each nonterminal node is opened. This allows us to express each strategy as a specific enumeration order of the nodes in a syntactic tree. Abney and Johnson (1991) demonstrated that different parsing strategies can be represented by strategy parameters i_v for each node v.[2] Let $u_1, \ldots, u_n$ be the children of v; $i_v = i$ indicates that the parent node v is opened immediately after its i-th child u_i is completed. The case of $i_v = 0$ indicates that v is opened before any of its children are created. By assigning i_v to every node v in a given syntactic tree, we

---

[1] The left-corner strategy predicts a phrase structure immediately after reading the leftmost token of that phrase.

[2] While Abney and Johnson (1991) originally defined the parameters for grammar rules, we generalize it to the nodes in syntactic trees.

can uniquely determine an incremental process of predicting the syntactic tree. Strategies represented by this parameterization are called syntax-directed strategies (Abney and Johnson, 1991).

In this study, we formulate a variety of distinct strategies within the class of syntax-directed strategies to investigate whether the optimal strategy is language-universal or language-specific. Our formulation is based on the concept of "speculation", which refers to the degree to which a model predicts syntactic structure before encountering the corresponding tokens. We consider 4 different parameterizations of speculation, each capturing a different aspect of this concept. By exploring multiple parameter settings within each parameterization, we analyze a total of 15 strategies. Table 1 shows some examples of the strategies used in this study. Note that both top-down and bottom-up strategies can be expressed by specific parameter settings within any of the four parameterizations.

3.1 Left-n-corner strategy

Besides top-down and bottom-up strategies, the left-corner strategy is another major strategy used in parsing research. In this study, we also experiment with a generalization of the left-corner strategy formulated by Abney and Johnson (1991), which we refer to as the "left-n-corner strategy".[3] In a left-n-corner strategy, the parent node v is predicted after at most n of its children have been completed. Formally, left-n-corner strategies are defined by a speculation parameter n as $i_v = \min(n, n_v)$. When $n = 0$, the left-n-corner strategy is equivalent to the top-down strategy. When $n = \infty$, it is equivalent to the bottom-up strategy.

3.2 Uniform-speculation Strategy

In the left-n-corner strategies, the number of children completed before predicting the parent n is constant for all nodes. However, with this parameterization, whether the timing of opening the parent node v is closer to top-down or bottom-up can vary across nodes, depending on the number of children n_v. Therefore, in this study, we introduce strategies in which the timing of opening the parent node v is less dependent on n_v and is consistent across all nodes.

Intuitively, this strategy, which we call the "uniform-speculation strategy", is defined by a real-valued speculation parameter $\theta \in [0, 1]$, representing the proportion of children created before the parent. For a node v with n_v children, i_v is calculated as $i_v = \lfloor \theta \cdot (n_v + 1) \rfloor$. Here, $\theta \to 0$ corresponds to strategies closer to top-down, while $\theta \to 1$ corresponds to strategies closer to bottom-up.

3.3 Local/global-first Strategy

The two strategies discussed above, left-n-corner and uniform-speculation, determine the timing of opening a node v independently of its position within the syntactic tree. In this study, we also analyze strategies where the timing of opening v – that is, the degree of speculation – varies depending on whether v belongs to a local or global structure.

Defining whether a structure is local or global is not trivial. Here, we use the height and depth of each node to define local and global structures, and use these as parameters to control the degree of speculation of the strategies. Intuitively, nodes closer to leaf nodes, i.e., nodes with smaller height, are considered local, while nodes closer to the root node, i.e., nodes with smaller depth, are considered global.

First, we consider a "local-first strategy", which predicts local structures in a top-down manner and global structures in a bottom-up manner. Specifically, the speculation parameter of this strategy is a height threshold h:

$$i_v = \begin{cases} 0, & \text{if } h_v \le h \\ n_v, & \text{otherwise} \end{cases}$$

where h_v is the height of node v.[4]

Similarly, we can also consider a "global-first strategy", which predicts global structures in a top-down manner and local structures in a bottom-up manner. This strategy is parameterized by a depth threshold d as follows:

$$i_v = \begin{cases} 0, & \text{if } d_v \le d \\ n_v, & \text{otherwise} \end{cases}$$

where d_v is the depth of node v.[5]

When $h \to \infty$, the local-first strategy is closer to top-down, and when $h = 0$, it is equivalent to bottom-up. Similarly, when $d \to \infty$, the global-first strategy is closer to top-down, and when $d < 0$, it becomes bottom-up.

---

[3] This formulation is called "uniform syntax-directed strategy" in (Abney and Johnson, 1991). However, we use the name left-n-corner instead to emphasize that it is a generalization of the left-corner strategy.

[4] We define the height of leaf nodes to be 0.

[5] We define the depth of the root node to be 0.

4 Shift-reduce Syntactic Language Modeling

This section formalizes the syntactic language modeling task (SLM). In SLM, structure prediction is typically performed by a shift-reduce parser with a stack (Dyer et al., 2016; Noji and Oseki, 2021; Choe and Charniak, 2016; Qian et al., 2021; Sartran et al., 2022; Kuncoro et al., 2018). Stack-based parsing is performed by predicting a sequence of actions defined as stack operations. However, previous work designed a separate action set for each parsing strategy, making it difficult to handle various strategies within a unified framework (Kuncoro et al., 2018). To address this limitation, we generalize the action set used by a shift-reduce parser to represent a wide range of strategies with a single, unified set of actions.

4.1 Generalizing Shift-reduce Actions

A simple approach to represent various strategies with a single action set is to extend the stack operations beyond push and pop to include an "insert" operation. This allows us to open nonterminal nodes at different positions within the stack, effectively controlling the timing of structure prediction. Specifically, we define the following action set:

- NT(X; n): Inserts an open nonterminal node "(X" at the n-th position from the top of the stack, opening a phrase with category X. Note that a new phrase cannot be opened deeper than any already open phrase.[6]
- SHIFT: Pushes the next token onto the stack.
- REDUCE: Completes the topmost open phrase on the stack, popping and combining its elements into a single constituent.

While strategies other than top-down typically require a special FINISH action to terminate the parsing process (Kuncoro et al., 2018), we do not explicitly introduce a FINISH action. Instead, we terminate the parsing process when the end-of-sentence (EOS) token is shifted. This simplifies the formulation of syntactic language modeling and the beam search procedure, which will be described later.

This generalized action set can represent various parsing strategies by restricting how actions are selected. For example, if the position to open a phrase is always $n = 0$, i.e., the top of the stack, the strategy becomes equivalent to top-down. If REDUCE action is always performed immediately after NT(X; n) action, the strategy becomes equivalent to bottom-up, because the prediction of a phrase with n children always occurs after all its children are completed.

4.2 Model Formulation

First, we introduce the notations used to formulate SLM. Let $\mathcal{A}$ be the set of actions defined above. We define $A_k \subset \mathcal{A}^*$ as the set of action sequences that contain exactly k SHIFT actions and end with a SHIFT action. For an action sequence $a = (a_1, \ldots, a_T)$, let l_i denote the index of the i-th SHIFT action a_{l_i} in a.

Given a token sequence x and an action sequence a, the syntactic language model $\mathcal{M}$ defines the following joint probability:

$$p^{\mathcal{M}}_{\text{joint}}(x, a) \equiv \prod_{t=1}^{|a|} p^{\mathcal{M}}_{\text{action}}(a_t \mid a_{<t}, x_{\le s(a_{<t})}) \cdot \prod_{i=1}^{|x|} p^{\mathcal{M}}_{\text{token}}(x_i \mid a_{<l_i}, x_{<i}),$$

where $p^{\mathcal{M}}_{\text{joint}}$ is the joint distribution of the token sequence and the parsing action sequence, $p^{\mathcal{M}}_{\text{action}}$ is the conditional probability of the next parsing action, $p^{\mathcal{M}}_{\text{token}}$ is the conditional probability of the next token, and $s(a_{<t})$ denotes the number of SHIFT actions in the given action sequence. While the probability of generating a token is not typically separated into $p^{\mathcal{M}}_{\text{action}}$ and $p^{\mathcal{M}}_{\text{token}}$ in the formulation, the probabilities are typically separated in the implementations (Dyer et al., 2016; Noji and Oseki, 2021). Here, we introduce a formulation that aligns more closely with actual implementations. During supervised training, the model is trained to maximize $\log p^{\mathcal{M}}_{\text{joint}}(x, a)$ on the train dataset.

The probability distribution over token sequences of length $|x|$ is computed as follows:

$$p^{\mathcal{M}}(x) = \sum_{a \in A_{|x|}} p^{\mathcal{M}}_{\text{joint}}(x, a).$$

To calculate the probability distribution over sentences of arbitrary length, one can simply calculate $p^{\mathcal{M}}$ for token sequences x that end with the EOS token.

4.3 Modeling Incremental Inference Process

The goal of this study is to evaluate the incremental structure prediction process in natural language.

---

[6] This restriction is for implementation simplicity.

Previous work on SLM has primarily focused on evaluating models by approximating $p^{\mathcal{M}}$ using a trained model $\mathcal{M}$.

Approaches to approximating $p^{\mathcal{M}}$ in SLM can be broadly categorized into two types. The first approach uses candidate actions $\tilde{A}$ obtained from an external parser (Dyer et al., 2016; Kuncoro et al., 2018; Sartran et al., 2022). The second approach uses word-synchronous beam search (Stern et al., 2017) and approximates $p^{\mathcal{M}}$ by the set of inferred action sequences (Hale et al., 2018; Noji and Oseki, 2021; Yoshida et al., 2021), which we denote by $\tilde{p}^{\mathcal{M}}$. In this study, we focus on the latter approach since the former does not involve inference with the SLM model itself.

The process of word-synchronous beam search aims to model the joint prediction of the next token and its corresponding syntactic structure. For a token sequence x, the process can be represented by a sequence of sets of action sequences ending with SHIFT: $B_0, B_1, \ldots, B_{|x|}$. Here, B_i represents the set of (partial) syntactic structures in the beam when predicting token x_i, corresponding to the i-th step of word-synchronous beam search, and satisfying $B_i \subseteq A_i$. Note that $B_0 = \emptyset$, and each B_i is deterministically computed based on Algorithm 1. While previous work (Stern et al., 2017) introduces a word beam bottleneck, we instead limit the maximum number of actions between SHIFT actions to k_n to reduce inference time. The score function for selecting an action sequence $b'c$ is the joint probability:

$$\begin{cases} p^{\mathcal{M}}_{\text{joint}}(x_{<i}x_i, b'c), & \text{if } c == \text{SHIFT}, \\ p^{\mathcal{M}}_{\text{joint}}(x_{<i}, b'c), & \text{otherwise.} \end{cases}$$

5 Experiments

Evaluation. Here, we describe the overall flow of the experiments. For each treebank and strategy, we convert the gold trees to action sequences and train a base model $\mathcal{M}$ in a supervised manner. We then perform inference using word synchronous beam search with the trained model to obtain the set of action sequences $B_{|x|}$. We evaluate performance across a range of beam sizes, $k \in \{50, 200, 800\}$. To reduce inference time, we utilize fast-track selection with $k_s = k/50$ and limit the maximum number of actions between SHIFT actions to $k_n = 20$. For each setting, we train models with 3 different random seeds and report the average performance.

Algorithm 1 Word synchronous beam search with fast-track selection and a step limit.

```
Input: $x_{\leq i}$                          ▷ Token sequence
Input: $k$                                   ▷ Beam size
Input: $k_s$          ▷ Number of fast-tracked samples
Input: $k_n$          ▷ Maximum number of actions
  between SHIFT actions
Input: $B_{i-1}$                             ▷ Last beam
  $B'_i \leftarrow B_{i-1}$
  for $j = 1, \ldots$ do
      $C_{\text{fast}} \leftarrow \text{top}k_s(\{b' \cdot \text{SHIFT} \mid b' \in B'_i\})$
      $B_i \leftarrow B_i \cup C_{\text{fast}}$        ▷ Fast-track selection
      $C \leftarrow \bigcup_{b' \in B'_i} \{b'c \mid c \in \mathcal{A}\}$
      $B'_i \leftarrow \text{top}k(C \setminus C_{\text{fast}})$   ▷ Select candidates
      for $b'c \in B'_i$ do
          if $c ==$ SHIFT then
              $B_i \leftarrow B_i \cup \{b'c\}$      ▷ Update beam
              $B'_i \leftarrow B'_i \setminus \{b'c\}$
      if $|B_i| = k \vee j \geq k_n$ then
          Break   ▷ Quit search when the beam is
  full or the step limit is reached
  return $B_i$
```

Dataset. We use treebanks from 10 languages: English (Penn Treebank (Marcus et al., 1993)), Chinese (Chinese Treebank (Palmer et al., 2005)), French, German, Korean, Basque, Hebrew, Hungarian, Polish, and Swedish (SPMRL (Seddah et al., 2013)). Following Noji and Oseki (2021), we remove POS tags and split words into subwords. All evaluations in this paper are performed on the validation datasets. To reduce the size of the action set and simplify model training, we limit the n in NT(X;n) actions to a maximum of 10. To ensure consistent parsability across strategies, we restrict the train and validation data to instances where the gold trees are parsable by all strategies with $n \leq 10$. Furthermore, we only use sentences that are parsable with $n \leq 10$ and $k_n = 20$ for evaluation. Further details are provided in Appendix A.

Strategy. In our experiments, we analyze a total of 15 strategies: top-down, bottom-up, left-n-corner with $n \in \{1, 2, 3\}$, uniform-speculation with $\theta \in \{0.26, 0.35, 0.65, 0.74\}$, local-first with $h \in \{1, 2, 3\}$, and global-first with $d \in \{1, 2, 3\}$.[7] For simplicity, we consider the insertion position of NT actions at the subword level rather than the word level.

---

[7] The values of θ are chosen such that i_v changes for a node v with $n_v = 2, 3, 4$ depending on θ.

Beam	English	Chinese	French	German	Korean
50	**BU** (88.7±0.3)	**LC-1** (86.1±0.2)	**TD** (81.8±0.2)	**LC-1** (86.4±0.1)	**BU** (84.5±0.1)
	LF-2 (87.3±0.1)	**BU** (86.1±0.1)	BU (79.6±0.1)	LC-2 (85.8±0.1)	LC-2 (84.1±0.1)
200	**LF-2** (89.4±0.1)	**LC-1** (87.0±0.2)	**TD** (83.3±0.2)	**LC-1** (87.3±0.1)	**BU** (84.5±0.1)
	LF-3 (89.4±0.0)	BU (86.6±0.3)	US-0.26 (81.1±0.1)	LC-2 (86.5±0.0)	LF-1 (84.2±0.1)
800	**TD** (90.9±0.1)	**LC-1** (87.0±0.2)	**TD** (83.7±0.2)	**TD** (87.7±0.1)	**BU** (84.4±0.1)
	LF-3 (90.2±0.0)	BU (86.7±0.2)	US-0.26 (81.8±0.1)	LC-1 (87.4±0.1)	LF-1 (84.2±0.1)

Beam	Basque	Hebrew	Hungarian	Polish	Swedish
50	**BU** (83.0±0.1)	**LF-1** (80.8±0.3)	**LC-1** (87.2±0.1)	**GF-1** (78.9±0.3)	**LC-1** (72.8±0.2)
	LF-1 (82.8±0.1)	LC-1 (80.5±0.3)	LC-2 (86.6±0.1)	BU (77.1±0.1)	US-0.26 (69.8±0.1)
200	**BU** (83.1±0.1)	**TD** (82.2±0.3)	**LC-1** (88.1±0.1)	**GF-1** (79.5±0.1)	**LC-1** (73.5±0.1)
	LC-1 (83.1±0.2)	LF-1 (81.6±0.3)	LC-2 (87.1±0.0)	BU (77.2±0.2)	TD (73.0±0.3)
800	**LF-1** (83.3±0.2)	**TD** (83.7±0.2)	**LC-1** (88.1±0.1)	**GF-1** (79.5±0.1)	**TD** (74.9±0.3)
	LC-1 (83.1±0.2)	LF-3 (82.3±0.2)	TD (87.9±0.1)	BU (77.0±0.3)	LC-1 (73.6±0.2)

Table 2: Top-2 strategies for the labeld parsing f1 scores for each dataset and beam size. TD and BU denote top-down and bottom-up strategies, and LC, US, LF, and GF denote left-n-corner, uniform-speculation, local-first, and global-first strategies with their corresponding parameters. Mean f1 scores and standard errors are shown in the parentheses.

Model. For the model, we extend the commonly used syntactic language model the Recurrent Neural Network Grammar (RNNG) (Dyer et al., 2016) to handle the proposed generalized shift-reduce action set. The implementation is based on the batched version of RNNG (Noji and Oseki, 2021). For the action set implementation, we simply represent SHIFT, REDUCE, and each NT(X;n) action by one-hot vectors. For each setting, we train a model for either 80 epochs or 8000 steps, whichever is larger, and evaluate the model with the lowest validation loss. Details of the training settings are provided in Appendix B.

5.1 Results on Parsing

First, we analyze parsing performance. We calculate the labeled F1 score using the highest-scoring action sequence in $B_{|x|}$. Table 2 shows the top two performing strategies for each language, and Figure 2 presents the parsing performance for all strategies. Note that in Figure 2, strategies are sorted from left to right in descending order of speculation degree, i.e., from top-down to bottom-up, for each speculation parameterization. The results reveal that the strategy that maximizes parsing performance depends on the language and beam size. For example, for English, bottom-up performs best when $k = 50$, local-first ($h = 2, 3$) performs best when $k = 200$, and top-down when $k = 800$. Similarly, top-down shows higher F1 scores than other strategies for French, German, Hebrew, and Swedish when $k = 800$. In contrast, for Chinese, Korean, and Basque, bottom-up, left-n-corner ($n = 1$), or local-first ($h = 1$) obtain higher F1 scores for all beam sizes. For these languages, the performance of top-down is lower compared to other strategies, especially when the beam size is small (Figure 2). The sentence probability marginalized over the beam, $\tilde{p}^{\mathcal{M}}$, showed a similar overall trend to the parsing performance. We show the results for $\tilde{p}^{\mathcal{M}}$ in Appendix C.

5.2 Results on Structure-conditioned Token Probability

Figure 3 shows the perplexity based on the $p^{\mathcal{M}}_{\text{token}}$ for the best action sequence obtained by beam search for English, Chinese, German, and Korean.[8][9] Generally, higher speculation leads to lower perplexity, i.e., higher $p^{\mathcal{M}}_{\text{token}}$, regardless of the speculation parameterization. However, for Chinese and Korean, perplexity tends to be higher when the degree of speculation is too high when the beam size is smaller.[10]

5.3 Additional Experiments for Polish

The experimental results for Polish in this section are based on the standard preprocessing, where

---

[8] This is different from the sentence probability $p^{\mathcal{M}}$.

[9] The results for other languages are shown in Appendix C.

[10] Basque and Hungarian also show similar trend (Appendix C).

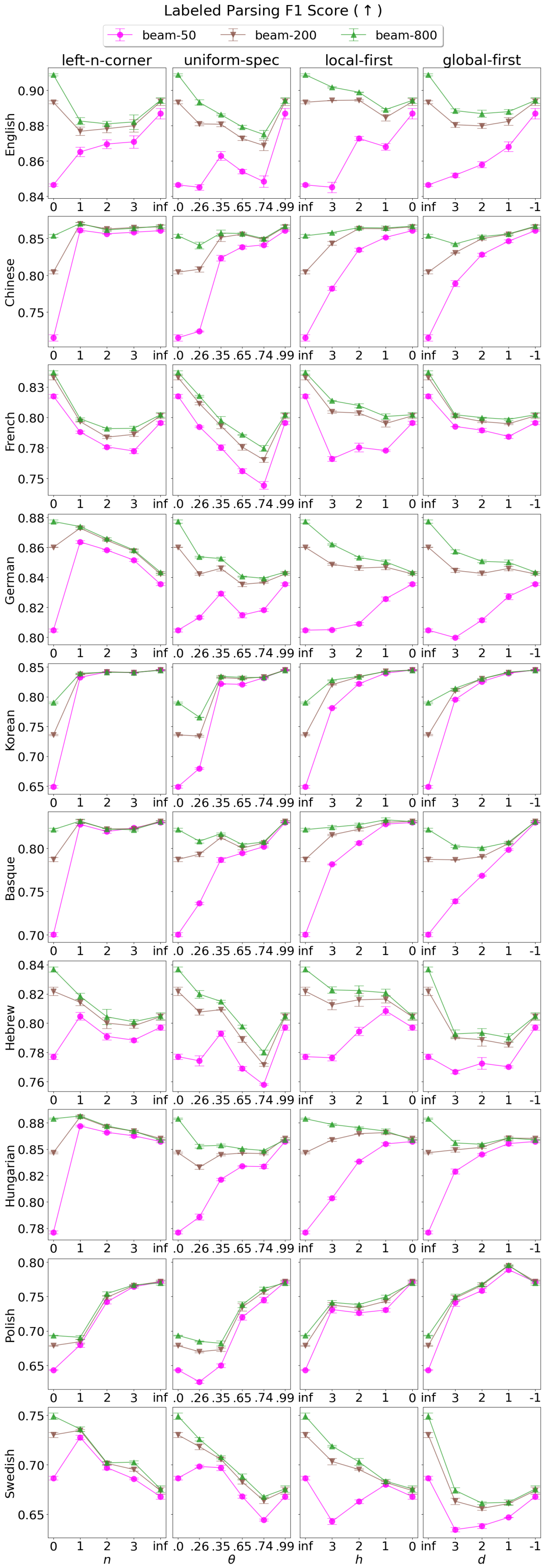


Figure 2: Labeled parsing F1 scores for all datasets. Error bars show the standard error of the mean.

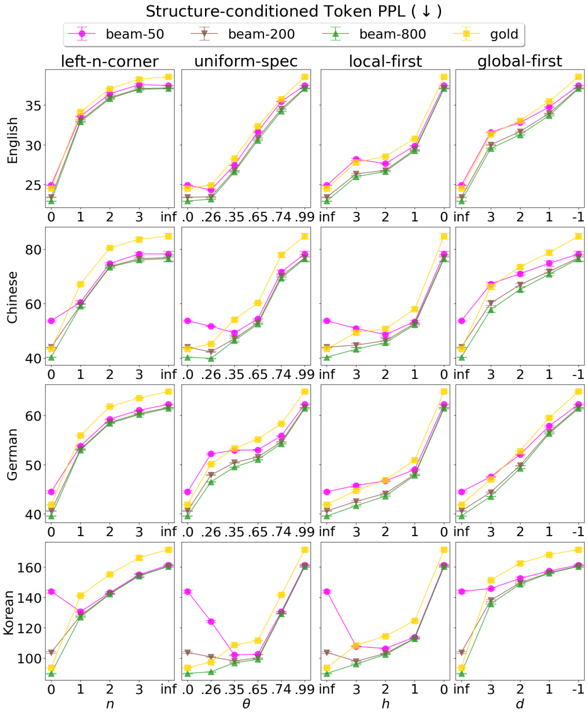


Figure 3: Perplexity based on $p^{\mathcal{M}}_{\text{token}}$. Error bars show the standard error of the mean.

preterminal nodes are removed. As observed in Table 2 and Figure 2, the results for Polish exhibit a distinct pattern from that of English. We found that this is due to an idiosyncratic structure in the Polish treebank; specifically, the lowest layer of nonterminals, i.e., nonterminals immediately above preterminals, functions similarly to standard preterminals. We conducted additional experiments and found that when both the lowest layer nonterminals and preterminals are removed, Polish exhibits a pattern similar to English. Further details are provided in Appendix D.

6 Discussion

6.1 Is the Optimal Strategy Universal across Languages?

The results of the experiments suggest that the optimal strategy for incremental structure prediction in syntactic language models is not universal across languages, but rather language-specific. Previous research has suggested that left-corner is a better strategy due to its stack size efficiency, but our findings indicate that it is not necessarily the best in practical tasks.

What factors contribute to these differences between languages? If we simply consider the amount of information available during inference, less speculative strategies should be advantageous

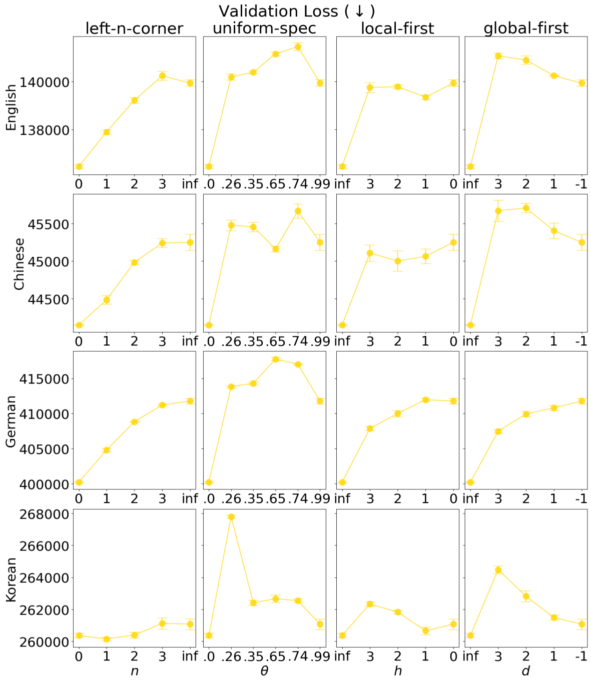


Figure 4: Validation loss, i.e., $-\log p^{\mathcal{M}}_{\text{joint}}$. Error bars show the standard error of the mean.

even with larger beam sizes. However, contrary to this expectation, top-down outperforms less speculative strategies in some languages. We hypothesize that this is due to a combination of two factors: the ease of learning of each strategy and the required parallel inference capacity.

First, Figure 4 shows the validation loss, i.e., negative joint log-likelihood $-\log p^{\mathcal{M}}_{\text{joint}}$, for English, Chinese, German, and Korean for the same data points as in Figure 2.[11] Generally across all languages except Korean, top-down has the lowest loss, followed by left-n-corner ($n = 1$), indicating that these strategies, especially top-down, are easier to learn.[12]

Second, top-down requires larger beam size, i.e., parallel inference capacity, than other less speculative strategies because top-down cannot use token information to predict structures. Furthermore, top-down requires even larger beam size for left-branching languages as discussed in the previous work (Abney and Johnson, 1991; Yoshida et al., 2021).

Overall, the top-down strategy exhibits a tradeoff between ease of learning, which contributes to strong performance, and the difficulty of inference due to the required large beam size. The differences in the optimal strategy across languages might be attributed to differences in the balance of this tradeoff. For example, in English, German, Hebrew, and Swedish, the parsing performance of top-down is low when the beam size is small, but it significantly improves as the beam size increases, becoming the best strategy at $k = 800$ (Figure 2). In Chinese and Korean, which are more left-branching and thus expected to require larger beam size than English, the performance of top-down tends to be lower than that of less speculative strategies like bottom-up, even with beam size $k = 800$. However, given the lower validation loss of the top-down strategy (Figure 4), it may be possible that top-down could become competitive with or even outperform less speculative strategies, even for these languages, with a sufficiently large beam size.

6.2 Does syntactic structure contribute to token prediction?

In speculative strategies, token prediction is conditioned on the already-predicted syntactic structures. Thus, if $p^{\mathcal{M}}_{\text{token}}$ increases with the degree of speculation, i.e., the amount of structures usable for token prediction, syntactic structure is likely to be informative for token prediction. As shown in Figure 3, $p^{\mathcal{M}}_{\text{token}}$ tends to increase with the degree of speculation, suggesting that syntactic structure indeed captures information about tokens. For some languages, e.g., Korean, Chinese, and Basque, $p^{\mathcal{M}}_{\text{token}}$ decreases for more speculative strategies, likely due to inference failure. Nevertheless, with gold actions, $p^{\mathcal{M}}_{\text{token}}$ increases with the degree of speculation across all languages, which also supports the informativeness of syntactic structures.

Meanwhile, the token probability conditioned on the gold tree is lower than that conditioned on the structures inferred by the model for most languages and strategies with the exception of highly speculative strategies with small beam sizes. This result suggests that, from the perspective of token prediction, a certain level of parsing accuracy is sufficient, and exact parsing may not be necessary. In fact, it is also argued that human language processing only utilizes partial shallow structures (Sanford and Sturt, 2002; Ferreira et al., 2002; Ferreira and Patson, 2007), and Noji and Oseki (2023) showed that syntactic ablation, i.e., removing some syntactic categories, improves the syntactic generalization ability of top-down models in English. Therefore,

---

[11] The results for other languages are shown in Appendix C.

[12] For other strategies, except for global-first parameterization, we generally observe that lower speculation leads to better learning, i.e., lower validation loss. However, bottom-up sometimes shows lower loss than strategies other than top-down.

to further investigate the extent to which syntax is necessary for token prediction, it would be necessary to perform syntactic ablation across various strategies.

6.3 Future Directions

The experiments revealed that the optimal strategy depends on both language and beam size. This finding leads to a hypothesis: if humans and (large) language models possess different internal "beam sizes", i.e., parallel inference capacities, they might also employ distinct parsing strategies. Moreover, investigating whether the cross-lingual differences in incremental processing affect second or multi language acquisition is an interesting future direction.

Furthermore, the analysis showed that structures inferred by the models yield higher $p^{\mathcal{M}}_{\text{token}}$ than gold trees. This raises a hypothesis: the gold tree structures in natural language treebanks are not optimal with respect to token prediction. This potential discrepancy might also explain the low performance of unsupervised parsing models trained with a sequence reconstruction objective (Li et al., 2020). Analyzing this relationship to unsupervised parsing presents another promising direction for future research.

7 Conclusion

This study analyzed whether the incremental structure prediction process in natural language is universal across languages or language-specific. We considered a total of 15 strategies based on 4 different parameterizations of speculation. Experiments on 10 typologically diverse languages suggest that the optimal strategy can vary across languages and is influenced by two factors: the ease of learning and the required parallel inference capacity.

Furthermore, a comparison between strategies with different degrees of speculation reveals that the syntactic structure of natural language is indeed informative for token prediction, while also suggesting that exact parsing might not be necessary.

Finally, this study focused on phrase structure; however, natural language also encompasses other structures such as dependency and semantic structures. Future work examining strategies for such structures is expected to further reveal universals and differences across languages.

Limitations

Dataset. While this study showed that the optimal strategy can vary across languages, a significant limitation is our inability to pinpoint which specific linguistic properties or dataset characteristics are responsible for these differences. As we discussed in section 6, one possible factor is the branching direction. For example, the top-down strategy, which requires a larger beam size for left-branching languages such as Chinese and Korean, showed lower performance for these languages in the experiments. Nevertheless, a quantitative analysis is necessary to evaluate the impact of branching direction. Other factors, such as differences in annotation schemes or tokenization, could also contribute to the observed differences in the optimal strategies.

Moreover, this study was exclusively limited to constituency treebanks. Experiments using treebanks based on other grammar formalisms, such as dependency grammar, Head-driven Phrase Structure Grammar, and Combinatory Categorial Grammar, etc., might reveal different findings.

Strategy. Another limitation stems from the use of subword tokenization. As described in section 5, the insertion position of NT actions is at the subword level. Thus, the strategies used in this study are based on subword-level speculation and do not explicitly consider word boundaries. Strategies based on word-level speculation, which allow NTs to be opened only around word boundaries, await further investigation.

Furthermore, it is possible to define symmetric counterparts to the strategies used in this study. For example, given a strategy parameter i_v, we can define more bottom-up oriented strategies with a parameter $j_v = n_v - i_v$. Analysis of such complementary strategies has yet to be explored.

Model. Our study is limited to a specific syntactic language model, RNNG, with a fixed set of hyperparameters (Appendix B). Various other architectures have been proposed, such as PLM (Choe and Charniak, 2016) and Transformer Grammar (Sartran et al., 2022). How the inductive biases of different architectures and hyperparameters influence the optimal strategies remains an open question.

Additionally, as mentioned in section 2, RNNG is considered less sensitive to stack size. How the optimal strategy changes in models that are more strongly affected by stack size also remains unclear.

Acknowledgments

This work was supported by JST SPRING Grant Number JPMJSP2108 and JSPS KAKENHI Grant Number JP24KJ0666 and JP24H00087.

References

Steven P Abney and Mark Johnson. 1991. Memory requirements and local ambiguities of parsing strategies. *J. Psycholinguist. Res.*, 20(3):233–250.

Do Kook Choe and Eugene Charniak. 2016. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas. Association for Computational Linguistics.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Fernanda Ferreira, Karl G D Bailey, and Vittoria Ferraro. 2002. Good-enough representations in language comprehension. *Curr Dir Psychol Sci*, 11(1):11–15.

Fernanda Ferreira and Nikole D Patson. 2007. The 'good enough' approach to language comprehension. *Lang. Linguist. Compass*, 1(1-2):71–83.

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

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

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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

Tatsuki Kuribayashi, Ryo Ueda, Ryo Yoshida, Yohei Oseki, Ted Briscoe, and Timothy Baldwin. 2024. Emergent word order universals from cognitively-motivated language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14522–14543, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020. An empirical comparison of unsupervised constituency parsing methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3278–3283, Online. 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*.

Hiroshi Noji and Yusuke Miyao. 2014. Left-corner transitions on dependency parsing. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2140–2150, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Hiroshi Noji and Yohei Oseki. 2021. Effective batching for recurrent neural network grammars. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4340–4352, Online. Association for Computational Linguistics.

Hiroshi Noji and Yohei Oseki. 2023. How much syntactic supervision is "good enough"? In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2300–2305, Dubrovnik, Croatia. Association for Computational Linguistics.

Martha Palmer, Fu-Dong Chiou, Nianwen Xue, and Tsan-Kuang Lee. 2005. Chinese treebank 5.1 LDC2005T01U01.

Peng Qian, Tahira Naseem, Roger Levy, and Ramón Fernandez Astudillo. 2021. Structural guidance for transformer language models. 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 3735–3745, Online. Association for Computational Linguistics.

Philip Resnik. 1992. Left-corner parsing and psychological plausibility. In *COLING 1992 Volume 1: The 14th International Conference on Computational Linguistics*.

Anthony Sanford and Patrick Sturt. 2002. Depth of processing in language comprehension: not noticing the evidence. *Trends Cogn. Sci.*, 6(9):382.

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.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Éric Villemonte de la Clergerie. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182.

Mitchell Stern, Daniel Fried, and Dan Klein. 2017. Effective inference for generative neural parsing. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1695–1700, Copenhagen, Denmark. Association for Computational Linguistics.

Yushi Sugimoto, Ryo Yoshida, Hyeonjeong Jeong, Masatoshi Koizumi, Jonathan R Brennan, and Yohei Oseki. 2024. Localizing syntactic composition with left-corner recurrent neural network grammars. *Neurobiology of Language*, 5(1):201–224.

Ryo Yoshida, Hiroshi Noji, and Yohei Oseki. 2021. Modeling human sentence processing with left-corner recurrent neural network grammars. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2964–2973, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Datset Setting

To split the words into subwords, we applied byte pair encoding (BPE). For datasets with 13K-30K different words that appear at least twice (English, Chinese, French, German, Korean, and Hungarian), we used BPE with a vocabulary size of 5000. For the remaining datasets (Basque, Hebrew, Polish, and Swedish), which have 5K-8K words appearing at least twice, we used BPE with a vocabulary size of 1500. We used SentencePiece for subword segmentation.[13]

B Model Setting

For the hyperparameters of RNNG, we used a 2-layer LSTM (Hochreiter and Schmidhuber, 1997) for hidden state transitions, a BiLSTM as the composition model, 256-dimensional embedding vectors, 256-dimensional hidden state vectors, and a dropout rate of 0.3. For optimization, we used Adam (Kingma and Ba, 2015) with a learning rate of 0.001. Training was performed for either 80 epochs or 8000 steps, whichever was larger for each dataset. Regarding the batch size, we set it to 512 for datasets with more than 10K data points (English, Chinese, French, German, and Korean), and 128 for datasets with fewer than 10K data points (Basque, Hebrew, Hungarian, Polish, and Swedish).

C Other Results

Figure 5 shows the perplexity based on sentence probability $\tilde{p}^{\mathcal{M}}$, calculated by marginalizing the joint probability $p^{\mathcal{M}}_{\text{joint}}$ within the last beam $B_{|x|}$ to approximate $p^{\mathcal{M}}$, for each language and strategy. Figure 6 shows the perplexity calculated using the $p^{\mathcal{M}}_{\text{token}}$ for the best action sequence obtained by beam search for each language and strategy. Figure 7 shows the validation loss, i.e., the negative joint log-likelihood $-\log p^{\mathcal{M}}_{\text{joint}}$, calculated for the same data points as in Figure 2 for each language and strategy.

D Additional Experiments for Polish

This section presents an additional experiment for Polish involving a different preprocessing procedure. As described in section 5, our standard preprocessing removes preterminal nodes from the constituency trees. The Polish treebank, however, is an exceptional case. In the Polish treebank, the

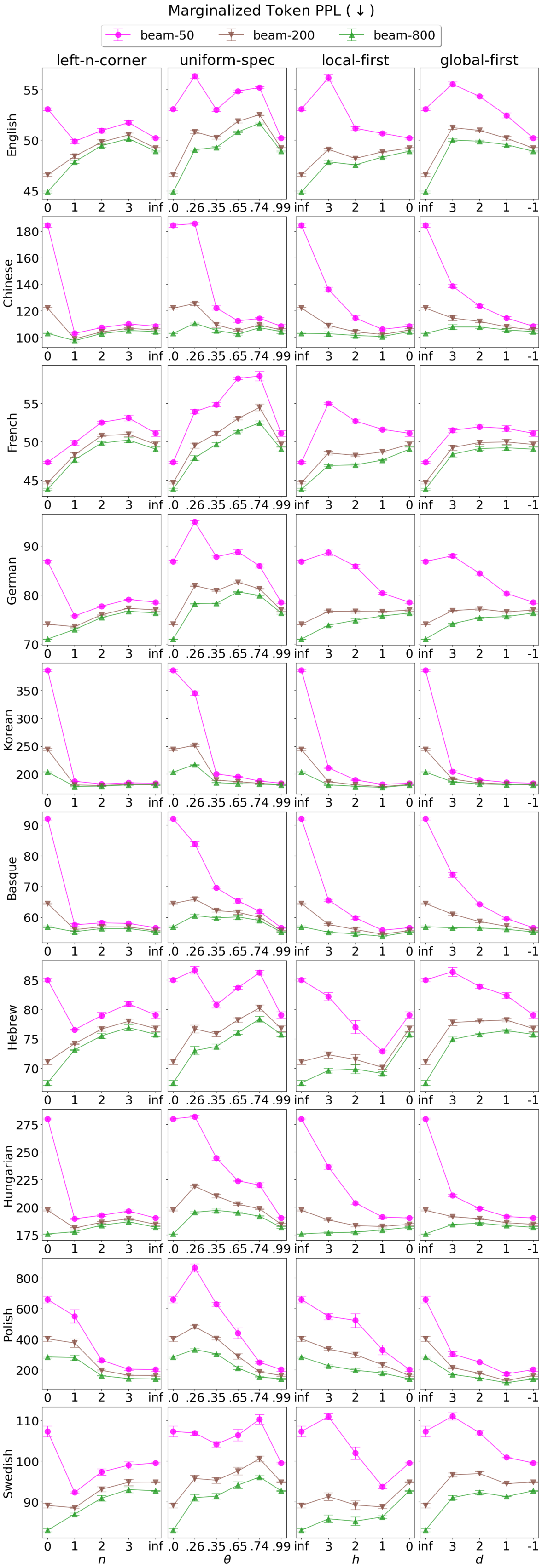


Figure 5: Perplexity based on $\tilde{p}^{\mathcal{M}}$ for all datasets. Error bars show the standard error of the mean.

---

[13] https://github.com/google/sentencepiece

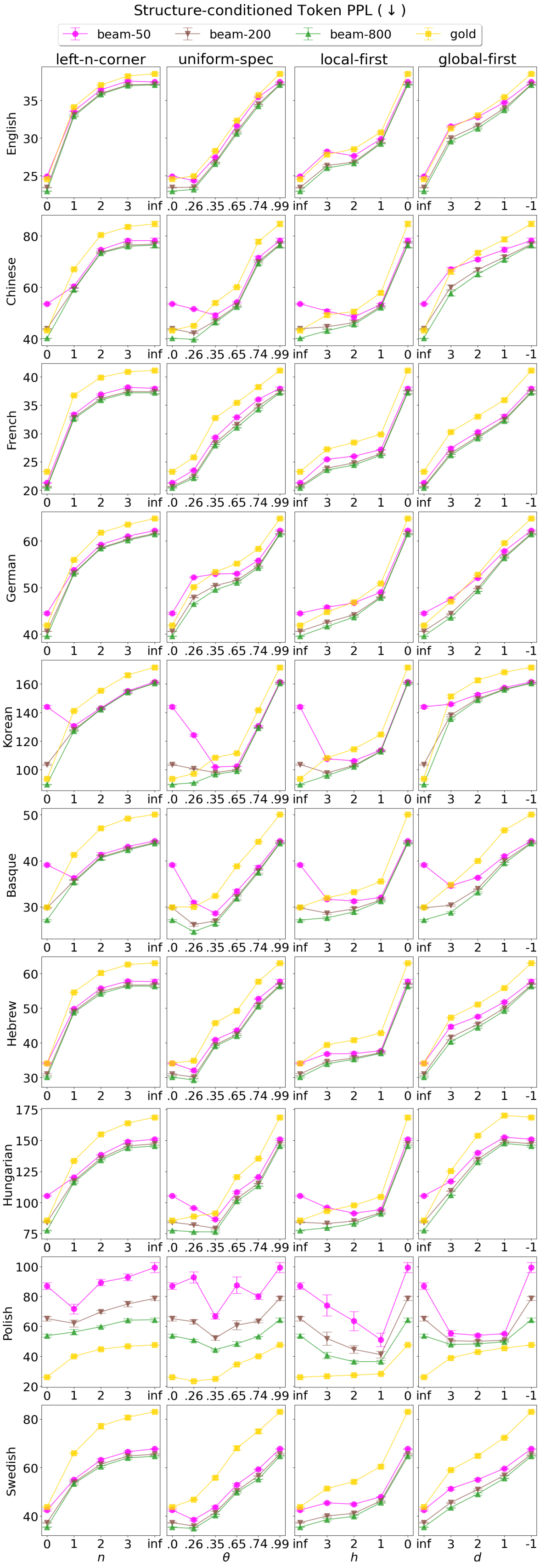


Figure 6: Perplexity based on $p_{\text{token}}^{\mathcal{M}}$ for all datasets. Error bars show the standard error of the mean.

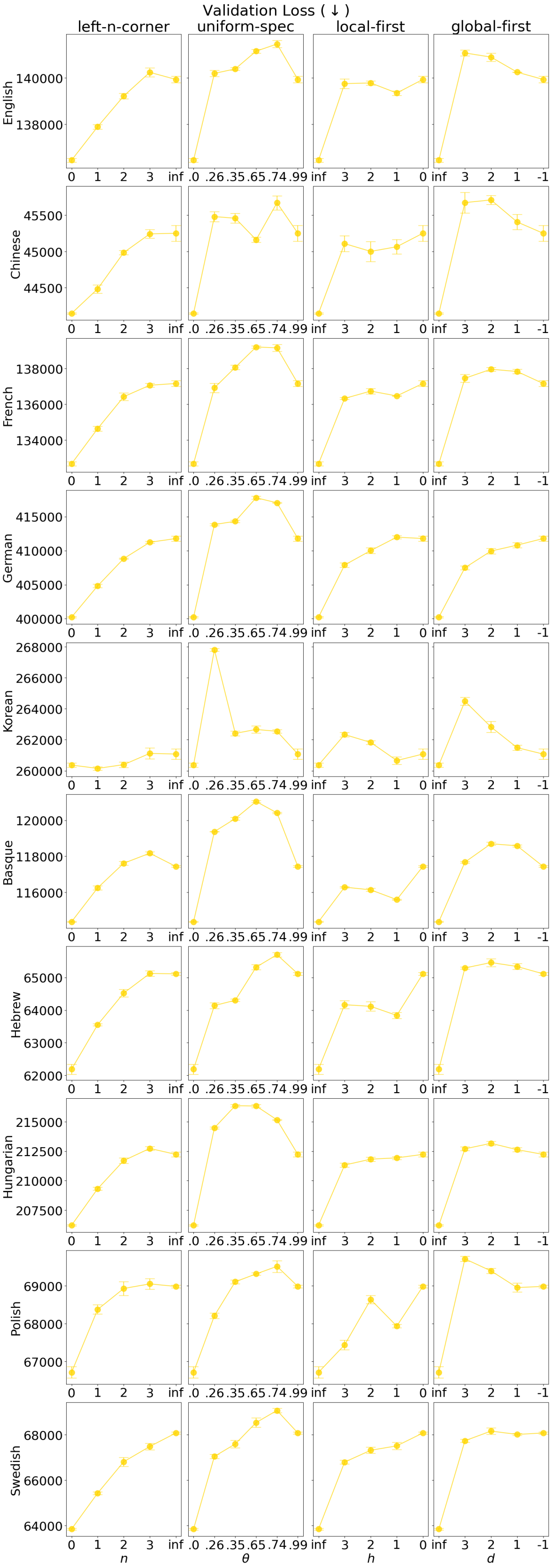


Figure 7: Validation loss, i.e., $-\log p_{\text{joint}}^{\mathcal{M}}$ for all datasets. Error bars show the standard error of the mean.

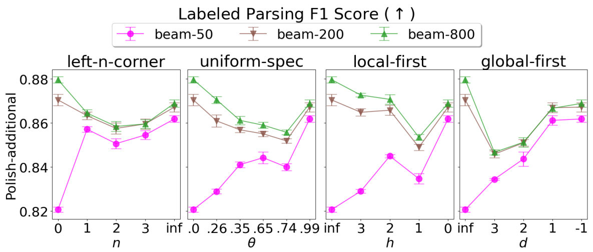


Figure 8: Labeled parsing F1 scores for Polish with lowest layer nonterminals removed. Error bars show the standard error of the mean.

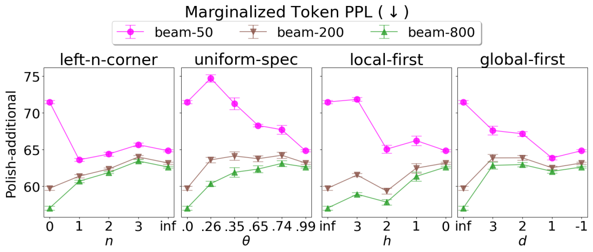


Figure 9: Perplexity based on $\tilde{p}^{\mathcal{M}}$ for Polish with lowest layer nonterminals removed. Error bars show the standard error of the mean.

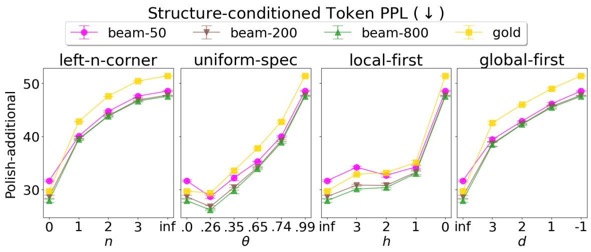


Figure 10: Perplexity based on $p^{\mathcal{M}}_{\text{token}}$ for Polish with lowest layer nonterminals removed. Error bars show the standard error of the mean.

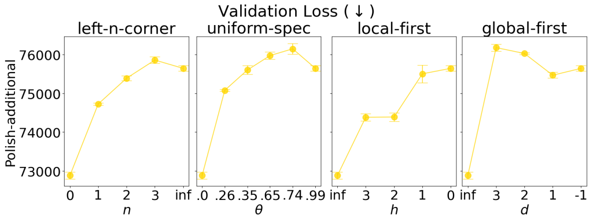


Figure 11: Validation loss, i.e., $-\log p^{\mathcal{M}}_{\text{joint}}$ for Polish with lowest layer nonterminals removed. Error bars show the standard error of the mean.

lowest layer of nonterminal nodes, i.e., those immediately above the preterminals, functions similarly to standard preterminals. Yet, these lowest layer nonterminals differ from standard preterminals in that they can be nested. The results for Polish shown in Figure 2, Figure 5, Figure 6 and Figure 7 are based on the standard preprocessing where only preterminals are removed, while preserving the lowest layer nonterminals. To investigate the effect of the lowest layer nonterminals, we also conduct experiments where models are trained on data with both the preterminals and the lowest layer of nonterminals removed.

Preprocess. To remove the lowest layer nonterminals including the nested ones, we remove all preterminal and nonterminal nodes within subtrees that have a minimum leaf depth of 2. Consequently, the leaf nodes are directly attached to the parent of the removed subtrees. Apart from this modification, all other settings are the same as the standard preprocessing procedure in section 5. We denote the dataset created with this preprocessing procedure as Polish-additional.

Results. Figure 8 shows the parsing performance for Polish-additional (corresponding results with standard preprocessing are shown in Figure 2). Figure 9 shows the perplexity based on sentence probability $\tilde{p}^{\mathcal{M}}$ for Polish-additional (corresponding results with standard preprocessing are shown in Figure 5). Figure 10 shows the perplexity calculated using the $p^{\mathcal{M}}_{\text{token}}$ for the best action sequence obtained by beam search for Polish-additional (corresponding results with standard preprocessing are shown in Figure 6). Figure 11 shows the validation loss, i.e., the negative joint log-likelihood $-\log p^{\mathcal{M}}_{\text{joint}}$ for Polish-additional (corresponding results with standard preprocessing are shown in Figure 7).

Interestingly, the experimental results reveal a significant difference between the Polish and Polish-additional. For instance, while Figure 2 shows that Polish and English exhibit distinct trends in parsing performance, the pattern for Polish-additional in Figure 8 closely resembles that of English. More specifically, for Polish (Figure 2), low speculation strategies, such as bottom-up and global-first ($d = 1$), tend to achieve higher scores across all beam sizes. In contrast, for Polish-additional (Figure 8), top-down yields the best performance at larger beam sizes, such as $k = 200, 800$. Furthermore, regarding the structure-conditioned token PPL (Figure 6), Polish is exceptional in that the PPL of the gold tree is lower than that of the structures inferred by models. For Polish-additional, on the other hand, the gold tree PPL is often higher than that of the inferred structures; this trend is consistent with other languages. Finally, in terms of validation loss, no significant difference is observed between Polish and Polish-additional.

These findings demonstrate that the optimal parsing strategy is sensitive to the presence or absence of the lowest layer of nonterminals in the Polish treebank. Given that Polish and Polish-additional show a significant difference in the performance patterns across strategies for parsing and structure-conditioned token PPL, while showing no such difference for validation loss, we hypothesize that the lowest layer nonterminals strongly influence the difficulty of inference.

Lost in Variation?
Evaluating NLI Performance in Basque and Spanish Geographical Variants

Jaione Bengoetxea and **Itziar Gonzalez-Dios** and **Rodrigo Agerri**
HiTZ Center - Ixa, University of the Basque Country UPV/EHU
`{jaione.bengoetxea,itziar.gonzalezd,rodrigo.agerri}@ehu.eus`

Abstract

In this paper, we evaluate the capacity of current language technologies to understand Basque and Spanish language varieties. We use Natural Language Inference (NLI) as a pivot task and introduce a novel, manually-curated parallel dataset in Basque and Spanish, along with their respective variants. Our empirical analysis of crosslingual and in-context learning experiments using encoder-only and decoder-based Large Language Models (LLMs) shows a performance drop when handling linguistic variation, especially in Basque. Error analysis suggests that this decline is not due to lexical overlap, but rather to the linguistic variation itself. Further ablation experiments indicate that encoder-only models particularly struggle with Western Basque, which aligns with linguistic theory that identifies peripheral dialects (e.g., Western) as more distant from the standard. All data and code are publicly available.[1]

1 Introduction

Sociolinguistics examines language variation in relation to various regional, contextual, or social factors. During the 70s and 80s, sociolinguist William Labov highlighted the social aspect of language, and his work on rule-governed language variation has thereby legitimized non-standard language and transformed the study of sociolinguistics. For example, Labov (2006) noted that "the linguistic behavior of individuals cannot be understood without knowledge of the communities that they belong to". Thus, *variation* is an intrinsic characteristic of language, influenced by factors such as gender, age, socio-economics, or geographical location. In fact, humans make no distinction when processing their own dialect or the standard variant.

In this regard, Coseriu (1956) offered a systematic typology of language variation, based on the following three types: (i) diatopic variation, or geographical variation such as dialects, (ii) diastratic variation, or speech of different societal groups, and (iii) diaphasic variation, or speech changes depending on the communicative environment.

Basque	
Standard	Variation
Zeharo hunkituta gelditu nintzen ezusteko agur honekin	Asko emoziona nintzen ezusteko agur horregaz
Spanish	
Me quedé completamente conmovido con esta despedida inesperada	Me quedé completamente conmovío con ehta dehpedía inehperá
English	
I was completely surprised by that unexpected goodbye	

Table 1: Example from standard to variation sentences in Basque and Spanish.

In this paper, we focus on geographical variation in Basque, a low-resource language isolate with around 1 million speakers that is still undergoing a normalization process (started in 1968), and in Spanish, a higher-resourced language whose standardization process started in the 18th century with around 600 million speakers worldwide.

Recent developments in Artificial Intelligence (AI) and Natural Language Processing (NLP) have underscored the significance of social factors in language for NLP systems, as noted by Hovy and Yang (2021). This indicates the importance of developing NLP technology that not only processes standard language but also variations, as this would alleviate any potential language-based discrimination by providing more linguistically-inclusive resources.

Although previous work on NLP has primarily focused on standard language, recent research has slightly shifted its attention to the exploration of

---

[1] `https://huggingface.co/datasets/HiTZ/XNLIvar`

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 452–468
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

language variation. For instance, Zampieri et al. (2020) or Joshi et al. (2024) present thorough outlines of variation-inclusive research. However, due to the lack of data on linguistic variation, most NLP research has focused on a narrow list of languages and their variants, such as Arabic, Indic languages, or German (Joshi et al., 2024). Furthermore, other larger efforts are either based on automatically obtained data or do not provide fine-grained variation distinctions for some widely-spread languages, such as Spanish (Faisal et al., 2024; Alam et al., 2024).

Regarding Basque, the few available works have focused on historical dialects (Estarrona et al., 2020) or northern Basque dialects (Uria and Etxepare, 2012). In Spanish, all datasets with linguistic variation have been automatically collected through geolocation techniques (España-Bonet and Barrón-Cedeño, 2024; Valentini et al., 2024).

In this context, the objective of this paper is to provide the first manually curated variation dataset for Basque and Spanish that captures language variation in real-world usage. To do so, we introduce XNLIvar, the first variation-inclusive Natural Language Inference (NLI) dataset in Basque and Spanish. An example of an instance from our dataset can be found in Table 1, which will be used to evaluate current state-of-the-art language models. The main contributions are the following:

1. The first publicly available manually-curated NLI dataset for Basque and Spanish geographic language variations.
2. A comprehensive evaluation of encoder-only and decoder-based Large Language Models (LLMs) demonstrates substantially worse performance when processing language variation, particularly in Basque. Detailed error analysis shows that lexical overlap between premise and hypothesis has no impact on the performance drop, which indicates that linguistic variation could be the primary factor for this decrease in accuracy.
3. Empirical results suggest that LLM performance with Spanish variants may be attributed to the substantial representation of Spanish-language content in pre-training corpora. Further error analysis suggests that orthographic changes have a substantially negative effect on Spanish language variation processing.

To the best of our knowledge, no work has extensively addressed the automatic processing of language variation of Western and Central Basque dialects in the task of NLI.

2 Related Work

This section presents previous work on language variation in the field of NLP, with a specific focus on Basque language variation.

Language variation in NLP In recent years, there has been an increasing interest in dialects in several fields of NLP, such as dialect identification (Ramponi and Casula, 2023), sentiment analysis (Ball-Burack et al., 2021), Machine Translation (MT) (Kuparinen et al., 2023), and dialogue systems (Alshareef and Siddiqui, 2020).

Aepli and Sennrich (2022) explored cross-lingual transfer between closely related varieties by adding character-level noise to high-resource data to improve generalization. Moreover, Ramponi and Casula (2023) pretrained LLMs for geographic variations of Italian tweets. Finally, Demszky et al. (2021) showed that BERT models trained on annotated corpora obtained high accuracy for Indian English feature detection.

One of the primary limitations of these studies is the scarcity of available dialectal data. Therefore, research has largely focused on developing resources such as lexicons and dialectal datasets on a small subset of languages: Artemova and Plank (2023) propose a bilingual lexicon induction method for German dialects using LLMs, while Hassan et al. (2017) introduce a synthetic data creation method through embeddings by transforming input data into its dialectic variant. With respect to language coverage, the Arabic family, due to its relative data availability, has received the most attention, followed by Indic languages, Chinese, and German (Joshi et al., 2024).

Basque language variation In dialectology, Zuazu (2008) established an extensive and comprehensive descriptive representation of features of modern Basque dialects. In NLP, Estarrona et al. (2020) worked on a morpho-syntactically annotated corpus of Basque historical texts as an aid in the normalization process. Moreover, Uria and Etxepare (2012) introduced a corpus of syntactic variation in northern Basque dialects.

Additionally, some dialectal benchmark works have included Basque in their experimentation: both Alam et al. (2024) and Faisal et al. (2024) presented benchmarks for MT with northern Basque dialects.

Spanish language variation Several works have dealt with Spanish varieties. For instance, España-Bonet and Barrón-Cedeño (2024) automatically filtered Open Super-large Crawled Aggregated coRpus (OSCAR) by geolocation into different Spanish variants and performed a stylistic analysis. Valentini et al. (2024) automatically collected Google queries from several Spanish-speaking countries and provided an Information Retrieval baseline for Spanish varieties.

Additionally, Lopetegui et al. (2025) introduced a Cuban Spanish dataset by collecting geolocated tweets from Twitter. They focused their study on common examples, i.e., instances that can be valid across several dialects. They performed a manual annotation of tweets into *Cuban dialect*, *other dialect*, or *common example*. Similarly, Castillo-lópez et al. (2023) collected tweets from European and Latin American geolocations and annotated them for hate speech.

3 Data

In this work, we introduce a novel dataset, **XNLI-var**, that expands the XNLI framework by generating various dialectal variations for both Basque and Spanish languages. We choose NLI as an evaluation framework because it is considered to be a general benchmark for evaluating language understanding, which requires dealing with semantic relationships, logical implications, world knowledge, and contextual nuances (Williams et al., 2018; Conneau et al., 2018; Artetxe et al., 2020), including figurative language (Naik et al., 2018; Stowe et al., 2022; Liu et al., 2022; Sanchez-Bayona and Agerri, 2024). NLI is a fundamental NLP task that involves classifying the logical relationship between two segments (a premise and a hypothesis) as one of three categories: entailment (the hypothesis logically follows from the premise), contradiction (the hypothesis contradicts the premise), or neutral (the hypothesis neither follows from nor contradicts the premise). The most popular dataset is the English **MultiNLI** (Williams et al., 2018).

XNLI (Conneau et al., 2018) provides an extension of MultiNLI in 15 languages, among them Spanish (**XNLIes**). The training set for each language was created by translating the original MNLI data. However, as the test partition of MNLI is not public, Conneau et al. (2018) collected 7,500 English examples via crowdsourcing, which were then professionally translated to create the development (2500 instances) and the test (5K instances) splits of XNLI. This parallel multilingual corpus has facilitated crosslingual NLI research beyond English-centric approaches by exploring model-transfer, translate-train, and translate-test techniques to alleviate the lack of annotated training data in a given target language (Artetxe et al., 2020, 2022).

XNLIeu is a professionally translated version of the English XNLI set into Basque (Heredia et al., 2024), a language not included in the original XNLI dataset. Additionally, we also use **XNLIeu$_{\textbf{native}}$**, an NLI dataset generated by collecting native Basque premises and hiring Basque annotators to create three hypotheses per premise (Heredia et al., 2024). The experimental results from XNLIeu demonstrate that NLI systems exhibit significant performance sensitivity to disparities between training and testing data distributions, highlighting the critical role of data provenance (Artetxe et al., 2020; Volansky et al., 2013).

3.1 XNLI with Geographic Variants

To investigate the impact of language variation via evaluation in NLI, we developed two novel **XNLI variants** datasets encompassing Basque and Spanish geographic-based linguistic variations, namely, **XNLIeu$_{\textbf{var}}$** and **XNLIes$_{\textbf{var}}$**. The methodology involved a language adaptation phase to ensure the incorporation of variant diversity within the data. These two variant datasets were developed taking **XNLIeu$_{\textbf{native}}$** as a starting point for dialectal augmentation due to its authentic representation of Basque language patterns and its suitable scale for manual paraphrasing.

The adaptation process was the same for Basque and Spanish, including native speakers as linguistic informants for variant transformation. We wanted to analyze the variation that naturally occurs among native speakers, employing minimally restrictive parameters to capture authentic dialectal features. Thus, informants were instructed to perform dialectal adaptations of source sentences, with allowance for modifications across multiple linguistic dimensions, including lexical, grammatical, phonetic, and orthographic alterations. The full adaptation guidelines are detailed in Appendix A.

XNLIeu$_{\textbf{var}}$ Twelve native Basque speakers were recruited from diverse geographical regions. All participants possessed expertise in NLP and held university degrees in either Linguistics, Computer

Science, or Engineering. Each participant was tasked with reformulating approximately 20 brief sentences, with the resulting adaptations categorized according to three major dialectal variants: Western, Central, and Navarrese. To facilitate cross-dialectal comparison, a subset of 10 identical sentences was assigned to more than one annotator, enabling parallel dialectal representations. The demographic and professional characteristics of the annotators, including age, gender, and educational background, are detailed in Appendix B.

It should be noted that during data collection, a single annotator generated two types of variants for each sentence, including both dialectal variations and allocutive agreement forms in Basque. The allocutive system in Basque requires morphological marking of the addressee's gender (masculine/feminine) within the verbal form. Consequently, **XNLIeu$_{\textbf{var}}$** exhibits a higher instance count (894) compared to the original **XNLIeu$_{\textbf{native}}$** dataset (621), as shown in Table 2.

In terms of dialect distribution, 592 instances correspond to the Central dialect, usually associated with the province of Gipuzkoa, 240 instances to the Western dialect (West Gipuzkoa and Biscay), and just 63 instances to the Navarrese dialect, comprising 7% of the data. Thus, the Navarrese dialect is clearly under-represented in our data.

XNLIes$_{\textbf{var}}$ XNLIeu$_{\text{native}}$ was automatically translated into Spanish using Claude 3.5 Sonnet[2], generating the **XNLIeu2es$_{\textbf{native}}$** dataset and facilitating the creation of a parallel corpus for Basque and Spanish texts with their respective variants. Quality verification was conducted through manual review of the machine-generated translations, making sure that they constituted an authentic representation of Spanish language patterns. Finally, the translated corpus was provided to Spanish-language annotators for variant-specific adaptation.

The adaptation task involved six independent annotators, each assigned a set of 50 sentences for dialectal adaptation into their respective Spanish variants. They represented four distinct geographical locations: Cuba, Ecuador, Spain, and Uruguay. Two annotators from Spain performed adaptations into separate dialectal variants (Andalusian and Tenerife), resulting in a total of five Spanish dialectal variations in the final dataset. The demographic and professional characteristics of the annotators, including age, gender, and educational background, are documented in Appendix B.

Train	
Dataset	Instances
MNLI	392k
MNLIeu	392k
MNLIes	392k
Test	
XNLIeu	5010
XNLIes	5010
XNLIeu$_{\text{native}}$	621
XNLIeu2es$_{\text{native}}$	621
XNLIeu$_{\text{var}}$	894
XNLIes$_{\text{var}}$	666

Table 2: Datasets used for training and testing.

It is worth noting that some annotators found it difficult to add dialectal features to the standard sentences. This could be due to the high number of common examples in Spanish varieties (Lopetegui et al., 2025; Zampieri et al., 2024). In other words, the distinctions between Spanish varieties tend to be more homogeneous and thus contain less variation compared to Basque (Section 6).

Similar to the Basque adaptation, multiple dialectal variants were documented by some annotators. These variants exhibited phonological phenomena such as word-final /s/ deletion (e.g., *digamos* → *digamo*) and /s/ to /j/ substitution in word-final position (resulting in *digamoj*). Thus, XNLIes$_{\text{var}}$ contains 666 examples, representing a marginally higher count than the base dataset.

Table 2 provides an overview of the datasets used for experimentation, including our newly generated **XNLIvar**, consisting of XNLIeu$_{\text{var}}$ and XNLIes$_{\text{var}}$.

4 Experimental settings

Empirical research was based on the aforementioned datasets to evaluate the impact of dialectal variation on NLI performance.

Discriminative experiments Table 3 illustrates the experiments performed using encoder-only Transformer models and the datasets specified in Table 2.

- **Model transfer:** The train split of the original MNLI (English) is used to fine-tune multilingual encoder models. Evaluation is performed on the test sets for Basque and Spanish specified in Table 2.

---

[2] `https://www.anthropic.com/news/claude-3-5-sonnet`

- **Translate-train:** The MNLI training is automatically translated into Basque and Spanish (MNLI_{eu} and MNLI_{es}); multilingual and monolingual encoders are then fine-tuned using the translated training data and evaluated in each of the target languages.
- **Translate-test:** Tests in the target languages are translated into English and evaluated using the MNLI fine-tuned encoders (in English).

Configuration	**Train**	**Test**
Model transfer	English	Target language
Translate-train	Target language	Target language
Translate-test	English	Target → English

Table 3: Discriminative model configurations and data. **→:** Translated to.

Summarizing, training is always done with MNLI, either in its original English form or using the automatically translated versions to Basque and Spanish. Moreover, there are three different test data types: (i) XNLI test data professionally translated into the target languages (XNLIeu, XNLIes) (ii) the manually created native Basque data and its translation to Spanish ($\text{XNLIeu}_{\text{native}}$, $\text{XNLIeu2es}_{\text{native}}$) and, (iii) the native datasets adapted to different variations for each of the target languages ($\text{XNLIeu}_{\text{var}}$, $\text{XNLIes}_{\text{var}}$).

We employed two multilingual encoder-only language models for our target languages: XLM-RoBERTa large (Conneau et al., 2020) and mDeBERTa (He et al., 2021). The hyperparameter configuration followed Heredia et al. (2024), implementing differential learning rates of 5e-5 and 10e-6 for BERT and RoBERTa architectures, respectively. All other parameters were maintained at their default values. The training process consisted of 10 epochs across all model configurations.

Generative experiments We experimented with generative LLMs to evaluate the decoders' ability to perform NLI when language variation is present. We started with a zero-shot setting, where we prompt LLMs to identify the NLI relation.

We also evaluated alternative prompting methodologies, specifically, few-shot and Chain of Thought (CoT) approaches. The few-shot prompt implemented a single example for each classification category. The CoT methodology incorporated detailed task-specific contextual information alongside a single example for each label.

To further evaluate the linguistic comprehension capabilities of LLMs with respect to Basque and Spanish variants, we implemented an alternative methodological approach by transforming the NLI task into a Question-Answering (QA) setting. In this experimental configuration, the input prompt was restructured as a question to be answered by the LLM, with the three possible answers based on the NLI inference labels. Zero-shot and few-shot prompting strategies kept the same. The complete set of prompt templates used across all task formulations is available in Appendix C.

We selected Llama-3.1-Instruct (8B and 70B versions) (Dubey et al., 2024) and Gemma 2 instruct (9B and 27B versions) (Mesnard et al., 2024) due to their strong performance in both Basque and Spanish languages[3] (Etxaniz et al., 2024; Figueras et al., 2025). In the next section we focus on the results obtained by the larger LLMs (performances with smaller LLMs in Appendix E).

5 Results

We first report the results obtained in the discriminative settings, while in Section 5.2, we discuss the results of in-context learning with LLMs.

5.1 Discriminative Experiments

By looking at the results reported in Table 4, the empirical results demonstrate a significant performance degradation when comparing XNLIeu and XNLIes against the native and variation datasets. This observation aligns with existing literature documenting the adverse effects of train-test distribution shifts in cross-lingual settings (Artetxe et al., 2020; Volansky et al., 2013). When comparing native and variation data results, where the only difference is the presence of dialectal data, we see a decrease in results. Therefore, results show that language models perform worse when variants are included in the NLI task.

By doing a cross-configuration analysis, we see that for Basque, the best results are obtained with XLM-RoBERTa in the translate-train for XNLIeu (83.42) and $\text{XNLIeu}_{\text{var}}$ (73.21), while for $\text{XNLIeu}_{\text{native}}$ (75.85), the train-test is superior. Overall, the empirical results demonstrate that the translate-train approach with XLM-RoBERTa yielded the best overall performance for Spanish and Basque. This suggests that training and evalu-

---

[3] `https://hf.co/spaces/la-leaderboard/la-leaderboard`

	Basque								
	Model transfer			Translate-train			Translate-test		
	XNLIeu	XNLIeu$_{\text{native}}$	XNLIeu$_{\text{var}}$	XNLIeu	XNLIeu$_{\text{native}}$	XNLIeu$_{\text{var}}$	XNLIeu	XNLIeu$_{\text{native}}$	XNLIeu$_{\text{var}}$
XLM-RoBERTa large	80.00	72.09	68.24	**83.42**	75.63	**73.21**	-	**75.85**	71.63
mDeBERTa	78.95	70.21	67.26	81.42	72.14	69.77	-	72.68	70.28
	Spanish								
	XNLIes	XNLIeu2es$_{\text{native}}$	XNLIes$_{\text{var}}$	XNLIes	XNLIeu2es$_{\text{native}}$	XNLIes$_{\text{var}}$	XNLIes	XNLIeu2es$_{\text{native}}$	XNLIes$_{\text{var}}$
XLM-RoBERTa large	83.05	74.02	73.07	**84.69**	**74.61**	**73.72**	-	73.86	71.77
mDeBERTa	82.02	74.13	71.57	83.27	72.25	70.77	-	72.30	69.89

Table 4: Accuracy results for Basque and Spanish discriminative experiments.

	Basque									
	Llama-3.1-Instruct-70B					**Gemma-2-it-27B**				
	nli-zero	nli-few	qa-zero	qa-few	chain	nli-zero	nli-few	qa-zero	qa-few	chain
XNLIeu	33.65	53.17	33.31	54.89	**55.25**	61.10	62.81	61.84	**65.27**	58.28
XNLIeu$_{\text{native}}$	38.81	56.68	39.61	58.61	**60.71**	64.90	66.67	65.70	**68.28**	66.99
XNLIeu$_{\text{var}}$	33.78	48.66	31.54	**50.11**	49.22	57.61	60.96	57.49	**61.52**	58.05
	Spanish									
	Llama-3.1-Instruct-70B					**Gemma-2-it-27B**				
	nli-zero	nli-few	qa-zero	qa-few	chain	nli-zero	nli-few	qa-zero	qa-few	chain
XNLIes	54.65	62.18	51.54	65.69	**73.97**	66.75	71.28	70.52	**73.05**	68.88
XNLIeu2es$_{\text{native}}$	62.96	62.48	62.16	70.69	**77.29**	71.50	72.62	73.91	73.43	**76.97**
XNLIes$_{\text{var}}$	59.42	62.32	54.27	69.24	**75.52**	70.37	72.30	72.79	72.14	**74.56**

Table 5: Results with generative LLMs.

ating in the target language constitutes the optimal method, irrespective of whether the data includes standard or variation-inclusive linguistic content.

Regarding *native* and *variant* results, analysis reveals that Spanish consistently outperforms Basque across all settings and evaluation datasets, demonstrating greater resilience to linguistic variation. In fact, while Spanish accuracy drops minimally (less than 1 percentage point in most cases), Basque performance suffers a higher decrease, with model-transfer and translate-test approaches showing an approximately 4-point drop and translate-train a 2.5-point drop. This highlights a sharper impact of variation on Basque performance.

These results show that when English is the source training data, model-transfer provides competitive results for a high-resource, structurally similar language such as Spanish, while for a low-resource and morphologically different language such as Basque, the data-transfer (translate-train) strategy remains preferable (Agerri et al., 2020; Artetxe et al., 2020; García-Ferrero et al., 2022).

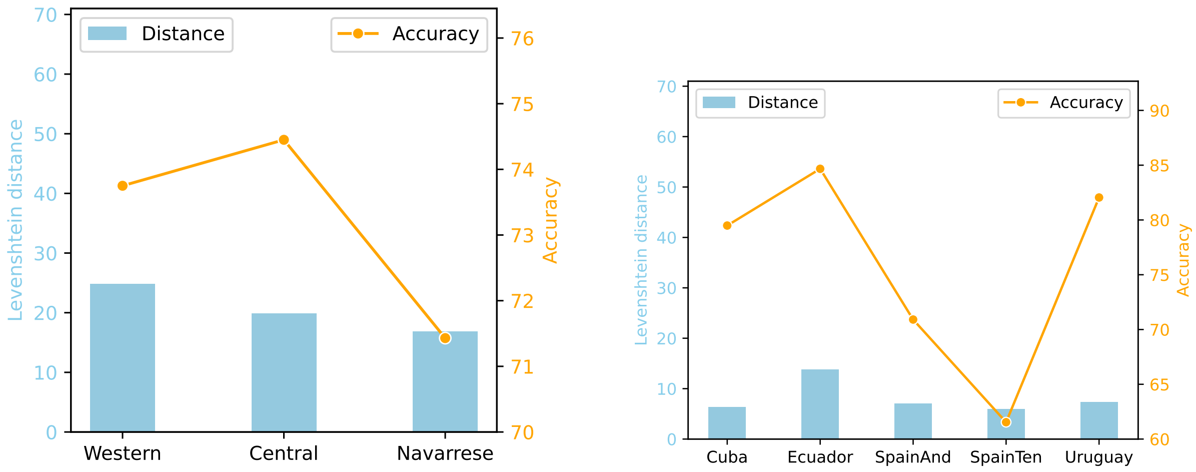
Overall, the consistently lower performance observed in Basque relative to Spanish across all evaluation conditions can be attributed to three key factors: (i) Basque's agglutinative morphological structure, (ii) its classification as a language isolate, and (ii) reduced Basque language representation in the models' pre-training data (Agerri et al., 2020; Etxaniz et al., 2024).

Finally, we also experimented with two Basque monolingual models, RoBERTa-Euscrawl (Artetxe et al., 2022) and BERTeus (Agerri et al., 2020), in the translate-train setting. However, while competitive, their results did not outperform those obtained by XLM-RoBERTa large. Further details can be found in Appendix D.

5.2 Generative Experiments

Table 5 presents the evaluation results for LLMs in the task on variation-inclusive NLI with the largest LLMs tested, namely, Llama-3.1-Instruct-70B and Gemma-2-it-27B.

A first observation reveals a significant performance degradation across all evaluated LLMs when transitioning from standard datasets (XNLIeu$_{\text{native}}$ and XNLIeu2es$_{\text{native}}$) to their variant counterparts (XNLIeu$_{\text{var}}$ and XNLIes$_{\text{var}}$). This suggests a substantial limitation in the capacity of LLMs to process and comprehend linguistic variations within the task. The results also indicate that including examples in the prompt engineering pro-



(a) Basque results

(b) Spanish results

Figure 1: Standard to dialectal Levenshtein distance vs accuracy of best discriminative models.

cess yields positive effects (in the qa-few and CoT methodologies). Notably, for Spanish, the CoT approach demonstrates superior performance compared to XLM-RoBERTa large on XNLIeu2es$_{\text{var}}$ and XNLIes$_{\text{var}}$ datasets.

Concerning Basque, the experimental results demonstrate that Gemma-2 exhibits better performance compared to Llama-3.1. Moreover, for XNLIeu$_{\text{var}}$ Gemma's optimal performance (61.52) experiences a reduction of 6.5 percentage points relative to the standard XNLIeu$_{\text{native}}$ (68.28). In contrast, Llama-3.1 exhibits a more substantial decline of 10 percentage points in XNLIeu$_{\text{var}}$ performance. These findings indicate that Gemma maintains greater robustness against linguistic variation compared to Llama-3.1.

For Spanish, CoT prompting generally yields the highest accuracy. The variation-inclusive evaluation dataset (XNLIes$_{\text{var}}$) produces results very close to those of XNLIes$_{\text{native}}$, with Llama 3.1 achieving 75.77 and 77.29, and Gemma 2 reaching 74.56 and 76.56, respectively. Despite this closeness, linguistic variation still causes a drop in accuracy. Overall, Llama 3.1 performs slightly better than Gemma 2, though the difference is minimal.

The empirical evidence obtained from these analyses of Basque and Spanish language understanding indicates that LLMs exhibit significant limitations in their capacity to comprehend linguistic content when confronted with dialectal and geographical variations.

6 Error analysis

This section presents a quantitative error analysis to evaluate XLM-RoBERTa large's performance with respect to variation-inclusive evaluation data.

Dialect to standard distance The Levenshtein distance metric, which quantifies the minimum number of single-character operations (insertions, deletions, or substitutions) necessary for string transformation, was computed between dialectal and standard sentences. The analysis of distance results demonstrates that Basque dialectal variants (Figure 1a) exhibit significantly greater divergence from the standard form compared to Spanish variants (Figure 1b), which display higher proximity to their standardized counterpart. The observed inter-dialectal variation patterns suggest a more pronounced linguistic differentiation within Basque dialectal systems relative to Spanish dialectal varieties. This emphasizes the difference in variation between languages and highlights the importance of language-specific analysis in the field of language variation processing in NLP.

Accuracy per dialect We analyzed the accuracy results for each individual dialect class, in order to see if some dialects are more difficult to process than others. The relation between the accuracy for each dialect and the distance from standard to dialect is illustrated in Figure 1.

In the case of Basque (Figure 1a), we see that, in terms of string distance, the Western dialect is the one that is the most different from the standard, followed by the Central and Navarrese dialects. However, the lowest accuracy is accounted for in the Navarrese dialect, which is the dialect label that seems to be closest to the standard form of language. This could be because of its underrepresentation in our dataset, as Navarrese examples comprise only 7% of our data (Appendix B). When focusing on Western and Central dialects, it can be observed that, as the distance from standard to dialectal gets higher, accuracy gets lower, suggesting that dialects further from the standard (in our case, Western) are harder to process. The Central dialect being closer to the standard is expected, as it served as the main foundation for the current standard form of Basque.

In fact, according to research in Basque dialectology, peripheral dialects have been found to be more distant from the rest (Mitxelena, 1981). This fact has also been corroborated by NLP studies analyzing Basque historical dialects, where Biscayan (Western) and Souletin display the greatest difference (Estarrona et al., 2023). Additionally, research has documented the Bizkaian dialect's historical tendency toward linguistic divergence, both from other Basque dialects and from its own earlier forms (Zuazu, 2015).

In Spanish variants, Figure 1b shows Ecuador and Uruguay displaying the highest distance values and accuracy scores. Further analysis has shown that adaptations into these two variants mostly include replacing lexical words with alternatives that are more commonly used in those varieties (e.g., *construccion futura* > *nuevos edificios*), as well as grammatical structures typical of those dialects (e.g., *he podido* > *pude*). However, standard orthography has been preserved throughout.

In turn, adaptations into Cuban, SpainAnd and SpainTen variants mostly include phonological or orthography changes (e.g. *misma* > *mihma, fuerza* > *fuersa*), which have resulted in lower distance to the standard form of Spanish, but a decrease in accuracy compared to the variants written in standard orthography (Ecuador and Uruguay). This reveals a correlation between standard orthography and high accuracy, and highlights the difficulties of discriminative models to deal with data which includes non-standard orthography. This analysis is illustrated in Appendix G. These results match those observed in earlier studies, where orthography variations have also been found to be problematic (De la Rosa et al., 2024). Additional results of per-dialect accuracy results are presented in Appendix F.

Dataset	Instances	Discrimin.	Genera.
Basque			
XNLIeu$_{\text{native}}$	621	75.63	68.28
XNLIeu$_{\text{var}}$	894	73.21	61.52
Less-western	834	73.14	60.79
Less-central	834	72.70	61.03
No repetitions	621	71.77	60.39
Spanish			
XNLIeu2es$_{\text{native}}$	621	74.61	77.29
XNLIes$_{\text{var}}$	666	73.72	75.52
No repetitions	621	73.00	77.13

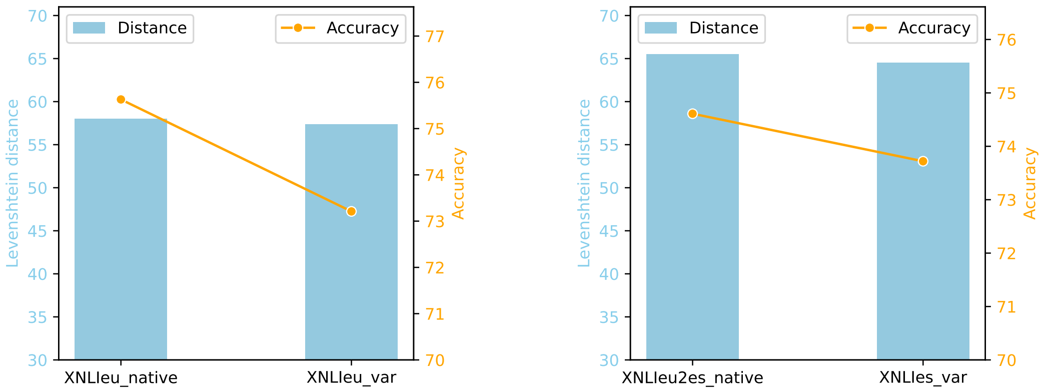
Table 6: Ablation experiments on Basque and Spanish variation data (XNLIeu$_{\text{var}}$ and XNLIes$_{\text{var}}$, respectively). Results obtained using the best discriminative setting (Translate-train XLM-RoBERTa large in Table 4) as well as best generative results for Basque (Gemma-2 qa-few) and Spanish (Llama-3.1 chain) in Table 5.

Ablation Tests As explained in Section 3.1 and illustrated in Table 2, test data in XNLIeu$_{\text{var}}$ and XNLIes$_{\text{var}}$ contains duplicated instances in different dialects. In order to see the effect that different types of variation have on accuracy, we have performed some ablation experiments.

Four different Basque speakers (two Western and two Central) adapted the same 10 sentences, providing us with four distinct versions of those 10 sentences. We used these instances to create two new versions of the dataset, one by removing the repeated sentences from Western variants (*Less-western*), and another one without the repeated instances from the Central one (*Less-central*). Table 6 presents accuracy results with these datasets.

The results show that accuracy is higher when Western-dialect instances are removed (73.14) than when Central instances are excluded (72.70).

Additionally, we removed all duplicated variant instances from XNLIvar$_{\text{eu}}$, resulting in a completely parallel variation dataset to XNLIeu$_{\text{native}}$ (*No repetitions*), which allows us to calculate whether the results between the standard and the variant versions are statistically significant. As reported in Table 6, accuracy between *No repetitions* and the *standard* substantially decreases (71.77 vs 75.63) for the Basque discriminative experiments. According to a chi-square test of independence, this difference is highly statistically significant (p < .001, df=1). Similar to Basque, all the repeated variant instances from the Spanish



(a) Premise-Hypothesis distance and accuracy for Basque

(b) Premise-Hypothesis distance and accuracy for Spanish

Figure 2: Levenshtein distance from premise to hypothesis and accuracy of discriminative models.

variation dataset were removed, obtaining a parallel dataset to XNLIes$_{\text{native}}$. Using the *No repetitions* split, a chi-square test of independence establishes that differences with results on XNLIeu2es$_{\text{native}}$ are highly statistically significant ($p < .001$, df=1).

Generative LLMs follow the same trend, with differences in performance being highly statistically significant in both languages ($p < .001$, df=1).

Premise and hypothesis lexical overlap To investigate the potential correlation between lexical overlap and accuracy, we measured the Levenshtein distance between premises and hypotheses. The analysis of the data in Figure 2 indicates that lexical overlap remains consistent across standard and dialectal varieties, while a substantial decrease in accuracy was observed in both Basque and Spanish datasets. These findings suggest that while lexical overlap appears to have minimal impact on accuracy metrics, linguistic variation emerges as the significant factor affecting performance. Therefore, the observed pattern implies that dialectal variations, rather than lexical similarities, may be the primary factor of accuracy degradation in this context.

In fact, Figure 2a demonstrates a more pronounced decrease in accuracy for Basque compared to Spanish, underscoring both the critical need to improve Basque representation in multilingual discriminative models and the necessity for additional investigation into language variation processing.

7 Concluding Remarks

This paper presents a novel dataset that includes geographical variants of Basque and Spanish. The dataset represents the first documented instance of a manually-curated, variation-inclusive corpus for these languages, facilitating research and evaluation on linguistic variants via NLI. Additional speaker metadata expands its value as a resource for sociolinguistic research on generational and geographical differences in Basque and Spanish. Our investigation involved the empirical evaluation of both discriminative and generative language models across various NLI task configurations.

Results indicate that language models' performance drops when linguistic variation is present. This performance degradation is particularly pronounced in Basque variants, where linguistic variation is higher compared to Spanish variants. Furthermore, the performance drop intensifies proportionally with the linguistic distance between dialectal variants and their respective standardized forms for Basque, with a higher impact in the Western dialect. This coincides with previously established linguistic theory, which states that some Basque dialects (such as Western) have a historical tendency to distance themselves from the standard. In the case of Spanish, variants with non-standard orthography have shown a significant accuracy drop. Finally, the lexical overlap between premises and hypotheses appears to have minimal impact, suggesting that lower performance is due to linguistic variation.

Future work will involve expanding the dataset to include additional geographical variants of both Basque and Spanish, as well as incorporating other languages. Investigation of variation-inclusive monolingual models represents a promising avenue for future research.

Limitations

In this paper, we have focused on geographic variants of language due to their low representation in NLP. We conducted our experiments for a lesser-resourced language, Basque, and a higher-resourced language, Spanish. However, we have only represented some of the variations of these languages, and our variation datasets have been created by 12 speakers for Basque and 6 speakers for Spanish. We tried to include the most representative dialects with different kinds of speakers, but we are aware that all the speakers have linguistic and NLP backgrounds, and laypeople could contribute differently.

Our empirical findings demonstrate decreased accuracy in natural language inference tasks within our variants dataset. However, generalization of these results requires expansion to include additional linguistic variants and evaluation across a broader range of NLP tasks.

To augment the dataset, we are recruiting speakers from diverse linguistic backgrounds to contribute additional variation data. We further intend to evaluate the performance of NLP tools and LLMs on tasks incorporating dialectal and register variation.

Acknowledgments

This work has been supported by the HiTZ center and the Basque Government (Research group funding IT-1805-22). Jaione Bengoetxea is funded by the Basque Government pre-doctoral grant (PRE_2024_1_0028). We also acknowledge the following MCIN/AEI/10.13039/501100011033 projects: (i) DeepKnowledge (PID2021-127777OB-C21) and by FEDER, EU; (ii) DeepMinor (CNS2023-144375) and European Union NextGenerationEU/PRTR.

We acknowledge the contributions of the following people to generate the Basque and Spanish linguistic variants: Ainara Estarrona, Alex Muñoz Alvarado, Ekhi Azurmendi, Irune Zubiaga, Itziar Aldabe, Izaskun Aldezabal, Jon Alkorta, Jose Maria Arriola, Leonel Cosme, María Teresa Martín-Valdibia, Marta Torres Martínez, Mikel Zubillaga, Naiara Perez, Nancy Cristina Álamo Suárez, Oier Ijurco, Yanisbel Ríos Laborde and Yoandra Chuen Gómez.

References

Noëmi Aepli and Rico Sennrich. 2022. Improving zero-shot cross-lingual transfer between closely related languages by injecting character-level noise. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4074–4083, Dublin, Ireland. Association for Computational Linguistics.

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for Basque. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4781–4788, Marseille, France. European Language Resources Association.

Md Mahfuz Ibn Alam, Sina Ahmadi, and Antonios Anastasopoulos. 2024. CODET: A benchmark for contrastive dialectal evaluation of machine translation. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1790–1859, St. Julian's, Malta. Association for Computational Linguistics.

Tahani Alshareef and Muazzam Ahmed Siddiqui. 2020. A seq2seq neural network based conversational agent for gulf arabic dialect. In *2020 21st International Arab Conference on Information Technology (ACIT)*, pages 1–7.

Ekaterina Artemova and Barbara Plank. 2023. Low-resource bilingual dialect lexicon induction with large language models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 371–385, Tórshavn, Faroe Islands. University of Tartu Library.

Mikel Artetxe, Itziar Aldabe, Rodrigo Agerri, Olatz Perez-de Viñaspre, and Aitor Soroa. 2022. Does corpus quality really matter for low-resource languages? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7383–7390, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Ari Ball-Burack, Michelle Seng Ah Lee, Jennifer Cobbe, and Jatinder Singh. 2021. Differential tweetment: Mitigating racial dialect bias in harmful tweet detection. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 116–128, New York, NY, USA. Association for Computing Machinery.

Galo Castillo-lópez, Arij Riabi, and Djamé Seddah. 2023. Analyzing zero-shot transfer scenarios across Spanish variants for hate speech detection. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 1–13, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Eugenio Coseriu. 1956. *La geografía lingüística*, volume 11. Universidad de la República, Facultad de Humanidades y Ciencias.

Javier De la Rosa, Álvaro Cuéllar, and Jörg Lehmann. 2024. The modernifa project: orthographic modernization of spanish golden age dramas with language models. *Anuario Lope de Vega Texto literatura cultura*, 30:410–425.

Dorottya Demszky, Devyani Sharma, Jonathan Clark, Vinodkumar Prabhakaran, and Jacob Eisenstein. 2021. Learning to recognize dialect features. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2315–2338, Online. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony S. Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, (...), and Zhiwei Zhao. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Cristina España-Bonet and Alberto Barrón-Cedeño. 2024. Elote, choclo and mazorca: on the varieties of Spanish. 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 3689–3711, Mexico City, Mexico. Association for Computational Linguistics.

Ainara Estarrona, Izaskun Etxeberria, Ricardo Etxepare, Manuel Padilla-Moyano, and Ander Soraluze. 2020. Dealing with dialectal variation in the construction of the Basque historical corpus. In *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 79–89, Barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL).

Ainara Estarrona, Izaskun Etxeberria, Manuel Padilla-Moyano, and Ander Soraluze. 2023. Measuring language distance for historical texts in basque. *Procesamiento del Lenguaje Natural*, 70:53–61.

Julen Etxaniz, Oscar Sainz, Naiara Miguel, Itziar Aldabe, German Rigau, Eneko Agirre, Aitor Ormazabal, Mikel Artetxe, and Aitor Soroa. 2024. Latxa: An open language model and evaluation suite for Basque. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14952–14972.

Fahin Faisal, Orevaoghene Ahia, Aarohi Srivastava, Kabir Ahuja, David Chiang, Yulia Tsvetkov, and Antonios Anastasopoulos. 2024. Dialectbench: A nlp benchmark for dialects, varieties, and closely-related languages. *ArXiv*, abs/2403.11009.

Blanca Calvo Figueras, Eneko Sagarzazu, Julen Etxaniz, Jeremy Barnes, Pablo Gamallo, Iria De Dios Flores, and Rodrigo Agerri. 2025. Truth knows no language: Evaluating truthfulness beyond english. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Iker García-Ferrero, Rodrigo Agerri, and German Rigau. 2022. Model and data transfer for cross-lingual sequence labelling in zero-resource settings. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6403–6416.

Hany Hassan, Mostafa Elaraby, and Ahmed Y. Tawfik. 2017. Synthetic data for neural machine translation of spoken-dialects. In *Proceedings of the 14th International Conference on Spoken Language Translation*, pages 82–89, Tokyo, Japan. International Workshop on Spoken Language Translation.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *CoRR*, abs/2111.09543.

Maite Heredia, Julen Etxaniz, Muitze Zulaika, Xabier Saralegi, Jeremy Barnes, and Aitor Soroa. 2024. XNLIeu: a dataset for cross-lingual NLI in Basque. 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 4177–4188, Mexico City, Mexico. Association for Computational Linguistics.

Dirk Hovy and Diyi Yang. 2021. The importance of modeling social factors of language: Theory and practice. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 588–602, Online. Association for Computational Linguistics.

Aditya Joshi, Raj Dabre, Diptesh Kanojia, Zhuang Li, Haolan Zhan, Gholamreza Haffari, and Doris Dippold. 2024. Natural language processing for dialects of a language: A survey. *Preprint*, arXiv:2401.05632.

Olli Kuparinen, Aleksandra Miletić, and Yves Scherrer. 2023. Dialect-to-standard normalization: A large-scale multilingual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13814–13828, Singapore. Association for Computational Linguistics.

William Labov. 2006. *The Social Stratification of English in New York City*, 2 edition. Cambridge University Press.

Emmy Liu, Chenxuan Cui, Kenneth Zheng, and Graham Neubig. 2022. Testing the ability of language models to interpret figurative language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4437–4452.

Javier A. Lopetegui, Arij Riabi, and Djamé Seddah. 2025. Common ground, diverse roots: The difficulty of classifying common examples in Spanish varieties. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 168–181, Abu Dhabi, UAE. Association for Computational Linguistics.

Gemma Team Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, L. Sifre, Morgane Rivière, Mihir Kale, J Christopher Love, Pouya Dehghani Tafti, L'eonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Am'elie H'eliou, Andrea Tacchetti, Anna Bulanova, (...), and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *ArXiv*, abs/2403.08295.

Luis Mitxelena. 1981. Lengua común y dialectos vascos. *Anuario del Seminario de Filología Vasca" Julio de Urquijo"*, 15:289–313.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.

Alan Ramponi and Camilla Casula. 2023. DiatopIt: A corpus of social media posts for the study of diatopic language variation in Italy. In *Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023)*, pages 187–199, Dubrovnik, Croatia. Association for Computational Linguistics.

Elisa Sanchez-Bayona and Rodrigo Agerri. 2024. Meta4xnli: A crosslingual parallel corpus for metaphor detection and interpretation. *arXiv*, 2404.07053.

Kevin Stowe, Prasetya Utama, and Iryna Gurevych. 2022. IMPLI: Investigating NLI models' performance on figurative language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5375–5388.

Larraitz Uria and Ricardo Etxepare. 2012. Hizkeren arteko aldakortasun sintaktikoa aztertzeko metodologiaren nondik norakoak: Basyque aplikazioa. *Lapurdum. Euskal ikerketen aldizkaria| Revue d'études basques| Revista de estudios vascos| Basque studies review*, (16):117–135.

Francisco Valentini, Viviana Cotik, Damián Ariel Furman, Ivan Bercovich, Edgar Altszyler, and Juan Manuel P'erez. 2024. Messirve: A large-scale spanish information retrieval dataset. *ArXiv*, abs/2409.05994.

Vered Volansky, Noam Ordan, and Shuly Wintner. 2013. On the features of translationese. *Digital Scholarship in the Humanities*, 30(1):98–118.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612.

Marcos Zampieri, Kai North, Tommi Jauhiainen, Mariano Felice, Neha Kumari, Nishant Nair, and Yash Mahesh Bangera. 2024. Language variety identification with true labels. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10100–10109, Torino, Italia. ELRA and ICCL.

Koldo Zuazu. 2008. *Euskalkiak. Euskararen dialektoak*. Elkar.

Koldo Zuazu. 2015. The unification of the basque language. `http://euskalkiak.eus/en/euskararen_batasuna.php`. Accessed: 09-06-2025.

A Guidelines for Variation Adaptation

Language	Adaptation guidelines
Basque	Ataza honetan testu motz batzuk hizkuntza formal/estandarretik hizkuntza informalagora/euskalkietara berridatzi behar dira. Bakoitzak bere hizkuntza informal/dialektalean esango lukeen bezala idaztea da helburua. Hau horrela, ondorengo aldaketak proposatzen ditugu: • Esamolde edo hizkuntza informalagoa bilakatu. • Ezaugarri dialektalak gehitu, bai lexiko aldetik eta bai gramatika edo fonetika aldetik. • Hika. Erregistroa edo dialektoak barne hartzen dituen beste edozein aldaketa ongietorria da. Adibidez: **Jatorrizkoa:** Bi dantzari horiek dantza hunkigarria eskaini zuten herriko frontoian. **Berridatzia:** Bi dantsari hoiek dantza emozionantia eskeiñi zuten herriko frontoien.
Spanish	En esta tarea se deben reescribir algunos textos cortos del lenguaje formal/estándar a un lenguaje más informal/dialectal. El objetivo es adaptar las frases como cada persona lo diría en su propio lenguaje dialectal. De esta manera, se propone hacer los siguientes cambios: • A nivel de registro: Más informal, reescribiéndola de manera más coloquial • Con rasgos dialectales, sean léxicos, gramaticales o fonéticos • Adaptar la ortografía para que refleje vuestra pronunciación, dialecto Cualquier otro cambio que refleje un cambio de registro o dialecto es bienvenido. Por ejemplo: **Frase original:** El amigo se quedó sin opciones cuando le dijeron que el autobús no pasaría más. **Frase adaptada:** El socio se quedo botao cuando le dijeron que la guagua no pasaba ma.
English	This task involves rewriting short texts from formal/standard language to a more informal/dialectal language. The objective is to rewrite the sentences as each person would say them in their own dialectal language. The following changes are proposed: • At the register level: More informal, rewriting in a more colloquial manner • With dialectal features, whether lexical, grammatical, or phonetic • Adapting spelling to reflect your pronunciation, dialect Any other changes that reflect a change in register or dialect are welcome. For example: **Original phrase:** Everyone, hurry up now, dinner is about to get cold. **Adapted phrase:** Y’all better hurry up now, supper’s fixin’ to get cold.

Table 7: Guidelines for standard to dialectal adaptations, both in Basque and Spanish, and an English translation

B Adaptation Process Information

B.1 Annotator Metadata

Variable	Category	N	%
Location	Gipuzkoa	7	58.34
	Biscay	4	33.34
	Navarre	1	8.34
Age	20-30	5	41.67
	30-40	3	25.00
	40+	4	33.34
Gender	Male	5	41.67
	Female	7	58.34
Background	Linguist	8	66.67
	Non-linguist	4	33.34

(a) Demographic metadata of annotators. **N** = Count; % = Percentage

Variable	Category	N	%
Location	Cuba	2	33.33
	Ecuador	1	16.67
	SpainAndalusia	1	16.67
	SpainTenerife	1	16.67
	Uruguay	1	16.67
Age	20-30	1	16.67
	30-40	3	50.00
	40+	2	33.33
Gender	Male	2	33.34
	Female	4	66.67
Background	Linguist	3	50.00
	Non-linguist	3	50.00

(b) Demographic metadata of annotators. **N** = Count; % = Percentage

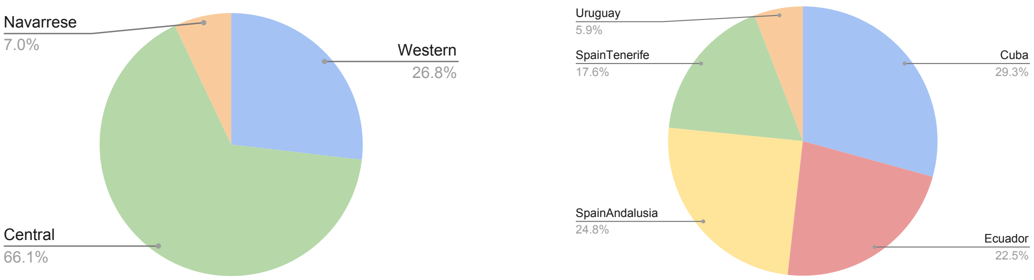
Table 8: Annotator metadata

B.2 Adaptation Type

	Basque		**Spanish**	
Change type	**N**	**%**	**N**	**%**
re-write	18	6.04	61	27.48
dialectal	223	74.83	161	72.52
Allocutive_masc	37	12.41	-	-
allocutive_fem	20	6.71	-	-
Total	298		222	

Table 9: Number and percentage of change types in Basque and Spanish data. **N**: Count of examples; **%**: Percentage

B.3 Geographical Variants Distribution in Data



(a) Number of examples per geographical variants in Basque

(b) Number of examples per geographical variants in Spanish

Figure 3: Geographical variant label representation in XNLIvar

C Prompts

Task formulation	Prompt
nli-zero	Please, answer in one word, with one of the following labels: <entailment>, <contradiction> or <neutral> Use exactly one of these three labels.
nli-few	"Say which is the inference relationship between these two sentences. Please, answer in one word, with one of the following labels: <entailment>, <contradiction> or <neutral> Use exactly one of these three labels. Here you have some examples: Postal Service were to reduce delivery frequency -> The postal service could deliver less frequently: <entailment>. This elegant spa town on the edge of the Lac du Bourget has offered cures for rheumatism and other ailments for centuries -> The town was only established in the past fifty years: <contradiction>. And while we allow people to give a kidney to their child, we do not allow them to donate their heart -> You can't always donate organs to your child: <neutral>.
qa-zero	Are these two sentences entailed, contradicted or undetermined to each other? Please, answer in one word, with one of the following labels: <entailment>, <contradiction> or <neutral> Use exactly one of these three labels.
qa-few	Are these two sentences entailed, contradicted or undetermined to each other? Please, answer in one word, with one of the following labels: <entailment>, <contradiction> or <neutral> Use exactly one of these three labels. Here you have some examples: Postal Service were to reduce delivery frequency -> The postal service could deliver less frequently: <entailment>. This elegant spa town on the edge of the Lac du Bourget has offered cures for rheumatism and other ailments for centuries -> The town was only established in the past fifty years: <contradiction>. And while we allow people to give a kidney to their child , we do not allow them to donate their heart -> You can't always donate organs to your child: <neutral>.
chain	You are an expert linguist and your task is to annotate sentences for the task of Natural Language Inference. This task consists in determining if a first sentence (premise) entails, contradicts or does not entail nor contradict the second sentence (hypothesis). Please, answer in one word, with one of the following labels: <entailment>, <contradiction> or <neutral> \n Use exactly one of these three labels \n Here you have a few examples:\n Premise: Postal Service were to reduce delivery frequency. \n Hypothesis: The postal service could deliver less frequently. \n Answer: <entailment> \n Premise: This elegant spa town on the edge of the Lac du Bourget has offered cures for rheumatism and other ailments for centuries. \n Hypothesis: The town was only established in the past fifty years. \n Answer: <contradiction> \n Premise: And while we allow people to give a kidney to their child , we do not allow them to donate their heart. \n Hypothesis: You can't always donate organs to your child. \n Answer: <neutral>

Table 10: Different task formulation prompts for generative model prompting

D Basque Monolingual Discriminative Results

	Translate-train		
	XNLIeu	XNLIeu$_{\text{native}}$	XNLIeu$_{\text{var}}$
RoBERTa-Euscrawl	82.63	73.43	72.24
BERTeus	78.15	68.81	63.67

Table 11: Accuracy results for Basque monolingual discriminative experiments

E Additional Generative Results

	Llama-3.1-Instruct-8B					**Gemma-2-it-9B**				
	nli-zero	nli-few	qa-zero	qa-few	chain	nli-zero	nli-few	qa-zero	qa-few	chain
XNLIeu	20.30	16.63	09.16	38.50	51.76	55.61	38.66	37.96	44.51	48.88
XNLIeu$_{\text{native}}$	21.36	17.90	07.05	36.24	41.83	61.19	39.94	41.55	49.11	54.43
XNLIeu$_{\text{var}}$	20.45	13.04	14.33	37.04	46.22	53.47	39.04	36.13	41.72	45.53

(a) Accuracy results with generative LLMs on Basque data.

	Llama-3.1-Instruct-8B					**Gemma-2-it-9B**				
	nli-zero	nli-few	qa-zero	qa-few	chain	nli-zero	nli-few	qa-zero	qa-few	chain
XNLIes	27.96	22.87	16.43	49.28	57.78	64.11	55.57	44.73	57.41	66.83
XNLIeu2es$_{\text{native}}$	28.34	15.62	23.19	48.79	62.80	69.24	55.72	50.24	59.42	71.82
XNLIes$_{\text{var}}$	26.73	21.62	19.37	48.35	56.46	67.63	53.14	44.28	55.72	68.92

(b) Accuracy results with generative LLMs on Spanish data.

Table 12: Results with 8B and 9B LLMs.

F Per-dialect Accuracy Results

	Model transfer			**Translate-train**			**Translate-test**		
	Western	Central	Navarrese	Western	Central	Navarrese	Western	Central	Navarrese
XLM-RoBERTa large	71.25	67.17	71.43	73.75	74.45	71.43	71.67	72.42	74.60
mDeBERTa	62.08	70.90	60.32	66.25	72.59	66.67	69.58	71.40	73.02

Table 13: Accuracy results for discriminative models in Basque dialects

	Model transfer				
	Cuba	Ecuador	SpainAndalusia	SpainTenerife	Uruguay
XLM-RoBERTa large	79.49	78.67	69.70	58.97	82.05
mDeBERTa	76.92	78.67	70.30	58.97	79.49
	Translate-train				
	Cuba	Ecuador	SpainAndalusia	SpainTenerife	Uruguay
XLM-RoBERTa large	79.49	84.67	70.91	61.54	82.05
mDeBERTa	73.33	81.33	69.09	53.85	87.18
	Translate-test				
	Cuba	Ecuador	SpainAndalusia	SpainTenerife	Uruguay
XLM-RoBERTa large	72.82	76.67	72.12	64.10	76.92
mDeBERTa	69.23	77.33	67.88	62.39	79.49

Table 14: Accuracy results for discriminative modes in Spanish variants

G Spanish Correlation Between Adaptation Types and Accuracy

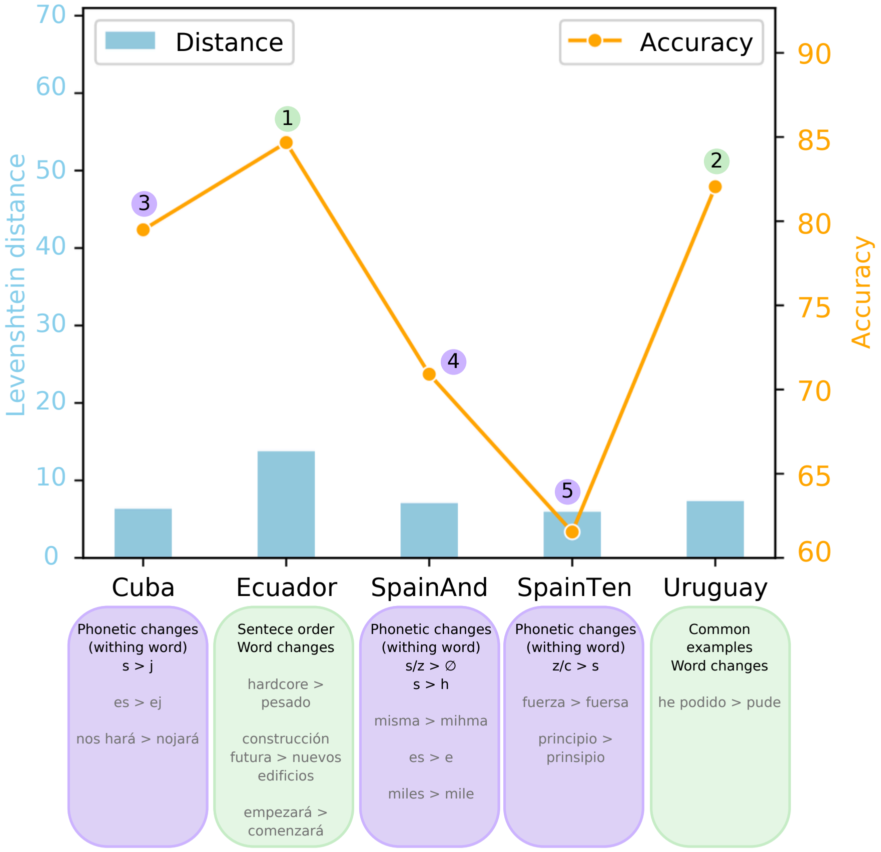


Figure 4: Spanish accuracy results and its correlation to types of linguistic adaptations. 1 and 2 have the highest accuracies, but changes usually involve word changes. For 3, 4 and 5, the accuracy decreases respectively, as variations majorly involve phonetic changes.

Compositionality and Event Retrieval in Complement Coercion: A Study of Language Models in a Low-resource Setting

Matteo Radaelli
Norwegian University of Science
and Technology
matteo.radaelli@ntnu.no

Emmanuele Chersoni
The Hong Kong Polytechnic University
emmanuele.chersoni@polyu.edu.hk

Alessandro Lenci
University of Pisa
alessandro.lenci@unipi.it

Giosuè Baggio
Norwegian University of Science and Technology
giosue.baggio@ntnu.no

Abstract

In sentences such as *John began the book*, the complement noun phrase, lexically denoting an entity, is interpreted as denoting an event. This is known in linguistics as *complement coercion*: the event associated with the verb is not overtly expressed but can be recovered from the meanings of other constituents, context and world knowledge. We investigate whether language models (LMs) can exploit sentence structure and compositional meaning to recover plausible events in complement coercion. For the first time, we tested different LMs in Norwegian, a low-resource language with high syntactic variation in coercion constructions across aspectual verbs. Results reveal that LMs struggle with retrieving plausible events and with ranking them above less plausible ones. Moreover, we found that LMs do not exploit the compositional properties of coercion sentences in their predictions.

1 Introduction

Sentences like *John began the book* are examples of complement coercion, originating from a type-mismatch between the required verb argument and the observed one (Pustejovsky, 1991, 1995): the aspectual verb (e.g., *begin*) semantically requires an event-denoting argument but is composed with an entity as its syntactic complement. Although the event is not overtly expressed, a plausible candidate can often be recovered by exploiting lexical and contextual information (Pustejovsky, 1991, 1995; Lapata and Lascarides, 2003): the sentence above can be interpreted as meaning *John began* {reading, writing, ...} *the book*.

Complement coercion has drawn attention as a potential violation of the Fregean principle of compositionality. Compositionality implies that all aspects of sentence meaning should derive from the meanings of the constituent parts and the way they are combined syntactically (Asher, 2015; Baggio et al., 2012). The interpretation of various covert elements results from enriched composition: semantic processes that exploit conceptual meaning, discourse context and world knowledge (Pustejovsky, 1991, 1995; Jackendoff, 1997; Baggio, 2018).

Language Models (LM) based on the Transformer architecture (Vaswani et al., 2017) have demonstrated remarkable capabilities in a wide range of NLP tasks, including natural language understanding. Despite their success, few studies have focused on enriched composition phenomena such as complement coercion (Gu, 2022; Ye et al., 2022). Some studies have investigated LM performance viewing complement coercion as an event retrieval task and demonstrating the challenges of recovering underlying semantic information from coercion sentences (Rambelli et al., 2020; Ye et al., 2022; Gietz and Beekhuizen, 2022; Gu, 2022; Im and Lee, 2024; Rambelli et al., 2024). However, most studies have been conducted in English, a language with low variability in the syntax of coercion constructions. As a consequence, little is known about the interplay of syntax and semantics in covert event retrieval in LMs: (how) do machines exploit compositional properties of coercion sentences to arrive at plausible interpretations?

The current study makes three contributions. First, to our knowledge, it is the first study of LMs on complement coercion that uses a language other than English (Norwegian) and that evaluates and compares different LMs (autoencoders and autoregressive models). Second, we investigate the interaction between different aspectual verbs Katsika et al. (2012) and post-verbal constituents in canonical syntactic constructions. Norwegian shows some variation in how complement coercion is syntactically realized, and therefore allows us to probe whether LMs are sensitive to syntactic and compositional semantic properties of these constructions across aspectual verbs. Finally, Norwegian is cur-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 469–480
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

rently considered a low-resource language (Kummervold et al., 2022; Liu et al., 2024; Samuel et al., 2025), and we are releasing our evaluation dataset for complement coercion resolution in Norwegian. Complement coercion with aspectual verbs is statistically rare in Norwegian corpora (see below): recovering implicit events could be challenging for a 'data hungry' technology such as LMs.

2 Related Work

2.1 Complement Coercion in Norwegian

Complement coercion has been studied in several high-resource languages. Apart from English, we find studies on German (Rüd and Zarcone, 2011; Zarcone and Padó, 2011; Zarcone et al., 2012, 2014), French (Godard and Jayez, 1993; Pustejovsky and Bouillon, 1995), Dutch (Sweep, 2012), and Chinese (Hsu and Hsieh, 2013), while there has been little research on Scandinavian languages. Spalek (2015) analyzed the cessation verb *avslutte* (to conclude), comparing Norwegian, English, Spanish and German. Spalek concluded that coercion is limited to a reduced set of entities that can be combined with the verb, especially "information-content entities" (e.g., text) (Spalek, 2015, p. 531). Spalek and Sæbø (2019) argued that Norwegian speakers tend to combine dynamic verbs with specific particles that denote a particular stage of the event (e.g., *å stryke ferdig*, to finish ironing).

Radaelli and Baggio (2025) conducted a study on the Norwegian Colossal Corpus (NCC) (Kummervold et al., 2022), a large set of corpora that includes approximately 21M documents for a total of 7B tokens. The study examined a wider class of aspectual verbs than previous theoretical research: *begynne* (to begin), *starte* (to start), *fortsette* (to continue), *ende* (to end), and *avslutte* (to conclude). The authors found that the syntax of complement coercion in Norwegian can vary according to the aspectual verb. Initiation verbs are usually combined with PPs introduced by the prepositions *på* or *med*:

(1) *Gutten begynte | startet på | med boken.*
(The boy began | started [with] the book.)

These combinations appear with higher frequency in complement coercion sentences compared to other aspectual verbs. The continuation verb *fortsette* introduces coercion mainly with *med*-PPs and, to a lesser extent, directly with nominals:

(2) *Gutten fortsatte [med] boken.*
(The boy continued [with] the book.)

The cessation verb *avslutte* prefers direct objects, while *med*-prepositional phrases appear less often:

(3) *Gutten avlsuttet [med] boken.*
(The boy finished [with] the book.)

Not all aspectual verbs can trigger complement coercion (e.g., the verb *ende* was excluded), nor do aspectual verbs significantly differ in occurrence frequency in coercion constructions.

The corpus analysis confirms the findings of Spalek (2015) and Spalek and Sæbø (2019): complement coercion occurs with a restricted set of entity categories. Although a similar trend can be found also in other languages (e.g., see Verspoor (1997) for English and Rüd and Zarcone (2011) for German), Norwegian shows even less variability, reducing the set of entities primarily to everyday objects such as text, music, songs, food and drinks.

Considering Pustejovsky's Generative Lexicon perspective (Pustejovsky, 1995), the productivity of coercion can also be limited by the interaction of syntactic and semantic factors. If, on the one hand, entities admit either AGENTIVE or TELIC qualia readings, their combination with prepositions may further reduce the set of plausible event candidates. The preposition *med* appears to play a 'passe-partout' role, with greater flexibility in event interpretation, including not only default qualia readings but also contextual information, if present. The preposition *på*, on the other hand, tends to further constrain interpretations: the corpus data showed a stronger tendency to express AGENTIVE interpretations with entities that are created rather than used. Radaelli and Baggio (2025) also found that Norwegian speakers prefer to express similar concepts to complement coercion through a broad range of phrasal constructions (e.g., *å sette i gang*, to begin). The study concluded that complement coercion is a relatively low-frequency phenomenon, with around 1500 cases over 80,000 sentences with aspectual verbs and syntactic constructions compatible with coercion.

2.2 LM Approaches to Complement Coercion

Before the LM era, complement coercion interpretation was carried out via either probabilistic (Lapata and Lascarides, 2003; Shutova, 2009; Shutova et al., 2013) or distributional semantic modeling (Zarcone et al., 2012, 2013; Chersoni et al., 2017;

McGregor et al., 2017; Chersoni et al., 2021). In one of the first studies testing LMs on complement coercion, Rambelli et al. (2020) evaluated the events retrieved by pretrained models of the BERT and the GPT families. They found that LMs performed well, but not significantly better than the best distributional models.

Ye et al. (2022) argued that Transformer-based models can learn coercion interpretations via *dense paraphrasing* (DP): DP involves the reformulation of a given coercion sentence in a way that eventive information is revealed, ambiguity is removed and the original sentence meaning is preserved. They found that BERT struggles in interpreting coercion, but a fine-tuning with explicitly paraphrased sentences improved its performance.

Finally, Gu (2022) investigated the behavior of GPT-2 on complement coercion by analyzing surprisal estimates. The goal was to understand how LMs process coercion constructions at the VP. Significant surprisal effects were observed at the target region, aligning with psycholinguistic findings of increased processing costs at the complement (McElree et al., 2001; Traxler et al., 2002; Baggio et al., 2010, 2011, 2016).

3 Experimental Settings

3.1 Task Proposal

In previous research on complement coercion in LMs, evaluations typically compared a narrow set of high-likelihood predictions against a predefined set of gold standard outputs. In contrast, our study introduces a novel evaluation approach, based on a ranked prediction distribution of class-specific verbs, rather than just the most probable outputs: for every context-neutral sentence[1] s belonging to a set S, a given model m generates a set of top-k ranked output predictions $O = \{o_1 ... o_k\}$. We then evaluate each output with a mean average precision metric, allowing us to determine not only whether a model predicts covert events, but also to what extent LMs consistently predict plausible event interpretations in their rankings.

The distribution should reflect a re-ranking of tokens when the model is exposed to coercion sentences, providing evidence of its sensitivity to coercion. In cases where a LM is exposed to a sentence such as *The boy began the book*, we expect that the combination of the triplet <subject, aspectual verb, entity> would result in a re-ranking of candidate implicit events (see Figure 1): the ranking should reflect the interaction of the triplet composition, where plausible verbs (events) are collocated at the top of the rank as the most likely interpretations.

Instead of using a set of predefined events, our study will consider any event that meets the syntactic and semantic constraints of complement coercion as correct. According to Piñango and Deo (2016) and Spalek and Sæbø (2019), the covert event of a complement coercion sentence should be *telic*: combined with the subject and complement, it should establish a natural endpoint or goal state.

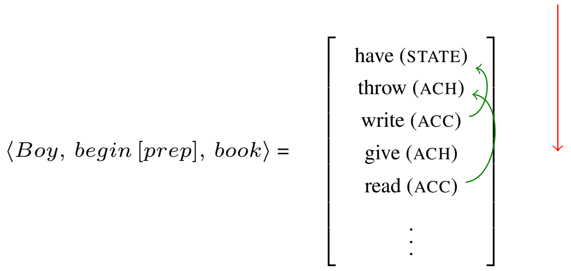


Figure 1: An example of re-ranking candidate events when the expressions in the given triplet are composed in a sentence. The matrix shows output verbs associated with their Aktionsart class such as state, achievements (ACH), and accomplishments (ACC).

Given the above requirements, events predicted by LMs should be evaluated considering their Aktionsart (lexical aspect) class by using Vendler's classification system (Vendler, 1967). We identify the class of **accomplishment** verbs as our ground truth in this task, as they denote dynamic and durative actions with a specific endpoint, aligning with the telicity criterion by Spalek and Sæbø (2019). As there is no predefined set of implicit events for interpreting a coercion sentence, we consider as compositionally plausible candidates all predicted verbs that belong to the accomplishment class. In case a model predicts events weakly associated with a specific coercion triplet (e.g., *begin the book* → *eat*, see Lascarides and Copestake 1998), this does not necessarily indicate low performance: the output can count as correct, if the retrieved event is an accomplishment. It is possible to construct contexts where even apparently deviant events are plausible, so long as they are accomplishments: e.g., *The goat began* {eating} *the book.*

---

[1] Context-neutral, canonical coercion sentences include the subject, the aspectual verb and its complement, with unmarked word order and no additional sentence context.

3.2 Dataset

We created a new dataset of sentence pairs with (a) a context-neutral sentence with a coercion triplet and variable syntactic structure (*på*-NP, *med*-NP, NP) and (b) a sentence prompting event resolution:

(a) `Kim {VERB-FIN} {PREP|Ø} {ENTITY-DEF}.`

(b) `Det som Kim {VERB-FIN} å gjøre, var å [MASK]. (What Kim {VERB-FIN} to do, was [MASK].)`

Each placeholder in brackets is replaced with the relevant lexical item. The template encompasses a combination of the following elements:

- 90 entities ({ENTITY-DEF}) were carefully selected to represent real artifacts, avoiding abstract and ambiguous concepts. In addition, following Piñango and Deo (2016, p. 387), we used entities that can be semantically interpreted as "incremental theme arguments of the implicit event", a crucial element in coercion configurations. We included entities that never occurred in coercion sentences in the NCC corpus study, ensuring that the models, especially those trained exclusively on NCC, are exposed to sentences not seen during pretraining. Six distinct entity categories were used: *food*, *text*, *clothing*, *everyday objects*, *construction/housing*, and *entertainment*. All nouns were only used in definite form.
- Four aspectual verbs ({VERB-FIN}), namely ***begynne*** (begin), ***starte*** (start), ***fortsette*** (continue), and ***avslutte*** (finish) were composed with each entity. The verb was always presented in the simple past form (*preteritum*) in both sentences in a pair.
- Three syntactic constructions were used ({PREP|Ø}): the complement is either introduced by a PP with the prepositions **på** or **med** followed by the NP denoting an entity, or only by the latter NP.
- The same subject was used for every sentence, with a neutral name (*Kim*) to avoid gender and other biases that may affect the results.
- In all pairs, the prompt (b) included the [MASK] token the model has to predict.

A total of 1080 sentence pairs in standard written Bokmål form were used with each model.

Model	# Par.	Tr. Data
MBERT CASED/UNCASED	178M	3.3B*
NB-BERT-BASE	178M	7B
NB-BERT-LARGE	355M	7B
NORBERT	111M	1.9B
NORBERT2	125M	15B
NORBERT3-base	123M	25B
NORBERT3-large	353M	25B
NORBERT3-SMALL	40M	25B
NORBERT3-XS	15M	25B
NORBLOOM-7B-SCRATCH	7B	26.7B
NORGPT-369M	369M	25B
NORGPT-3B	3B	25B
NORGPT-3B-CONTINUE	3B	25B
NORLLAMA-3B	3B	26.7B
NORMISTRAL-7B-SCRATCH	7B	26.7B
NORMISTRAL-7B-WARM	7B	26.7B

Table 1: Tested LMs with approximate information on number of parameters (*#Par.*) and training data (*Tr. Data*). *mBERT was trained on 114 languages.

3.3 Models

We evaluated a total of 17 different pre-trained Norwegian LMs varying in architecture, parameter size and training data. The models belong to two broad families: BERT-like autoencoder models, and autoregressive models such as GPT-2 (Radford et al., 2019), LLAMA-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023) and Bloom (Scao et al., 2023) (Table 1). All models are available on Hugginface[2].

3.4 Baseline Model

To assess event retrieval in complement coercion by LMs, it is necessary to use a baseline model, here provided by the NCC, an open-source corpus used for training most LMs in Norwegian [3]. For each entity in the dataset, we extracted the most likely verbs (events) associated with the entity. The extracted verbs were determined on the basis of the Pointwise Mutual Information (PMI) score, a metric evaluating the association strength between two words $w1$ and $w2$ (Church and Hanks, 1990):

$$\mathrm{PMI}(w_1, w_2) = \log \frac{P(w_1, w_2)}{P(w_1) \cdot P(w_2)}$$

In our study, the score was calculated through the joint probability between each sentence predicate (event) and its object (entity) in the entire

---

[2] https://huggingface.co/

[3] The National Library of Norway, the maintainer of the NCC corpus, in December 2024 has updated its distribution policies, limiting access to some subcorpora. This has caused a decrease in the corpus size from approximately 7B to 4.5B tokens. For further information, see `https://huggingface.co/datasets/NbAiLab/NCC`

corpus. The PMI score provides a measure of general lexical association between verbs and entities. A comparison with this baseline allows us to understand whether an LM genuinely interprets covert events based on coercion settings or simply mirrors statistical co-occurrence patterns observed during pre-training.

3.5 Evaluation and Annotation

We used two common evaluation metrics. One is mean average precision (mAP) (see Manning et al. (2009, from p. 159) and Kotlerman et al. (2010)):

$$\text{mAP} = \frac{1}{S}\sum_{s=1}^{S}\text{AP}(s)$$

It consists of the weighted means of average precision (AP) scores across all sentences (S):

$$\text{AP(q)} = \sum_{k=1}^{5} P(k) \cdot \Delta R(k)$$

where P is the precision value calculated at the cut-off rank k and $\Delta R(k)$ is the change in recall (R) from rank $k-1$ to k. mAP provides the ranking direction of models when complement coercion occurs. A high mAP value indicates a model that mostly considers accomplishment verbs in the prediction list, collocating them at the top, whereas a low mAP value suggests a failure in prioritizing accomplishment verbs as completions.

The second metric is the mean top-ranked accuracy (A1) in all sentences, considering only the most likely prediction in the ranking. This metric allows us to study what types of verb (events) the models consider as the most salient ones.

4 Results and Task Discussion

Table 2 shows the performance results of all LMs in the covert event retrieval task in Norwegian, with mAP and A1 scores. Model performance varies according to the interplay of two main factors: model framework and model size (number of parameters and training data). The NORBERT3 family shows relatively high performance compared to other BERT-like frameworks, with NORBERT3-BASE and NORBERT3-LARGE outperforming the baseline on both measures. Larger LMs outperform NB-BERT models and the previous generations of NORBERT models, which showed poorer performances, possibly due to less training data available. Models like NORBERT3-XS performed less well probably due to their reduced parameter size, despite the same amount of training data.

Model	**mAP**	**A1**
NCC (baseline)	0.59	0.47
NorGPT-369M	0.56	0.54
NorGPT-3B	0.48	0.42
NorGPT-3B-CONTINUE	0.46	0.42
NorLlama-3B	0.71	0.67
BERT-BASE-MULTILINGUAL-CASED	0.07	0.00
BERT-BASE-MULTILINGUAL-UNCASED	0.27	0.22
NB-BERT-BASE	0.38	0.33
NB-BERT-LARGE	0.54	0.47
NORBERT	0.25	0.18
NORBERT2	0.44	0.34
NORBERT3-BASE	0.63	0.58
NORBERT3-LARGE	0.60	0.55
NORBERT3-SMALL	0.59	0.55
NORBERT3-XS	0.29	0.16
NORBLOOM-7B-SCRATCH	0.46	0.34
NORMISTRAL-7B-SCRATCH	0.38	0.29
NORMISTRAL-7B-WARM	0.63	0.54

Table 2: Mean average precision (mAP) and top-rank accuracy (A1) results in the covert event retrieval task in Norwegian. NorLlama-3B is the best performing model overall.

Almost all GPT-2-based models, as well as NORBLOOM-7B-SCRATCH and NORMISTRAL-7B-SCRATCH performed poorly, ranking below the baseline, despite their size. NORMISTRAL-7B-WARM outperforms the baseline in both cases, compared to the version trained from scratch: pretraining on the English vanilla version and successive pretraining on Norwegian data may have given the model an advantage, allowing for the transfer of rich representations from English text. Finally, NorLlama-3B can be considered as the most capable model among those tested here. Its success could be attributed perhaps to its large training corpus, with more than 25B training tokens in Norwegian and other Scandinavian languages.

Language models generally struggle to perform the completion task. Overall low mAP scores suggest difficulties in generating plausible accomplishments among high-ranked candidate mask replacements. This is confirmed when cross-analyzing A1 scores: even the best model, NorLlama-3B fails to reach a 70% level of accuracy, indicating that non-accomplishments and other implausible verbs are predicted as candidate interpretations at the top of the list. Similarly, the top-10 ranked models achieve an A1 score ranging from 0.42 to 0.58: they have around 50% chance of failing to rank accomplishments at the top.

We will now turn to an analysis of model performance taking into account both mAP and A1

NorLlama-3B **verb**	**prep**	**mAP**	**A1**
avslutte	ø	0.66	0.61
	med	0.75	0.69
	på	0.64	0.53
begynne	ø	0.75	0.72
	med	0.79	0.81
	på	0.73	0.71
fortsette	ø	0.64	0.57
	med	0.64	0.56
	på	0.59	0.46
starte	ø	0.79	0.80
	med	0.81	0.83
	på	0.76	0.76

Table 3: Mean average precision (mAP) and top-rank accuracy (A1) results for NORLLAMA-3B categorized by aspectual verbs (*begynne*, *starte*, *fortsette* and *avslutte*) and syntactic composition (introduced by prepositions, *på* or *med*, or by a nominal, *ø*) in coercion sentences.

scores subdivided according to aspectual verbs and their syntactic structures in coercion sentences. For the sake of simplicity, we will consider the best performing model NORLLAMA-3B. The results are shown in Table 3.

Consistently high mAP scores are found with initiation verbs. The verb *starte* shows high mAP scores reaching 0.81 precision when entity arguments in coercion sentences are introduced by the preposition *med*, 0.79 with nominals and 0.76 with the preposition *på*. The verb *begynne* was associated with worse performance, while showing a similar trend as *starte*. Sentences with entity arguments introduced by *med* reached 0.79 precision, 0.75 with nominals and 0.73 with *på*. The two remaining aspectual verbs showed similar results, and arguments with *med* as preposition obtained higher precision scores. In sentences with *fortsette*, both nominals and *med*-prepositional phrases reached the same score (0.64).

A1 scores show a similar trend. The model performs better when coercion sentences are introduced by *starte*, with 0.83 of A1 accuracy when the entity NP is introduced by *med*-PP, 0.80 without a preposition and 0.76 by *på*-PP. The verb *begynne* also serves as a trigger for complement coercion, with an A1 score of 0.81 with *med*-prepositional phrases, 0.72 with simple nominals and 0.71 with verb argument phrases introduced by *på*.

Two key observations are suggested by this analysis. First, different aspectual verbs are associated with differences in model performance. Our results indicate that the model can recover the implicit meaning more easily with initiation verbs in coercion sentences. This is consistent with the corpus analysis of Radaelli and Baggio (2025), which showed that among all aspectual verbs, initiation verbs feature more frequently in coercion sentences. Second, we only find weak differences in performance as a function of the syntax of post-verbal constituents. This suggests that the type of syntactic structures in complement coercion sentences plays only a minor role in the model's process of recovery of implicit meaning.

4.1 Sentence surprisal

Previous studies (see above) indicated that LMs struggle to consistently retrieve covert events in complement coercion sentences. To understand the reasons behind these prediction difficulties, one can study the model's behavior when it is exposed to complement coercion sentences. We conducted a further analysis that complements the previous ranking results by computing surprisal estimates for coercion sentences. Surprisal is used in NLP and psycholinguistic studies to quantify effort during sentence processing (Hale, 2001; Levy, 2008; Smith and Levy, 2013; Salicchi et al., 2023; Oh and Schuler, 2023; Shain et al., 2024):

$$S(w_i) = -\log_2 P(w_i \mid w_1, \ldots, w_{i-1})$$

Surprisal measures how unexpected a given word (w_i) is, given its left context ($w_1 ... w_{i-1}$). Higher surprisal values indicate greater processing difficulty, as upcoming words are less predictable.

We will use surprisal to assess whether models expect an entity-denoting noun in coercion triples. Specifically, we will compare surprisal estimates for complement coercion sentences (e.g., *Kim begynte på boken*, Kim began (on) the book) with their overt counterparts (e.g., *Kim leste boken*, Kim read the book). The events were selected considering the highest PMI scores between each accomplishment and its associated entity. In total, we examined 2,160 sentences, using the same sentences from the previous task (1,080 coercion, 1,080 overt) combining all aspectual verbs, all entities, and the same three different syntactic structures. To compute surprisal estimates, we used log-probabilities provided by model logits. As coercion and overt sentences may differ in length, we computed surprisal for sentences as the mean of each word's surprisal:

$$S_{mean}(s) = \frac{1}{N}\sum_{i=1}^{N} S(w_i)$$

where N is the number of tokens in a sentence s. Here too, we tested surprisal for NorLlama-3B as a high performance model in this task. For the calculation of surprisal estimates, we used the tool *minicons* on Python. The data were analyzed using Wilcoxon signed-rank tests to compare the surprisal values between coercion and overt sentences. We hypothesized that the model would show higher surprisal values for coercion sentences than for overt ones. The results confirmed the hypothesis, showing a statistical difference in surprisal ($W = 367176, p < 0.001$). This suggests a tendency of the model to assign prediction logits with lower probabilities for coercion sentences.

In order to analyze the extent to which syntactic structure can influence surprisal in coercion sentences, we compared two regression models. As a baseline, we ran a model on surprisal using only sentence type (coercion *vs* explicit) and sequence length as predictors. The second model also included syntactic structure as a predictor (with *på*-PPs, *med*-PPs and direct nominals as levels). The baseline model ($R^2 = 0.173$) revealed that coercion sentences significantly increased surprisal. Moreover, sequence length negatively correlated with surprisal, meaning that longer sentences led to lower surprisal values. The second model ($R^2 = 0.181$) shows a significant positive trend in the coercion condition, as the baseline model. On the other hand, sequence length shows in this case a positive effect on surprisal. Sentences with *med*-prepositional phrases demonstrate lowest surprisal, while sentences with *på* exhibit slightly higher surprisal, but still lower than in the nominal conditions. Comparing the variance of the two models ($\Delta R^2 = +0.008$), we find small improvements attributable to syntax. Prepositions therefore reduce surprisal in comparison to sentences with direct nominals, where *med*-sentences led to lower surprisal, followed by *på* (Figure 2).

	Baseline model Coefficient (β)	**Model With Syntax Coefficient (β)**
Intercept (Nominals)	9.7983 ($p < 0.001$)	2.0463 ($p < 0.001$)
Coercion	0.9429 ($p < 0.001$)	1.4946 ($p < 0.001$)
Sequence Length	-1.0612 ($p < 0.001$)	1.3388 ($p < 0.001$)
Explicit	—	0.5517 ($p < 0.001$)
Med	—	-2.5418 ($p < 0.001$)
På	—	-2.2583 ($p < 0.001$)

Table 4: Effects of syntax on sentence surprisal.

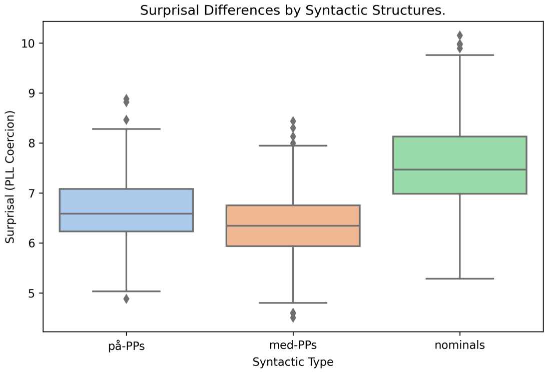


Figure 2: Surprisal values across syntactic structures.

5 Error Analysis

To study model errors, a relatively straightforward approach is to examine the overall prediction distribution of events and their Aktionsart. For practical reasons, the analysis is restricted again to the best performing model, NorLlama-3B. The analysis revealed the following findings. First, among 5,400 predictions made in 1,080 coercion sentences (divided by 90 different entities and presented with 4 different aspectual verbs and 3 post-verbal syntactic constructions), the model predicted 68 unique events. This small set suggests, on the one hand, that the model tends to predict events by avoiding many unrelated or random outputs. However, the low variation of events also suggests a tendency to reuse the same verbs across many entities.

Second, the distribution of events as predicted by the model is skewed and follows a Zipfian law, with the first most frequently predicted events in the ranking accounting for a substantial proportion of the total distribution, while the frequency of verbs ranked at lower positions rapidly decreases. Table 5 presents the distribution of the first 10 most predicted events across all coercion instances, including both their absolute and relative frequencies based on the total 5,400 predictions (5 predictions per instance). The most frequent verbs predicted by the model are *lage* (make), followed by *sette* (put/set), *ta* (take), *få*, and *gi* (give), which together amount to almost 67% of the total predictions. The remaining verbs have lower frequencies with a considerable subset of events that occur only once. This long-tail behavior further strengthens the hypothesis of a biased tendency of the model towards a very limited set of events.

Third, based on a close qualitative examination

Verb	Freq. (Rel._freq)
lage (make)	969 (0.18)
sette (put/set)	768 (0.14)
ta (take)	762 (0.14)
få (get/receive)	739 (0.14)
gi (give)	387 (0.07)
skrive (write)	175 (0.03)
spille (play)	155 (0.03)
legge (lay/put down)	113 (0.02)
male (paint)	109 (0.02)
dele (share/divide)	96 (0.02)
gå (go / walk)	96 (0.02)
sy (sew)	94 (0.02)
blande (mix)	84 (0.02)
strikke (knit)	83 (0.02)
synge (sing)	81 (0.02)

Table 5: Top predicted events made by NORLLAMA-3B, showing both absolute frequency and relative frequency out of a total of 5,400 model outputs.

of the restricted set of predicted events at the top of the ranking, we can notice the following patterns. The most frequently output events are usually non-accomplishment verbs: the only acceptable accomplishment verb is *lage*, which is the most frequent. Yet, this verb is polysemous and can be combined with a wide range of entities, denoting an action of creating or producing something, e.g., *lage pizza* (make pizza), *lage skulptur* (make a sculpture) and *lage sang* (compose a song). In contrast, other frequently predicted events, like *ta* (take), *få* (get) and *gi* (give), are either implausible in many coercion instances or typically denote achievements, and are therefore not acceptable in coercion constructions.

Despite a quite positive performance overall of NORLLAMA-3B in interpreting coercion items, the strong presence of semantically inappropriate verbs in the ranking may be due to their high frequency in the corpora used during pretraining. Since the task was designed to constrain the model to retrieve only infinitival verbs, the prediction of verbs that are not plausible accomplishments suggests that the model may rely more on the co-occurrence frequency between a verb and its nominal object during training, rather than on the semantic compatibility between the event and the entity, even in contexts in which a more compositionally appropriate event would be expected and could be retrieved.

6 General Discussion and Conclusion

The analyses carried out in the present study clearly show that complement coercion remains an open challenge for LMs in low-resource languages such as Norwegian. We investigated the extent to which LMs could recover implicit events in complement coercion sentences. If models recognize these as coercion constructions, that require event retrieval, they should be able to distribute verb (event) predictions in such a way that accomplishments are ranked as the most probable covert events.

However, the outcomes of the event retrieval task indicate that LMs still have difficulties recovering viable implicit events. In particular, A1 scores are consistently low across models, suggesting a failure to retrieve potential accomplishment verbs as the most likely event predictions in the task. Moreover, the mAP scores confirmed the models' limitations, as they fail to systematically and consistently rank accomplishment verbs higher. Only few models could outperform the baseline, whose predictions are based on simple statistical calculations on the NCC corpus frequency: this is significant, considering that such models were trained on corpora 3.5 times larger than the baseline. The results also highlight performance differences across models:

- NORLLAMA-3B outperformed all the models that were tested here; its success may be due to its new improved architecture and training optimization (e.g., SwiGLU activation function, Grouped query attention (GQA) mechanism, rotary positional embeddings), combined with a large amount of training data.
- On the other hand, even the largest GPT-class models could not perform the task efficiently. The traditional autoregressive GPT-2 may lack an architecture that can capture covert information like covert events.
- Even LMs such as NORBLOOM-7B-SCRATCH and NORMISTRAL-7B-SCRATCH performed poorly for their size. Their low performance could be due to training carried out exclusively on Norwegian data, especially compared to the best performing NORMISTRAL-7B-WARM with a pre-training phase that also included English data.
- The NORBERT3 family, in particular the base and large versions, could attain moderate performance levels despite their reduced number

of parameters. The BERT architecture then seems to be well-suited for learning and storing world knowledge and relational knowledge between words during pretraining, making them effective in cloze tasks (Petroni et al., 2019; Rogers et al., 2020). In addition, their customized autoencoder framework, incorporating the extended MLM pre-training task (Samuel et al., 2023), may have facilitated acquisition of syntactic and semantic information relevant for the present task. To this purpose, it should be noted that also in the complement coercion study of Rambelli et al. (2020) in English, a bidirectional architecture (RoBERTa) was the one showing the highest correlations with human production frequencies for the candidate covert event. However, vanilla architectures combined with less training data would drastically reduce performance as seen in the NB-BERT models.

To better understand how LMs process complement coercion sentences and investigate the causes behind their difficulties in event retrieval task, we compared surprisal estimates between coercion sentences and their overt event counterparts. Higher surprisal values for complement coercion sentences suggest that LMs generally find coercion constructions less predictable, which should be expected given their relative infrequency in Norwegian corpora. However, rare constructions in human language can still be interpreted compositionally by exploiting lexical meaning and syntactic structure, even when context is minimal or absent (Baggio, 2018, 2021). Overall, our results suggest that many LMs are largely unable to make productive use of the available compositional information to generate accomplishments as plausible event completions in complement coercion sentences. These results apply to Norwegian, but may well extend to other languages with similar characteristics, such as other Scandinavian or Germanic languages, and to other constructions infrequent in linguistic corpora.

Language models have been often argued to lack ‘common sense’, which makes them unsuitable as (general) problem solvers in real-world situations. Our results show that LMs may also have limited *linguistic common sense*, the ability to select and use all and only relevant (non)linguistic knowledge to interpret inputs to comprehension and learning (Lascarides and Copestake, 1998; Piñango and Deo, 2016; Baggio, 2018; Rambelli et al., 2024).

A more detailed analysis of the best performing model (NORLLAMA-3B) revealed only moderate variation in performance according to the specific aspectual verb used. Initiation verbs lead to better performance. Based on results of corpus studies, this may be due to stronger statistical associations between these aspectual verbs and (particular classes of) entity-denoting nominals. However, we could not find clear differences between different syntactic constructions within the same aspectual verbs, which suggests that models do not exploit differences in syntactic structure to recognize these as coercion constructions and accordingly attempt the retrieval of plausible accomplishments.

Linear regression models were also employed to assess whether coercion surprisal estimates were influenced by the syntactic structures proposed in the dataset. Results revealed weak differences in surprisal estimates, especially between coercion sentences with entity-denoting complements introduced by prepositions or directly by NPs, showing greater processing difficulties in the latter cases. This partially aligns with the results presented in table 3, where nominals led to lower scores, while *med*-PPs were associated with better performance. Furthermore, LM behavior aligns weakly with the NCC corpus study by Radaelli and Baggio (2025): the authors found that *med*-prepositional phrases occur more frequently in coercion constructions and allow greater flexibility in event interpretations.

Considering LM’s failure to exploit compositionality (lexical meaning and syntactic structure) with complement coercion sentences, future work should explore what other factors can impact LM’s performance in this type of task. There are at least two possible research directions. First, an analysis of the role of linguistic context as a factor in performance improvement: what aspects of sentence or discourse context can facilitate event retrieval? Second, an analysis of the extent to which LM’s performance is dependent on ontology: can event retrieval be facilitated by specific classes of entities, as is suggested by theoretical linguistic and corpus research?

7 Acknowledgments

EC was supported by a GRF grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 15612222). We thank three anonymous reviewers for their valuable suggestions.

References

Nicholas Asher. 2015. Types, meanings and coercions in lexical semantics. *Lingua*, 157:66–82.

Giosuè Baggio, Keith Stenning, and Michiel Van Lambalgen. 2016. Semantics and cognition. In Maria Aloni and Paul Dekker, editors, *The Cambridge Handbook of Formal Semantics*, pages 756–774. Cambridge University Press.

Giosuè Baggio, Michiel Van Lambalgen, and Peter Hagoort. 2011. The balance between memory and unification in semantics: A dynamic account of the N400. *Language and Cognitive Processes*, 26(9):1338–1367.

Giosuè Baggio, Michiel Van Lambalgen, and Peter Hagoort. 2012. The processing consequences of compositionality. In Markus Werning, Wolfram Hinzen, and Edouard Machery, editors, *The Oxford Handbook of Compositionality*, pages 655–672. Oxford University Press.

Giosuè Baggio. 2018. *Meaning in the Brain*. MIT Press.

Giosuè Baggio. 2021. Compositionality in a parallel architecture for language processing. *Cognitive Science*, 45(5):e12949.

Giosuè Baggio, Travis Choma, Michiel van Lambalgen, and Peter Hagoort. 2010. Coercion and Compositionality. *Journal of Cognitive Neuroscience*, 22(9):2131–2140.

Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Logical metonymy in a distributional model of sentence comprehension. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*.

Emmanuele Chersoni, Enrico Santus, Alessandro Lenci, Philippe Blache, and Chu-Ren Huang. 2021. Not all arguments are processed equally: A distributional model of argument complexity. *Language Resources and Evaluation*, 55(4):1–28.

Kenneth Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Frederick G Gietz and Barend Beekhuizen. 2022. Remodelling complement coercion interpretation. *Proceedings of the Society for Computation in Linguistics 2022*, pages 158–170.

Daniele Godard and Jacques Jayez. 1993. Towards a proper treatment of coercion phenomena. In *EACL '93: Proceedings of the sixth conference on European chapter of the Association for Computational Linguistics*, pages 168–177.

Yuling Gu. 2022. Measure more, question more: Experimental studies on transformer-based language models and complement coercion. *arXiv preprint arXiv:2212.10536*.

John Hale. 2001. A probabilistic earley parser as a psycholinguistic model. In *NAACL '01: Proceedings of the second meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, pages 1–8.

Chan-Chia Hsu and Shu-Kai Hsieh. 2013. To Coerce or Not to Coerce: A Corpus-based Exploration of Some Complement Coercion Verbs in Chinese. In *Proceedings of the 6th International Conference on Generative Approaches to the Lexicon (GL2013)*, pages 13–20.

Seohyun Im and Chungmin Lee. 2024. What GPT-4 knows about aspectual coercion: Focused on "begin the book". In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon @ LREC-COLING 2024*, pages 56–67.

Ray Jackendoff. 1997. *The Architecture of the Language Faculty*. MIT Press.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Argyro Katsika, David Braze, Ashwini Deo, and Maria Mercedes Piñango. 2012. Complement Coercion: Distinguishing between type-shifting and pragmatic inferencing. *The Mental Lexicon*, 7(1):58–76.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Per Kummervold, Freddy Wetjen, and Javier de la Rosa. 2022. The Norwegian Colossal Corpus: A Text Corpus for Training Large Norwegian Language Models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3852–3860.

Maria Lapata and Alex Lascarides. 2003. A Probabilistic Account of Logical Metonymy. *Computational Linguistics*, 29(2):261–315.

Alex Lascarides and Ann Copestake. 1998. Pragmatics and word meaning. *Journal of linguistics*, 34(2):387–414.

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

Peng Liu, Lemei Zhang, Terje Farup, Even W Lauvrak, Jon Espen Ingvaldsen, Simen Eide, Jon Atle Gulla, and Zhirong Yang. 2024. NLEBench+NorGLM: A Comprehensive Empirical Analysis and Benchmark Dataset for Generative Language Models in Norwegian. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5543–5560.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2009. *An Introduction to Information Retrieval*. Cambridge University Press.

Brian McElree, Matthew J Traxler, Martin J Pickering, Rachel E Seely, and Ray Jackendoff. 2001. Reading time evidence for enriched composition. *Cognition*, 78(1):B17–B25.

Stephen McGregor, Elisabetta Ježek, Matthew Purver, and Geraint Wiggins. 2017. A geometric method for detecting semantic coercion. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*.

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.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Maria Mercedes Piñango and Ashwini Deo. 2016. Reanalyzing the Complement Coercion Effect through a Generalized Lexical Semantics for Aspectual Verbs. *Journal of Semantics*, 33(2):359–408.

James Pustejovsky. 1991. The Generative Lexicon. *Computational Linguistics*, 17(4):409–441.

James Pustejovsky. 1995. *The Generative Lexicon*. MIT Press.

James Pustejovsky and Pierrette Bouillon. 1995. Aspectual Coercion and Logical Polysemy. *Journal of Semantics*, 12(2):133–162.

Matteo Radaelli and Giosuè Baggio. 2025. Complement coercion with aspectual verbs is statistically infrequent in written norwegian. Forthcoming.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 1(8):9.

Giulia Rambelli, Emmanuele Chersoni, Alessandro Lenci, Philippe Blache, and Chu-Ren Huang. 2020. Comparing Probabilistic, Distributional and Transformer-based Models on Logical Metonymy Interpretation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 224–234.

Giulia Rambelli, Emmanuele Chersoni, Davide Testa, Philippe Blache, and Alessandro Lenci. 2024. Neural generative models and the parallel architecture of language: A critical review and outlook. *Topics in Cognitive Science*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Stefan Rüd and Alessandra Zarcone. 2011. Covert events and qualia structures for german verbs. In *Proceedings of the Metonymy 2011 Workshop*, pages 17–22.

Lavinia Salicchi, Emmanuele Chersoni, and Alessandro Lenci. 2023. A study on surprisal and semantic relatedness for eye-tracking data prediction. *Frontiers in Psychology*, 14.

David Samuel, Andrey Kutuzov, Samia Touileb, Erik Velldal, Lilja Øvrelid, Egil Rønningstad, Elina Sigdel, and Anna Palatkina. 2023. NorBench – A Benchmark for Norwegian Language Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 618–633.

David Samuel, Vladislav Mikhailov, Erik Velldal, Lilja Øvrelid, Lucas Georges Gabriel Charpentier, and Andrey Kutuzov. 2025. Small Languages, Big Models: A Study of Continual Training on Languages of Norway. *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 573–608.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, and 373 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

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

Ekaterina Shutova. 2009. Sense-based interpretation of logical metonymy using a statistical method. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, pages 1–9.

Ekaterina Shutova, Jakub Kaplan, Simone Teufel, and Anna Korhonen. 2013. A computational model of logical metonymy. *ACM Transactions on Speech and Language Processing*, 10(3):1–28.

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

Alexandra Anna Spalek. 2015. The Influence of Context in Meaning: The Panorama of Complement Coercion. In *Modeling and Using Context*, pages 526–531. Springer International Publishing.

Alexandra Anna Spalek and Kjell Johan Sæbø. 2019. To Finish in German and Mainland Scandinavian: Telicity and Incrementality. *Journal of Semantics*, 36(2):349–375.

Josefien Sweep. 2012. Logical Metonymy in Dutch and German: Equivalents of Begin, Finish, and Enjoy. *International Journal of Lexicography*, 25(2):117–151.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint*.

Matthew J Traxler, Martin J Pickering, and Brian McElree. 2002. Coercion in sentence processing: Evidence from eye-movements and self-paced reading. *Journal of Memory and Language*, 47(4):530–547.

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

Zeno Vendler. 1967. *Linguistics in Philosophy*. Cornell University Press.

Cornelia Maria Verspoor. 1997. Conventionality-governed logical metonymy. In *Proceedings of the Second International Workshop on Computational Semantics*, pages 300–312. Citeseer.

Bingyang Ye, Jingxuan Tu, Elisabetta Jezek, and James Pustejovsky. 2022. Interpreting logical metonymy through dense paraphrasing. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Alessandra Zarcone, Alessandro Lenci, Sebastian Padó, and Jason Utt. 2013. Fitting, not clashing! A distributional semantic model of logical metonymy. In *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)*, pages 404–410.

Alessandra Zarcone and Sebastian Padó. 2011. Generalized Event Knowledge in Logical Metonymy Resolution. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 33.

Alessandra Zarcone, Sebastian Padó, and Alessandro Lenci. 2014. Logical Metonymy Resolution in a Words-as-Cues Framework: Evidence From Self-Paced Reading and Probe Recognition. *Cognitive Science*, 38(5):973–996.

Alessandra Zarcone, Jason Utt, and Sebastian Padó. 2012. Modeling covert event retrieval in logical metonymy: probabilistic and distributional accounts. In *Proceedings of the 3rd Workshop on Cognitive Modeling and Computational Linguistics (CMCL 2012)*, pages 70–79.

DLU: Dictionary Look-Up Data and Prediction

David Strohmaier[1], Gladys Tyen[1]*, Hongyi Gu[2], Diane Nicholls[3],
Zheng Yuan[4, 1], Paula Buttery[1]

[1]ALTA Institute, [2]NetMind.AI
[3]Cambridge University Press & Assessment
[4]The University of Sheffield
Correspondence: david.strohmaier@cl.cam.ac.uk

Abstract

Knowing which words language learners struggle with is crucial for developing personalised education technologies. In this paper, we advocate for the novel task of "dictionary look-up prediction" as a means for evaluating the complexity of words in reading tasks. We release the *Dictionary Look-Up development* dataset (DLU-dev) and the *Dialogue Dictionary Look-Up* dataset (D-DLU), which is based on chatbot dialogues. We demonstrate that dictionary look-up is a challenging task for LLMs (results are presented for LLaMA, Gemma, and Longformer models). We explore finetuning with the ROC* loss function as a more appropriate loss for this task than the commonly used Binary Cross Entropy (BCE). We show that a feature-based model outperforms the LLMs. Finally, we investigate the transfer between DLU and the related tasks of Complex Word Identification (CWI) and Semantic Error Prediction (SEP), establishing new state-of-the-art results for SEP.

1 Introduction

When a learner is reading a text, they may encounter unfamiliar words. When this happens, a learner can choose to seek further information about the word, such as definitions or examples of use. The event of *looking-up* a word is an indication that the word was difficult for the learner in its textual context. By noting the look-up events of many learners, we can discover the relative *contextual lexical complexity* of words for different ability groups; and by collecting look-up data en masse, we can build empirical models of the same. Such models can be used to:

1. Improve readability of texts for specific learner groups;
2. Predict reading competence of learners; or
3. Generate proficiency calibrated test items (e.g. cloze tests).

As a means of evaluating models of *contextual lexical complexity* built from look-up data, we introduce the task of *dictionary look-up prediction*; that is, the task of predicting which words a learner will look up when reading a word in context.

We also introduce the first widely available *Dictionary Look-Up* dataset (DLU). The source of the DLU data is the publicly accessible *Read&Improve* platform,[1] where second language (L2) learners engage in a reading-summarisation task. This dataset captures the words that learners looked up whilst reading a passage of text. Look-ups are recorded within their original context, and metadata regarding the learner is also recorded (specifically, their first language L1, and their estimated language proficiency on the CEFR[2] scale).

With this paper, we release a development portion of this data, DLU-dev, consisting of over 16,000 content word tokens and 630 clicks. The full dataset consists of >260,000 content word tokens and >8,800 lookup events, and is expected to be released for an open participation shared task.

To evaluate the generalisability of *contextual lexical complexity* models built from the DLU dataset, we present a second dataset, the *Dialogue Dictionary Look-Up* dataset (D-DLU). This dataset is sourced from a language learning system that allows learners to look up words in generated chatbot responses (Tyen et al., 2024).

Compared to existing complexity-oriented datasets, such as complex word identification and eye-tracking data (e.g. Paetzold and Specia, 2016; Yimam et al., 2017; Shardlow, 2013; Shardlow et al., 2020; Berzak et al., 2022), our datasets have the following benefits:

---

[*] Now at Google DeepMind.

[1] `https://readandimprove.englishlanguageitutoring.com/`

[2] The Common European Framework of Reference for Languages defines levels of language competence (CoE, 2020).

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 481–501
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

1. **High external validity:** It provides behavioural patterns of L2 learners engaged in a naturalistic language learning task.
2. **Document-level context:** It captures lookup events that are interdependent across the length of the document.
3. **Learner meta-data:** It provides estimated CEFR levels for all learners and first language (L1) for ∼33% of learners.

Dictionary Look-up Prediction poses significant challenges for NLP models. Dictionary lookups are sparse events that vary widely based on context and individual knowledge, and are thus extremely difficult to predict. In response to these challenges, we argue that F_2 and ROC-AUC are appropriate evaluation metrics that reflect how useful a look-up prediction model would be for personalised learning applications.

Formulating Look-up Prediction as a sequence-to-sequence task, we evaluate fine-tuned LLaMA, Gemma, and Longformer models (Touvron et al., 2023; Gemma Team et al., 2024; Beltagy et al., 2020). During fine-tuning, in addition to a standard cross-entropy loss function, we also investigate the ROC* loss function that more directly targets the ROC-AUC (Yan et al., 2003). We conjecture that this is a more appropriate metric (see Section 5), and we find evidence that, in certain conditions, this ROC* function increases performance.

The goal of our research is to assess the suitability of different models for evaluating and aiding learners of English. Our contributions are as follows:

1. We release to the research community **two datasets**: a) DLU-dev, a dataset of >16,000 content word tokens and 630 clicks; and b) D-DLU, a set of 51 chatbot dialogues containing 43,000 content word tokens and 72 clicks.
2. We propose a new NLP task, **Dictionary Look-up Prediction**, and present a number of results for a sequence-to-sequence approach to this task.
3. We are the **first to apply ROC* loss to an NLP task**, and find that for certain cases it seems to outperform BCE loss.[3]
4. We argue that **fine-tuned LLMs are unable to satisfactorily capture contextual lexical complexity**. Not only do fine-tuned LLMs fail to outperform a feature-based ensemble model, but they also fail to generalise to other related tasks, such as Complex Word Identification (CWI) (see Section 2).

We release our data at `https://englishlanguageitutoring.com/`.

2 Related Work

The literature on word complexity includes contributions from not only NLP but also psycholinguistics and education (e.g. Bulté and Housen, 2012). This section focuses on data contributions.

CWI and LCP: Complex Word Identification (CWI) and Lexical Complexity Prediction (LCP) are two tasks in which the complexity of a word is predicted, either in the form of a binary label (CWI) or a continuous value (LCP). Both CWI and LCP have been explored in a series of shared tasks and other publications (Paetzold and Specia, 2016; Yimam et al., 2017; Shardlow, 2013; Shardlow et al., 2020, 2021; Gooding and Kochmar, 2018; Zaharia et al., 2022). Neural approaches are prevalent, but contemporary LLMs (such as GPT-4o, OpenAI et al., 2024) exhibit little to no benefit over smaller transformer models, such as $\text{RoBERTa}_{\text{large}}$ (see Smădu et al., 2024).

Existing CWI/LCP datasets have a number of shortcomings: They rely on annotators, which are often proficient L1 speakers of the language (but see Yimam et al., 2017, for the use of L2 speakers). Furthermore, the datasets typically operate on the word-in-sentence level; but a word that is difficult at the beginning of a document might be easy towards the end, when more context has been established. Thus, CWI and LCP datasets are unlikely to reflect the specific challenges L2 learners face when engaging in natural reading exercises.

Among the CWI and LCP datasets, the one for the 2018 shared task (Yimam et al., 2017, 2018) is especially interesting, as one of its sources was WikiNews (in addition to other news sources and Wikipedia), which forms also the basis for DLU. We use this dataset for further evaluation in Section 8.

Eye-Tracking Datasets: Eye-tracking is another way of approximating perceived word difficulty. Similar to the DLU dataset, eye-tracking datasets are often (but not always) created using reading tasks (Cop et al., 2017; Luke and Christianson, 2018; Hollenstein et al., 2018, 2020; Schmidtke et al., 2021; Berzak et al., 2022).

Compared to DLU, however, eye-tracking datasets are typically less naturalistic because the

---

[3] The performance difference is not statistically significant.

participants are not engaged in the task for the sake of learning, as is the case for our data, but are recruited for the experiment.

Additionally, eye-tracking datasets do not capture definition-seeking behaviour, but rather, a wide variety of cognitive processes. Definition-seeking is a behaviour aimed distinctly at lexical information, while the causes of eye movement are diverse and therefore harder to interpret.

Word Lists: Researchers and education specialists have created various word lists graded for difficulty (Negishi et al., 2013; Capel, 2015; Volodina et al., 2016; Flor et al., 2024). For example, the *English Vocabulary Profile* (Capel, 2015) provides CEFR levels for many definitions (CoE, 2020).

Compared to complexity-graded word lists, our data is specific to words in context.[4] Such contextualisation is useful when selecting e.g. reading tasks at an appropriate level for a learner.

Word lists cover only a limited vocabulary and struggle when new senses for a word are introduced. Furthermore, existing word lists describe a generic level of difficulty, and do not reflect e.g. how different L1s influence how challenging words in an L2 are. As our dataset provides L1 information when available, it helps address this gap.

Semantic Error Prediction: Since it is based on a reading and summarisation task, our dataset is specifically focused on complexity in comprehension. A comparable *production-side* dataset is the Semantic Error Prediction dataset by Strohmaier and Buttery (2024), which provides information on which content words learners fail to produce when writing essays. Like DLU, the SEP dataset is also based upon behavioural data from L2 learners engaged in a naturalistic learning task. That being said, we can expect differences to exist between production and comprehension, because in the case of production, learners have a (rough) meaning in their mind and have to retrieve correct word forms, while in the case of comprehension, the form is given by the text and learners have to access the correct meaning for it (cf. Jiang, 2000). We use the SEP dataset for evaluation in Section 8.

3 Description of the DLU Dataset

This section describes DLU's main features.

3.1 Data Source

We use the *Read&Improve* (R&I) platform as our data source. This platform allows L2 learners to engage in the task of reading and summarising an article to improve their English (see Figure 2 in Appendix A for a screenshot of the platform interface). Upon submission of the summary, they receive automated feedback. During reading, learners can click on words to retrieve definitions and examples.

The texts used for this task are taken from WikiNews (available under a Creative Commons license). Different users might be presented with the same article to summarise.

3.2 Scope of DLU

Our data shows which content words in a seen document have been clicked on to retrieve dictionary information. That is, for each content word token of a document, the data specifies whether it has been clicked on or not by the user. Tokens are considered content words if they have been tagged as adjectives, adverbs, nouns, or verbs by the RASP pipeline used by R&I (Briscoe et al., 2006).

To ensure that the learner has seen all tokens, only documents for which they have successfully submitted a summary are included. We also exclude data from users who have clicked less than five times in total, as this might indicate that they are unfamiliar with the functionality of clicking words to look up their dictionary information, or that they are so proficient as to never require dictionary information.

3.3 Data Selection

Overall, our dataset includes more than 260,000 seen content word tokens, with more than 8,800 clicks on these tokens. We split the DLU dataset into three parts by document: a train split, a dev split, and a test split, where the dev and test splits contain slightly more than 10% of unique documents. More information on the size of the dataset and its splits can be found in Table 1.

The dev-split (DLU-dev) is released with this paper, while the train and test splits are reserved for a future shared task open to public participation. All data will be released upon completion of the shared task.

3.4 User Information

Users likely differ in their look-up patterns both idiosyncratically and systematically based on their:

---

[4] For an application of word lists to contextualised uses, see Aleksandrova and Pouliot (2023).

split	tot. docs	uniq. docs	users	clicks	con. tokens
all	1327	221	663	8858	266011
train	1143	176	616	7822	235786
dev	101	21	90	630	16084
test	83	24	68	406	14141

Table 1: Description of data and splits, including the number of content tokens. Multiple users might see the same document, therefore the number of total documents can diverge from that of unique documents.

	A2	B1	B2	C1	C2	UNK	sum
all	135	198	126	34	1	169	663
train	123	185	117	33	1	157	616
dev	21	32	17	6	0	14	90
test	13	22	15	5	0	13	68

Table 2: Essay-based estimation of user CEFR levels.

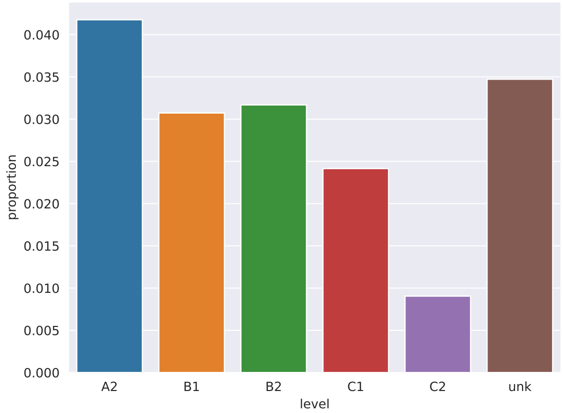


Figure 1: Proportion of content words that were looked up, for each CEFR-level as estimated using W&I essays.

- first language (L1)
- language ability as estimated CEFR level

For the wide range of L1s in DLU-dev, see Table 10 in the appendix. The language with most users is Spanish (93) followed by Italian and Turkish (both 17). For some languages (e.g. Serbian, Hindi), data is only available for a single user.

Our datasets include two estimates of the learner CEFR level. One estimate is based on submissions to the associated essay writing platform *Write&Improve* (W&I)[5] and described in Table 2, while the other relies on self-reports. While the self-reports have full coverage (see Table 6), the essay data are more comparable across users, as it is based on the same automatic grading system. We therefore only use essay-based estimates of CEFR-levels in our experiments, even though both are included in our data release.

While the automatic scores are likely imperfect, we believe that they provide a reasonable approximation of the learner proficiency because they correlate with look-up propensity (see Figure 1). With only one exception, learners at higher levels tend to look up a smaller proportion of word tokens.

3.5 Noise and Uncertainty

Look-up events are affected by many idiosyncratic features, not all of which are captured by our dataset. Notably, how often a learner has previously encountered a word will strongly affect whether they look it up.[6] As a result, our dataset leads to high aleatoric uncertainty for the models trained on it (Hüllermeier and Waegeman, 2021).

For many applications, however, perfect prediction of look-ups are not required. Rather, the main goal is to separate words that are difficult enough to require a definition, from words that are easier. With this separation, text readability can be improved and vocabulary test items can be created.

4 Description of Chatbot Dataset

The chatbot dialogue dataset (D-DLU) is derived from an earlier experiment by Tyen et al. (2024) using BlenderBot v1 (Roller et al., 2021). It consists of two types of dialogue data:

1. A reading condition (D-read), where participants read self-chats between only the bot.
2. A chat condition (D-chat), in which participants chat with the bot.

We filtered this dataset manually to remove chats containing unsafe texts such as insults or inappropriate topics, and instances in which the chatbot behaved erratically, e.g. when the chatbot defined similar words repeatedly. As a result, we ended up with a set of 51 chats from the original 80.

A closer look at the data distribution (Table 13) reveals that the remaining 25 dialogues in D-chat only contained 5 look-up events. This number is too low for informative evaluation. While we release both portions of the dataset, we recommend only using the D-read split for lookup-prediction.

---

[5] `https://writeandimprove.com/`. For W&I essay data, see Nicholls et al. (2024).

[6] Research on language acquisition suggests that many encounters are needed for even passive knowledge of vocabulary. See Hargis et al. (1988) for some estimates of required repetitions. See Peters and Webb (2018) for how encounters during TV consumption lead to vocabulary acquisition.

5 Evaluation Metrics for DLU

Considering data sparsity, noise, and intended application areas, we argue that F_2 and ROC-AUC are most appropriate evaluation metrics for DLU.

5.1 F_2 and adaptive F_2 Metric

Unlike F_1, F_2 prioritises recall over precision:

$$F_\beta = (1+\beta^2) \cdot \frac{\text{Precision}\cdot\text{Recall}}{\beta^2\cdot\text{Precision}+\text{Recall}},\ \beta = 2$$

That is, we accept more false positives to avoid false negatives. In our case, false positives are predictions of look-up events where no such event has occurred. However, we expect that learners do not click on every difficult word, as browsing dictionary information disrupts the flow of reading.

Furthermore, for many use cases it is acceptable to wrongly predict that a few words are difficult. For example, if the goal is to select words for a cloze test, then adding a small portion of comparatively easy gap words to the test will not create a problem.

Look-up events (and therefore positive labels) are rare, which might render a decision threshold of 0.5 too strict. We are instead looking for a metric that is robust to the high sparsity of DLU and provides more general information about whether the models are able to separate words that are likely to be looked up from those that are unlikely to be looked up.

Therefore, we explore using an adaptive threshold for the F_2 metric. That is, we estimate which value in the inclusive range 0.01 – 0.99 (step size 0.01) leads to the highest F_2 score on the training data and then use the same threshold on the evaluation data to calculate an adaptive F_2 (aF_2).

5.2 ROC-AUC Metric

Compared to the F_2 metric, the ROC-AUC provides more direct information on whether the models distinguish easy from difficult words. The area under the curve provides this information, because it can be interpreted as the probability that the model ranks a randomly chosen positive example higher than a randomly chosen negative example (Fawcett, 2006, p. 868).

For many applications, using the raw scores rather than the binary classification is useful. Consider the case of creating a number of test items for the most difficult words in a text. In such a case, we are not committed to any particular threshold at which a score indicates that the token would be looked up in a dictionary. In contrast to the F-metrics, the AUC does not rely on any such threshold. Thus, we believe that the AUC metric is well suited for our purposes. In Section 6.2, we describe the ROC* loss function, which targets the AUC.

6 DLU Models

This section describes 3 types of DLU models.

6.1 Feature-based Baselines

Frequency Baseline First, we provide a baseline based on word frequencies using the `wordfreq` package (Speer, 2022).[7] We use the Zipf frequency estimate provided by `wordfreq`, which consists of a value within the 0 to 10 range.[8] We rescale this value to be between 0 and 1. All together, the score for a token is calculated as follows:

$$\text{score}(word) = 1 - \frac{\log_{10}(\text{proportion}(word))+9}{10}$$

Ensemble Baseline We also explore a more complicated feature-based ensemble model using scikit-learn (Pedregosa et al., 2011), consisting of six classifiers combined in a soft voting ensemble. The features of this model include the frequency baseline and a variety of features pertaining to the word lemma and the word token's position in the text. The exact features are listed in the appendix Appendix E. Due to reliance on the W&I-estimated CEFR-level of the learner, we do not report results of this baseline on D-read, which does not have this feature.

6.2 Finetuned Models

We fine-tune two models on the DLU train split, both from Huggingface transformers (Wolf et al. 2020; see also Appendix F):

1. Longformer (Beltagy et al., 2020), which we choose over other BERT-like models because we operate on the document-level;
2. LLaMA 3.2 (Touvron et al., 2023), for which we choose the 1B parameter version due to compute considerations.

Loss Functions We explore two loss functions. As is common for binary classification tasks, we use the Binary Cross Entropy (BCE) loss as the basis

---

[7] `https://github.com/rspeer/wordfreq`, which is based on the ExquisiteCorpus (`https://github.com/LuminosoInsight/exquisite-corpus`).

[8] While this calculation can reach 10, due to the distribution of words, the effective range tends to be between 0 and 8. The package also uses 0 as the default value of words not found in the word list, even though 0 does not correspond to zero occurrences due to the Zipfian transformation.

for our first loss function. To adjust for the label imbalance, we use a weight for positive cases (w_p). Thus our BCE loss takes the following form:

$$l_{\text{BCE}} = -\left(w_p y \log\left(\sigma(x)\right) + (1-y)\log\left(1-\sigma(x)\right)\right)$$

We treat the weight as a hyperparameter to be decided through search, but the search space is biased towards higher values as positive cases are under-represented (see Section F.1).

The second loss function, called ROC*, targets the ROC-AUC directly. This function was developed by Yan et al. (2003) and is based on the equivalence between the ROC and the Wilcoxon-Mann-Whitney statistic. We explore this loss-function because we take the correct ranking of words as measured by ROC-AUC to be an excellent metric reflecting probable use-cases (see Section 5.2).

Let **N** be the set of scores for non-clicked content word tokens and **C** the set of scores for clicked content word tokens. As the loss function compares between pairs of these two sets, it is useful to introduce their product: $\mathbf{P} = \mathbf{N} \times \mathbf{C}$

The loss takes the following form (batching is ignored here for illustration):

$$l_{\text{ROC*}} = \frac{1}{|\mathbf{P}|}\sum_{(x,y)\in\mathbf{P}} \begin{cases} (x-y+\gamma)^2 & : x+\gamma > y \\ 0 & : \text{otherwise} \end{cases}$$

where $\gamma > 0$ is a hyperparameter ensuring that a sufficiently large distance exists between clicked and non-clicked cases.[9] Thus, we allow for mini-batch training by storing previously seen scores for content word tokens and sampling them for comparison against scores calculated in the mini-batch. The size of the samples is treated as a hyperparameter.

Hyperparameter Search We perform a 20-trial hyperparameter search using Optuna (Akiba et al., 2019) maximizing ROC-AUC, training on the train split and evaluating on the dev split. The selected hyperparameters are in Section F.1.

Data Processing To account for L1 and CEFR level, we add special tokens for them to the model and append them at the start of each document. While the model will see the same document multiple times with different look-up patterns during training, these will often differ in either indicated L1 or CEFR level. For adding the special token, we merge CEFR levels such that B1 and B2 are represented as B, and so forth (see Table 7). This addresses the problem of having relatively few cases for some CEFR levels. Under-represented L1s are merged into the "unknown" category.

We evaluate also on the chatbot dataset split *D-read*. However, the length of 7 chats in particular pose a problem as the Longformer model we use is limited to 4096 subtokens. To circumvent this, we split longer dialogues after reaching this threshold, which might affect performance on D-read.

Significance Tests We perform permutation significance tests to see if ROC* trained models achieve higher AUC compared to BCE trained models. With 0.05 as the starting p-value, the Bonferroni-corrected threshold for this paper is 0.0027. To avoid further lowering of the threshold, we only perform tests for the aggregated DLU test split.

6.3 Prompting Models

For comparison, we also prompted LLMs on the dictionary look-up task, specifically the instruction-tuned versions of Gemma and LLaMA (Gemma Team et al., 2024; Touvron et al., 2023) (Appendix F). We use both zeroshot and fewshot prompting, as described in Appendix G, except for D-read where we only use zeroshot prompting due to the challenging document length.

Our prompts return complex words from the text. To address cases in which word types occur more than once in the text, we explored two approaches: 1) Predicting only the first occurrence to be looked-up and 2) predicting look-ups for all occurrences. We focus on the first option, as learners usually only need to look up a word once, and we report the results of the second approach in the appendix. The overall picture is not affected by this choice.

As our prompting models output binary results, there is no changeable threshold for an adaptive F_2. Similarly, the AUC is less useful as the score for each word token is 0 or 1.

7 DLU Results

The results of the baselines as well as the transformer models on the DLU test split can be found in Table 3. We report the results for the coarse-grained CEFR-levels (A, B, C, and unknown) separately. Because we release only the dev split of DLU with this paper, we also report the results on this split in the appendix (see Table 22).

Due to the sparsity of look-up events and because the hyperparameter search targets the AUC, some model settings lead to F_1 and F_2 values of 0. The aF_2 consistently takes a value above 0, but

---

[9] Our implementation follows the public ROC* repository (`https://github.com/iridiumblue/ROC*`). However, similar to Yan et al. (2003), we keep γ as a hyperparameter, instead of deriving it.

		A				B				C				unk				All				D-read			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	10.2	14.3	-	55.9	14.2	17.2	-	57.1	**15.0**	**22.3**	-	64.4	11.1	16.1	-	58.3	12.8	16.8	-	57.5	2.2	4.6	-	57.4
	fewshot	9.7	14.3	-	55.9	13.4	17.2	-	57.1	13.8	22.3	-	65.8	12.3	18.2	-	60.0	12.4	17.1	-	57.8	-	-	-	-
LLaMA-Inst.	zeroshot	10.1	17.8	-	59.6	8.3	12.4	-	53.4	6.5	11.6	-	57.8	9.5	16.5	-	60.1	8.8	14.2	-	55.8	1.0	2.4	-	56.1
	fewshot	10.3	16.7	-	58.0	9.0	12.5	-	53.8	6.5	11.7	-	57.9	6.0	10.0	-	53.6	8.5	12.9	-	54.8	-	-	-	-
LLaMA	ROC*	0.0	0.0	7.2	75.7	0.0	0.0	8.2	65.9	0.0	0.0	0.0	62.1	0.0	0.0	11.2	68.1	0.0	0.0	7.9	67.8	0.0	0.0	3.0	76.7
	BCE	11.0	22.2	21.9	72.7	12.4	21.8	19.4	64.9	5.6	11.8	5.8	63.0	8.0	16.7	19.7	69.4	10.4	20.0	18.9	66.2	2.7	5.6	**5.2**	77.1
Longformer	ROC*	14.4	25.6	18.6	78.5	16.0	25.5	21.7	72.4	5.7	11.0	13.4	64.9	12.1	20.7	18.7	77.6	14.0	23.6	20.0	73.9	**2.7**	**5.8**	2.8	83.4
	BCE	0.0	0.0	12.3	70.9	0.0	0.0	13.2	70.7	0.0	0.0	3.4	60.5	0.0	0.0	8.4	76.6	0.0	0.0	11.6	71.1	0.0	0.0	3.9	74.6
Baseline	freq.	8.7	18.9	24.7	75.8	9.6	20.6	23.1	71.4	4.2	9.9	10.8	72.3	5.7	12.9	16.6	72.2	8.1	17.7	21.0	72.5	0.9	2.2	3.3	**84.9**
	ens.	**22.2**	**32.8**	**31.6**	**85.9**	**17.3**	**26.1**	**28.0**	**76.2**	12.3	18.8	**19.6**	**81.9**	**14.2**	**24.3**	**23.6**	**80.4**	**17.3**	**26.5**	**27.4**	**79.2**	-	-	-	-

Table 3: Results on the DLU test split. "aF_2" stands for F_2 with a adaptive threshold, as discussed in Section 5.

because the threshold is estimated only on the training data and only for the entire dataset (i.e. not separately for each CEFR level), the aF_2 is sometimes lower than the F_2. The impact of the adaptive threshold is discussed in Appendix J and does not bear on the general conclusions.

The best finetuned models outperformed the prompt-based models with the exception of the C split of data. This split contains data from learners at the C1 and C2 CEFR-levels, who rarely look up words (24 in the test split, see Table 9).

The results show that the frequency baseline is strong, often outperforming other models. The ensemble baseline is even stronger, outperforming all other models convincingly with only minor exception on the C split. Some of the differences are substantial, e.g. the ensemble baseline achieves an 79.2% AUC on the overall DLU test split, with the next best model reaching only 73.9%.

Among the transformer models, the Longformer ROC* model performs best on the test split. We note, however, that these results do not directly translate to the dev split of DLU (see Table 22), suggesting some overfitting. No difference in AUC scores between ROC* and BCE model is statistically significant, although the ROC* versions consistently perform better.

The results on the D-read split in D-DLU described in Section 4 are also included in Table 3. As is to be expected for a different data source with a different distribution, performance is lower. The highest F_2 (5.8%) and aF_2 (5.2%) are achieved by the ROC*-Longformer and BCE-LLaMA model respectively. The frequency baseline achieves the highest AUC (84.9%), follow by the ROC*-Longformer (83.4%).

As described in Section 3, multiple users might interact with the same document, leading to different look-up events. To account for any effects this might have, we also evaluated on a filtered version of our dataset so that each document was unique. See Appendix K, Table 25, and Table 24 for the results, which show the same overall picture.

8 Evaluation on CWI/SEP

To investigate the degree to which our DLU dataset captures word difficulty information that is specific to the dataset's construction, including the underlying reading task, we ask the following question: Can models transfer DLU knowledge to other tasks that also attempt to track word complexity? We address this question by performing additional experiments using the CWI and SEP datasets.

We chose the CWI task because it is the most widely explored binary task targeting word complexity. However, CWI datasets are usually not as sparse, often do not provide information on the document-level, and frequently rely on annotators proficient in the language, rather than learners.

We also evaluate on the SEP dataset because it not only targets word complexity, but provides highly sparse binary learner behaviour data for longer contexts[10]; as is the case for DLU prediction. Furthermore, investigating the transfer to SEP addresses the question of whether learners struggle to produce the words that they find difficult enough to look up during a reading task. We can thus provide evidence for how similar comprehension difficulty and production difficulty are.

8.1 Experimental Setup

We train all DLU-finetuned models an additional time on the CWI and SEP datasets by Yimam et al. (2017) and Strohmaier and Buttery (2024). For comparison, we finetune the base models on the

---

[10] The SEP dataset is standardly formatted to chunks of one or more paragraphs.

CWI and the SEP task without using DLU data, and provide the frequency baseline. The experimental procedure follows the same pattern as described in Section 6, i.e. an initial hyperparameter search followed by evaluation on the dataset.

Significance Tests We use permutation significance tests to see if the models finetuned on both DLU and CWI or SEP perform better than models only finetuned on the latter. We perform these tests for the F_1 and AUC metrics because the F_1 was used in previous work and the AUC was targeted by the hyperparameter search.[11]

8.2 CWI/SEP Results

We present the CWI results split by data source (News, Wikipedia, and WikiNews) and in aggregate. For the SEP dataset, we offer the same split by CEFR level as for DLU.

The CWI results (see Table 4) suggest that the BCE-Longformer architecture is best suited for this task when considering F-scores and AUC. The DLU-finetuned version of the BCE-Longformer model produces the highest AUC (85.6%) and the base model the highest F_1 (78.5%), but only the comparison of the F_1 is statistically significant. The only other result significant at the 0.0027 threshold was the difference between the F_1 of the base ROC*-LLaMA (74.3%) and the DLU-finetuned model (71.1%), which favours the base model.

The SEP results (reported in Table 5) clearly suggest a strongest model on the F-score metrics: the BCE-Longformer model finetuned only on SEP. Among the 6 significant results (see Table 20), only the difference between the F_1-scores of the DLU-finetuned BCE-LLaMA (7.3%) and the CWI-only version (4.2%) points towards positive transfer, the rest pointing in the opposite direction. The overall best F_1 (11.9%) and AUC (71.0%) slightly outperform the numbers (11% and 69.8%) previously reported (Strohmaier and Buttery, 2024).

9 Discussion

Our results show that **look-up prediction is a challenging, but addressable task**. Finetuned transformer models outperform a frequency baseline, but fail to beat a feature-based ensemble model.

Similar to the CWI results reported by Smădu et al. (2024), we find that **model size is not the decisive variable**: among the finetuned models, the ROC*-Longformer model outperformed the LLaMA models, even though the latter has considerably more parameters ($\sim$149M vs 1B). Similarly, the prompting models were considerably larger than the finetuned models and yet performed worse.

The ROC* loss which we explored following Yan et al. (2003) performed well on DLU for its target metric, the ROC-AUC. Among the finetuned neural models, the highest AUC is always produced by a model using ROC* loss. Thus, we suggest that **the ROC* loss function is of value for tasks in which the AUC is the correct metric**. That being said, neither on the DLU-dev split nor the aggregated CWI data is the highest AUC achieved by a ROC* model. The improvements, thus, appear dependent on the specific data distribution.

The simple frequency baseline proved competitive on all considered tasks. Even more impressive was the performance of the feature-based ensemble model on DLU, which showed a leading performance. Even compared to LLMs, **feature-based baselines remain very competitive in the field of word complexity**. On the combined data of DLU, the highest scores on all four considered metrics were achieved by the ensemble baseline. In the case of the English Wikipedia split of the CWI data (Yimam et al., 2017), the F_1 of the simple frequency baseline (73.9%) outperformed every one of the 14 few-shot prompting models reported by Smădu et al. (2024), where the best one only achieved 70.6%.

These strong baseline results and irrelevance of model size suggests to us that **modelling difficulty in L2 vocabulary acquisition is not solved by existing NLP methods**. We believe that further personalisation is required to move forward, and we see DLU as a major step in this direction.

Furthermore, models will have to be more specifically adapted to the high variance between learners. Providing information about proficiency level and first language as special tokens proved insufficient. To account for the variance between learners, it might be necessary to adapt the training procedure or architecture details of the model even further to information about the learner. With more personalised DLU data, it might, for example, be possible to train layers specialised for certain L1s, CEFR levels, or other background data.

Looking at the CWI and SEP experiments, the significant results do not support that knowledge from finetuning on DLU is transferred to other

---

[11] We do not run tests for other metrics as this would increase the number of significance tests, decreasing the Bonferroni-corrected threshold further.

		N					W					WN					All				
		Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC
f.-LLaMA	ROC*	76.7	73.7	72.4	81.7	84.2	68.8	69.1	64.2	75.0	78.7	72.0	68.2	65.9	76.7	79.4	73.7	71.1	68.6	78.7	81.7
	BCE	73.6	75.4	83.1	82.9	82.3	69.7	75.4	78.9	76.5	73.2	68.8	70.5	75.7	74.6	74.9	71.4	74.0	80.0	79.0	78.2
f.-Longformer	ROC*	**79.7**	78.7	80.8	86.6	87.7	72.4	74.9	73.4	84.6	80.7	76.4	74.9	75.3	82.0	84.0	77.2	76.8	77.5	84.8	85.3
	BCE	72.3	76.2	**87.6**	**88.2**	**87.9**	68.8	77.8	**87.7**	**87.2**	81.0	67.3	73.0	**84.5**	**84.1**	84.3	70.1	75.6	**86.7**	**86.8**	**85.6**
LLaMA	ROC*	77.4	77.0	80.4	81.9	84.2	69.4	71.8	69.9	73.3	76.5	72.4	71.7	73.6	75.0	78.3	74.3	74.3	76.0	77.9	81.0
	BCE	73.9	76.3	85.0	78.6	83.5	68.9	76.3	82.9	70.1	74.3	66.9	70.9	79.3	68.4	76.9	70.8	74.7	82.9	73.7	79.9
Longformer	ROC*	79.4	79.1	82.8	85.9	87.3	73.2	75.8	74.6	84.2	**81.5**	**77.2**	76.2	77.4	83.1	**84.4**	**77.4**	77.5	79.3	84.7	85.3
	BCE	77.8	**79.1**	86.7	85.4	87.6	**74.6**	**79.4**	83.3	78.0	81.1	74.9	**76.6**	83.0	80.3	83.6	76.3	**78.5**	84.8	82.2	85.1
Baseline	freq.	62.2	65.6	73.0	80.7	67.5	66.1	73.9	80.0	86.8	67.3	61.7	67.8	77.7	81.4	66.9	62.9	68.2	76.0	82.3	67.6

Table 4: Prediction results on the 2018 CWI dataset (Yimam et al., 2017). Models with the prefix “f.-” for “finetuned” have first been finetuned on DLU. “aF_2” stands for F_2 with a adaptive threshold, see Section 5.

		A					B					C					N					All				
		Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC	Acc.	F_1	F_2	aF_2	AUC
f.-LLaMA	ROC*	96.3	0.7	0.4	15.3	58.1	97.1	0.0	0.0	15.6	63.9	98.2	1.2	0.8	9.2	59.1	99.4	0.0	0.0	2.9	53.7	97.8	0.5	0.3	11.3	59.4
	BCE	88.7	9.2	12.5	13.3	62.2	90.8	10.5	14.6	16.0	65.7	91.9	5.7	9.2	8.7	63.8	93.6	1.2	2.5	2.0	53.4	91.4	7.3	11.1	11.5	63.8
f.-Longformer	ROC*	**96.5**	0.0	0.0	7.4	**71.1**	**97.2**	0.0	0.0	8.9	**73.4**	**98.4**	0.0	0.0	7.7	**69.7**	**99.4**	0.0	0.0	0.0	58.5	**98.0**	0.0	0.0	7.4	**71.0**
	BCE	**96.5**	0.0	0.0	15.2	51.1	**97.2**	0.0	0.0	12.4	54.2	**98.4**	0.0	0.0	7.5	53.0	**99.4**	0.0	0.0	2.7	54.9	**98.0**	0.0	0.0	9.4	52.0
LLaMA	ROC*	92.0	8.4	9.6	9.7	59.8	93.4	10.4	12.3	12.1	65.7	95.1	5.6	7.3	6.8	63.1	96.5	1.1	1.9	2.0	53.4	94.4	7.4	9.2	9.1	63.4
	BCE	95.3	3.2	2.5	1.8	63.3	96.3	7.4	6.0	3.4	66.2	97.7	1.9	1.6	0.8	63.0	98.9	0.0	0.0	0.0	52.6	97.1	4.2	3.4	2.1	64.7
Longformer	ROC*	93.3	11.6	12.3	2.2	67.8	95.2	**16.3**	16.6	7.2	72.0	96.4	**11.0**	**12.6**	2.5	67.8	97.7	**2.6**	**3.7**	0.0	**61.0**	95.8	11.8	13.0	3.9	70.1
	BCE	90.0	**13.9**	**18.4**	**19.8**	67.8	92.8	14.2	**17.9**	**19.6**	71.1	95.0	9.4	12.5	**14.9**	68.4	97.6	0.8	1.2	1.7	55.6	94.1	**11.9**	**15.6**	**17.0**	70.5
Baseline	freq.	61.3	7.6	15.2	14.4	54.1	59.6	6.5	13.6	12.9	56.3	55.6	3.7	8.3	8.5	54.5	50.4	1.2	2.9	**3.2**	54.7	56.5	4.4	9.7	9.5	53.3

Table 5: Results on the Semantic Error Prediction (SEP) dataset (Strohmaier and Buttery, 2024). Models with the prefix “f.-” have first been finetuned on DLU. “aF_2” stands for F_2 with a adaptive threshold, see Section 5.

tasks. Based on this observation, we conjecture that **different approaches that all ostensibly concern word complexity, in fact track different phenomena**. In particular, for the CWI task the significant results point in the direction of negative transfer between DLU and CWI. We take this to show that the data distribution diverge too strongly.

Part of the distributional differences are, without doubt, the sparse nature of DLU and the shorter length of the CWI texts. Another difference, however, is that the CWI data we used was derived from *proficient* speakers of English rather than *learners*. Because DLU directly records dictionary usage during a naturalistic learning task, it has higher external validity. Hence, we speculate that the CWI data do not sufficiently reflect which words L2 learners of English struggle with.

In contrast to CWI, we found at least one significant result on the aggregated SEP dataset pointing in the direction of improvement, although with five other significant result pointing in the opposite direction. Like DLU, SEP is derived from learner behaviour in a naturalistic task. However, DLU targets difficulties in comprehending words, while SEP targets production difficulties, which may explain the differences.

10 Conclusion

We introduce the dictionary look-up task, which provides insight into word complexity for the purpose of supporting personalised learning technologies. We release the *Dictionary Look-Up development* (DLU-dev) dataset. Additionally, we release a look-up dataset of chatbot dialogues (D-DLU) for evaluation. We provide results from zero- and few-shot prompting as well as fine-tuning.

Investigating the transfer from DLU to other tasks such as complex word identification (CWI) and semantic error prediction (SEP), we find that DLU and CWI appear to track different phenomena. For SEP, we set new state-of-the-art results, but find mixed to negative results on transfer.

The leading performance of a feature-based model on DLU strongly suggests that further research is required to adequately incorporate information about individual learners and their lexical acquisition into neural models of word complexity. The release of DLU-dev is a major step toward achieving this goal. We release our data at `https://englishlanguageitutoring.com/`.

Acknowledgments

We thank Andrew Caines, Øistein Andersen, Paul Ricketts, Jon Phillips, Scott Thomas, and other members of ALTA and CUP&A for their support. Furthermore, we thank the anonymous reviewers for their comments.

This research is supported by Cambridge University Press & Assessment.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: a next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining*.

Desislava Aleksandrova and Vincent Pouliot. 2023. Cefr-based contextual lexical complexity classifier in english and french. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, page 518–527, Toronto, Canada. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. (arXiv:2004.05150). ArXiv:2004.05150 [cs].

Yevgeni Berzak, Chie Nakamura, Amelia Smith, Emily Weng, Boris Katz, Suzanne Flynn, and Roger Levy. 2022. Celer: A 365-participant corpus of eye movements in l1 and l2 english reading. *Open Mind*, 6:41–50.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the rasp system. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, page 77–80, Sydney, Australia. Association for Computational Linguistics.

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

Bram Bulté and Alex Housen. 2012. Defining and operationalising L2 complexity. In Alex Housen, Folkert Kuiken, and Ineke Vedder, editors, *Dimensions of L2 Performance and Proficiency: Complexity, Accuracy and Fluency in SLA*, Language Learning & Language Teaching, pages 21–46. John Benjamins Publishing Company.

Annette Capel. 2015. The english vocabulary profile. *English profile in practice*, 5(1):9–27.

Council of Europe CoE. 2020. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment; Companion Volume*. Council of Europe Publishing, Strasbourg.

Uschi Cop, Nicolas Dirix, Denis Drieghe, and Wouter Duyck. 2017. Presenting geco: An eyetracking corpus of monolingual and bilingual sentence reading. *Behavior Research Methods*, 49(2):602–615.

Tom Fawcett. 2006. An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874.

Michael Flor, Steven Holtzman, Paul Deane, and Isaac Bejar. 2024. Mapping of american english vocabulary by grade levels. *ITL - International Journal of Applied Linguistics*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas

Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *Preprint*, arXiv:2408.00118.

Sian Gooding and Ekaterina Kochmar. 2018. Camb at cwi shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, page 184–194, New Orleans, Louisiana. Association for Computational Linguistics.

Charles H. Hargis, Marge Terhaar-Yonkers, Patricia Couch Williams, and Mellissa Testerman Reed. 1988. Repetition Requirements for Word Recognition. *Journal of Reading*, 31(4):320–327.

Nora Hollenstein, Jonathan Rotsztejn, Marius Troendle, Andreas Pedroni, Ce Zhang, and Nicolas Langer. 2018. Zuco, a simultaneous eeg and eye-tracking resource for natural sentence reading. *Scientific Data*, 5(11):180291.

Nora Hollenstein, Marius Troendle, Ce Zhang, and Nicolas Langer. 2020. Zuco 2.0: A dataset of physiological recordings during natural reading and annotation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, page 138–146, Marseille, France. European Language Resources Association.

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 110(3):457–506.

Nan Jiang. 2000. Lexical representation and development in a second language. *Applied Linguistics*, 21(1):47–77.

Steven G. Luke and Kiel Christianson. 2018. The provo corpus: A large eye-tracking corpus with predictability norms. *Behavior Research Methods*, 50(2):826–833.

Masashi Negishi, Tomoko Takada, and Yukio Tono. 2013. A progress report on the development of the cefr-j. In *Exploring language frameworks: Proceedings of the ALTE kraków conference*, page 135–163. Citation Key: negishi2013progress.

Diane Nicholls, Andrew Caines, and Paula Buttery. 2024. *The Write & Improve Corpus 2024: Error-Annotated and CEFR-Labelled Essays by Learners of English*.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai

Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. GPT-4o System Card. *Preprint*, arXiv:2410.21276.

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Elke Peters and Stuart Webb. 2018. Incidental Vocabulary Acquisition Through Viewing L2 Television and Factors That Affect Learning. *Studies in Second Language Acquisition*, 40(3):551–577.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for Building an Open-Domain Chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Daniel Schmidtke, Julie Van Dyke, and Victor Kuperman. 2021. Complex: An eye-movement database of compound word reading in english. *Behavior Research Methods*, 53:59–77.

Matthew Shardlow. 2013. The CW Corpus: A New Resource for Evaluating the Identification of Complex Words. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 69–77, Sofia, Bulgaria. Association for Computational Linguistics.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — A New Corpus for Lexical Complexity Prediction from Likert Scale Data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. SemEval-2021 Task 1: Lexical Complexity Prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Răzvan-Alexandru Smădu, David-Gabriel Ion, Dumitru-Clementin Cercel, Florin Pop, and Mihaela-Claudia Cercel. 2024. Investigating large language models for complex word identification in multilingual and multidomain setups. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, page 16764–16800, Miami, Florida, USA. Association for Computational Linguistics.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

David Strohmaier and Paula Buttery. 2024. Semantic error prediction: Estimating word production complexity. *Proceedings of the 13th Workshop on Natural Language Processing for Computer Assisted Language Learning*, 13:209–225.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. *Preprint*, arXiv:2307.09288.

Gladys Tyen, Andrew Caines, and Paula Buttery. 2024. Llm Chatbots as a Language Practice Tool: A User Study. In *Swedish Language Technology Conference and NLP4CALL*, pages 235–247.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Elena Volodina, Ildikó Pilán, Lorena Llozhi, Baptiste Degryse, and Thomas François. 2016. Swellex: Second language learners' productive vocabulary. In *Proceedings of the joint workshop on NLP for Computer Assisted Language Learning and NLP for Language Acquisition*, page 76–84, Umeå, Sweden. LiU Electronic Press.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, page 38–45, Online. Association for Computational Linguistics.

Lian Yan, Robert Dodier, Michael C. Mozer, and Richard Wolniewicz. 2003. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 848–855, Washington, DC, USA. AAAI Press.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A Report on the Complex Word Identification Shared Task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. CWIG3G2 - Complex Word Identification Task across Three Text Genres and Two User Groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407, Taipei, Taiwan. Asian Federation of Natural Language Processing.

George-Eduard Zaharia, Răzvan-Alexandru Smădu, Dumitru Cercel, and Mihai Dascalu. 2022. Domain adaptation in multilingual and multi-domain monolingual settings for complex word identification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 70–80, Dublin, Ireland. Association for Computational Linguistics.

A Platform

The *Read&Improve* (R&I) platform and its sister platform *Write&Improve* are available free of charge. Users agree to the sharing of their input for research purposes. For a screenshot of the R&I user interface, see Figure 2.

B Limitations

Dictionary look-up events are rare, sparse, and noisy. While DLU includes more than 8,800 look-up events among 260,000 content word tokens, these features of look-up events inherently limit model performance and some applications. The additionally released chatbot-dialogue dataset is smaller, and therefore its usefulness is limited to evaluation.

Our data is exclusive to English language texts and the first languages of the learners who performed click actions are unevenly distributed (see Table 10). The same is true for CEFR levels. Further personalisation would require more even data distribution.

Due to compute restrictions, we focused on models with comparatively few parameters, although we do include evaluation on LLMs such as LLaMA-3.2-1B. Since we and others (Smădu et al., 2024) found that model size does not appear to predict model performance well, we believe that this restriction poses no major problems. Our focus is on using publicly available models, ensuring replicability.

C Safety and Privacy Considerations

The information in the DLU data poses few risks. While we release information about learner L1 and estimated CEFR-level, personal identification is practically impossible since this information is very broad and the lookup patterns themselves are specific to the platform.

The additional chatbot-dialogue data we release should be handled with greater care, because it includes user input and the chatbot model was not filtered for sensitive content (Tyen et al., 2024). As described above (see Section 4), we have manually filtered the dataset and removed critical personal information about the chat participants, e.g. changing first names.

D Dataset Description

For the overall description of the DLU dataset, see Section 3. Further description of CEFR levels and first languages (L1s) across the dataset can be found in tables 6 to 8 and 10 to 12.

	B2	B1	A2	C1	C2	C2+	sum
all	228	242	112	55	17	9	663
train	208	227	108	52	15	6	616
dev	29	44	11	4	1	1	90
test	26	10	11	14	4	3	68

Table 6: Self-reported CEFR levels of users.

	A	B	C	UNK	sum
all	135	324	35	169	663
train	123	302	34	157	616
dev	21	49	6	14	90
test	13	37	5	13	68

Table 7: CEFR levels for users as estimated by essays from W&I.

	A	B	C	UNK	sum
all	270	669	116	272	1327
train	229	577	97	240	1143
dev	23	53	8	17	101
test	18	39	11	15	83

Table 8: CEFR levels as estimated by essays from W&I across documents by users (i.e. some users and WikiNews articles appear more than once in this table).

D.1 Format of the Data

The data is formatted as a document-level token-classification task. Tokenisation follows the RASP pipeline used by R&I (Briscoe et al., 2006) For each token a label is provided, with the default label -100 used for non-content word tokens.

Example

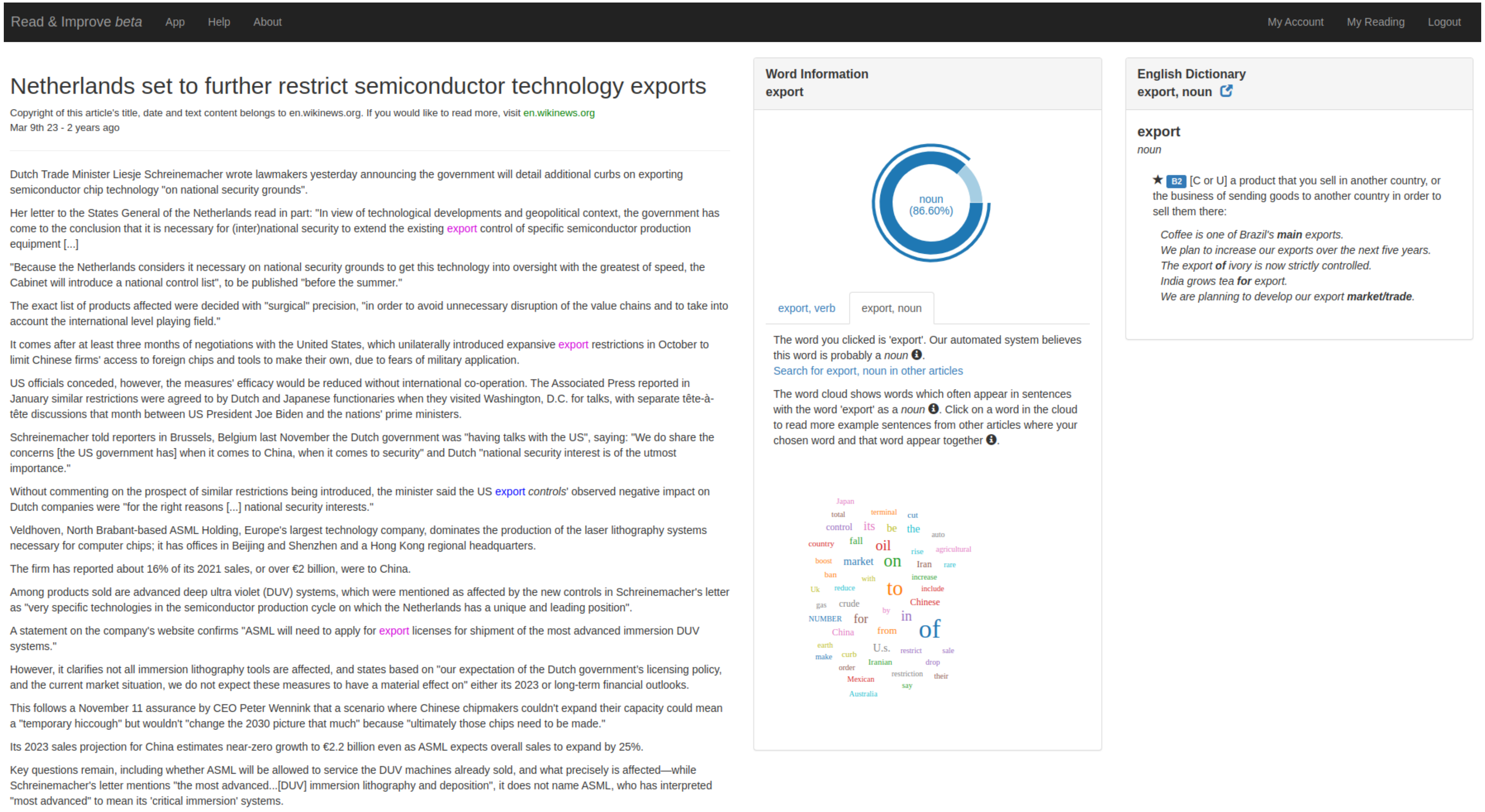
Text	Taco	Bell	restaurants	decided	Wednesday	to	remove	…
Labels	0	0	0	0	0	-100	1	…

A 0 label indicates no click, a 1 a click. -100 indicates non-content word POS. A text is a document, i.e. an entire WikiNews article.

E Ensemble Baseline

The classifiers used for the ensemble model are (using sklearn class names):

1. `RandomForestClassifier`
2. `GradientBoostingClassifier`
3. `HistGradientBoostingClassifier`
4. `MLPClassifier`
5. `LogisticRegression`
6. `BaggingClassifier`



Netherlands set to further restrict semiconductor technology exports

Copyright of this article's title, date and text content belongs to en.wikinews.org. If you would like to read more, visit en.wikinews.org
Mar 9th 23 - 2 years ago

Dutch Trade Minister Liesje Schreinemacher wrote lawmakers yesterday announcing the government will detail additional curbs on exporting semiconductor chip technology "on national security grounds".

Her letter to the States General of the Netherlands read in part: "In view of technological developments and geopolitical context, the government has come to the conclusion that it is necessary for (inter)national security to extend the existing export control of specific semiconductor production equipment [...]

"Because the Netherlands considers it necessary on national security grounds to get this technology into oversight with the greatest of speed, the Cabinet will introduce a national control list", to be published "before the summer."

The exact list of products affected were decided with "surgical" precision, "in order to avoid unnecessary disruption of the value chains and to take into account the international level playing field."

It comes after at least three months of negotiations with the United States, which unilaterally introduced expansive export restrictions in October to limit Chinese firms' access to foreign chips and tools to make their own, due to fears of military application.

US officials conceded, however, the measures' efficacy would be reduced without international co-operation. The Associated Press reported in January similar restrictions were agreed to by Dutch and Japanese functionaries when they visited Washington, D.C. for talks, with separate tête-à-tête discussions that month between US President Joe Biden and the nations' prime ministers.

Schreinemacher told reporters in Brussels, Belgium last November the Dutch government was "having talks with the US", saying: "We do share the concerns [the US government has] when it comes to China, when it comes to security" and Dutch "national security interest is of the utmost importance."

Without commenting on the prospect of similar restrictions being introduced, the minister said the US export *controls*' observed negative impact on Dutch companies were "for the right reasons [...] national security interests."

Veldhoven, North Brabant-based ASML Holding, Europe's largest technology company, dominates the production of the laser lithography systems necessary for computer chips; it has offices in Beijing and Shenzhen and a Hong Kong regional headquarters.

The firm has reported about 16% of its 2021 sales, or over €2 billion, were to China.

Among products sold are advanced deep ultra violet (DUV) systems, which were mentioned as affected by the new controls in Schreinemacher's letter as "very specific technologies in the semiconductor production cycle on which the Netherlands has a unique and leading position".

A statement on the company's website confirms "ASML will need to apply for export licenses for shipment of the most advanced immersion DUV systems."

However, it clarifies not all immersion lithography tools are affected, and states based on "our expectation of the Dutch government's licensing policy, and the current market situation, we do not expect these measures to have a material effect on" either its 2023 or long-term financial outlooks.

This follows a November 11 assurance by CEO Peter Wennink that a scenario where Chinese chipmakers couldn't expand their capacity could mean a "temporary hiccough" but wouldn't "change the 2030 picture that much" because "ultimately those chips need to be made."

Its 2023 sales projection for China estimates near-zero growth to €2.2 billion even as ASML expects overall sales to expand by 25%.

Key questions remain, including whether ASML will be allowed to service the DUV machines already sold, and what precisely is affected—while Schreinemacher's letter mentions "the most advanced...[DUV] immersion lithography and deposition", it does not name ASML, who has interpreted "most advanced" to mean its 'critical immersion' systems.

Word Information
export

The word you clicked is 'export'. Our automated system believes this word is probably a *noun* ❶.
Search for export, noun in other articles

The word cloud shows words which often appear in sentences with the word 'export' as a *noun* ❶. Click on a word in the cloud to read more example sentences from other articles where your chosen word and that word appear together ❶.

English Dictionary
export, noun

export
noun

★ B2 [C or U] a product that you sell in another country, or the business of sending goods to another country in order to sell them there:

*Coffee is one of Brazil's **main** exports.*
We plan to increase our exports over the next five years.
*The export **of** ivory is now strictly controlled.*
*India grows tea **for** export.*
*We are planning to develop our export **market/trade**.*

Figure 2: Screenshot of *Read&Improve* platform with information provided by lookup of the word “export”.

	A2	B1	B2	C1	C2	unk	sum
all	2102	2540	1638	522	6	2050	8858
train	1882	2295	1424	343	6	1872	7822
dev	143	139	71	155	0	122	630
test	77	106	143	24	0	56	406

Table 9: Look-up events across CEFR levels as estimated by essays from W&I.

	ar	bg	ca	cs	de	en	es	fa	fr	hi	hu	id	it	ja	jv	ka	ml	my	ne	pt	ro	ru	sr	ta	te	tr	ur	vi	zh	unk	sum
all	5	1	2	3	2	12	93	2	4	1	1	2	27	1	1	1	1	1	1	14	1	4	1	2	1	27	1	6	7	438	663
train	4	0	2	3	0	10	83	2	4	1	1	2	24	1	0	1	1	1	1	13	1	4	1	1	1	23	1	6	7	417	616
dev	1	0	0	0	1	1	15	0	1	0	0	0	6	1	1	0	0	1	0	2	0	0	0	0	0	4	0	1	3	52	90
test	2	1	0	0	1	3	16	1	0	0	1	0	2	0	0	0	0	1	0	4	0	0	0	1	0	3	0	0	0	32	68

Table 10: Users per L1. For experiments, less frequent languages are merged into the unknown category (unk).

They were combined using the sklearn `VotingClassifier` class, which was set to soft voting. No systematic hyperparameter tuning was required, instead we compared a variety of settings and combinations manually on the dev-set (to avoid over-fitting) and then applied the best to the test set.

The used features were:

- The frequency baseline score as described in Section 6.
- Relative position of the token in the text, defined as the proportion of seen tokens for the first 1000 tokens.
- Proportion of look-up events by user, calculated from the training split.
- Length of word in characters.
- CEFR-level as estimated by essays submitted by the user.
- Count of definitions for the word in the *Cambridge Advanced Learner's Dictionary*.
- Proportion of people who did not know the word type as retrieved from the ratings by Brysbaert et al. (2014).

For missing values, the average was used. To address label imbalance, we upsampled positive cases to achieve a proportion of 1-to-1. For the additionally added positively labelled data, we added small Gaussian noise to the frequency score, proportion of look-up event by user, the relative position.

	ar	en	es	it	pt	tr	vi	zh	unk	sum
all	12	19	169	70	29	48	10	15	955	1327
train	8	14	135	62	23	40	9	12	840	1143
dev	1	2	16	6	2	5	1	3	65	101
test	3	3	18	2	4	3	0	0	50	83

Table 11: L1s across documents seen by users (i.e. some users and articles appear multiple times in this table).

	ar	en	es	it	pt	tr	vi	zh	unk	sum
all	5	12	93	27	14	27	6	7	472	663
train	4	10	83	24	13	23	6	7	446	616
dev	1	1	15	6	2	4	1	3	57	90
test	2	3	16	2	4	3	0	0	38	68

Table 12: L1s across users – less frequent languages merged into unknown (unk). This merging process is used for our transformer models.

split	chats	clicks	con. tokens
D-chat	25	5	10027
D-read	26	67	33130

Table 13: Description of data and splits, including the number of content tokens for chatbot dialogues.

F Neural Models

The models used are described in Table 14. We used the LLaMA 3.1-8B, rather than a LLaMA 3.2 version, because it was closer to the size of the Gemma model.

model	hf-name	approach
Longformer	allenai/longformer-base-4096	finetuning
LLaMA 3.2	meta-llama/Llama-3.2-1B	finetuning
LLaMA Instruct	unsloth/Meta-Llama-3.1-8B-Instruct	prompting
Gemma	unsloth/gemma-2-9b-it	prompting

Table 14: Details of models used, including name on huggingface hub and experimental approach.

F.1 Hyperparameters

The datasets for the different tasks strongly differ in input length. Both the SEP and DLU dataset operate on data longer than sentences, but while DLU consists of WikiNews texts, the SEP consists of student essays split into chunks of one or more paragraphs. The 2018 CWI dataset (Yimam et al., 2017) is on the sentence level, i.e. the inputs are much shorter than for the other datasets. To work with these different datasets, we found it necessary to change the hyperparameter space, in particular the space for the training batch size.

The hyperparameter spaces as well as the selected hyperparameters are described in tables 15 to 17. For each combination of model and loss function, we run 20 trials without pruning, where the searches were performed with Optuna. Additional settings for Optuna, such as using the log space, are noted in the table. The target metric for maximization was the AUC.

G Prompting

We use two prompt templates, one for zero-shot and one for few-shot inference. Both prompts instruct the LLM to consider a paragraph of text and the learner's English CEFR level. The models are asked to predict which words the learner is likely unfamiliar with, and return these words in a JSON format. The zero-shot prompt directly provides the task instructions and desired output format, while the few-shot prompt includes three illustrative examples of different learners' word choices in different paragraphs of text.

G.1 Prompts

```
CLICK_DATA_APPROXIMATION_PROMPT = {'system': """ # Task Introduction You are an AI assistant now doing a language test. You will receive a paragraph of text. you will need to predict based on your user's English level what words the user might click on(The user will click on the words he or she is not familiar with.
  –
  # About the user's english level A1: Can write personal information (e.g. likes and dislikes, family, pets) using simple words, phrases and sentences.
  A2: Can write a series of simple phrases and sentences, linked with words like 'and', 'but' and 'because'.
  B1: Can write straightforward texts about familiar topics or simple information and ideas. Can link sentences into a connected text.
  B2: Can write clear, detailed texts on different subjects. Can use information and arguments from other sources in their writing.
  C1: Can write clear, well-structured, detailed texts on complex subjects, showing the important issues, giving examples and writing a conclusion if appropriate. Can use the correct style of writing relevant to the target reader.
  C2: Can write clear, smoothly flowing, complex texts in an appropriate and effective style. Can use a logical structure which helps the reader find the main points.
  –
  # Expected Output Your answers should be formatted in JSON format with following keys and values: 1. output_tokens: a list of tokens that you predict the user will click on, each token should appear only once 2. reason: a short string explaining your prediction of the tokens
  NOTE: please make sure the output tokens are unique. each token in the list should appear only once """, 'user': """
  # task detail
  The user's english level is:
  {cefr_level}
  The paragraph you need to predict on:
  {paragraph_text}
  The tokens in the paragraph:
  {tokens}
  Respond only with valid JSON.
  –
  """ }
  CLICK_DATA_APPROXIMATION_FEWSHOT_PROMPT = {'system': """ # Task Introduction You are an AI assistant now doing a language test. You will receive a paragraph of text. you will need to predict based on your user's English level what words the user might click on(The user will click on the words he or she is not familiar with.
  –
  # About the user's english level
  A1: Can write personal information (e.g. likes and dislikes, family, pets) using simple words, phrases and sentences.
  A2: Can write a series of simple phrases and sentences, linked with words like 'and', 'but' and 'because'.
  B1: Can write straightforward texts about familiar topics or simple information and ideas. Can link sentences into a connected text.
  B2: Can write clear, detailed texts on different subjects. Can use information and arguments from other sources in their writing.
  C1: Can write clear, well-structured, detailed texts on complex subjects, showing the important issues, giving examples and writing a conclusion if appropriate. Can use the correct style of writing relevant to the target reader.
  C2: Can write clear, smoothly flowing, complex texts in an appropriate and effective style. Can use a logical structure which helps the reader find the main points.
  –
  # Expected Output Your answers should be formatted in JSON format with following keys and values: 1. output_tokens: a list of tokens that you predict the user will click on, each token should appear only once
  2. reason: a short string explaining your prediction of the tokens
  NOTE: please make sure the output tokens are unique. each token in the list should appear only once
  –
  # Examples Here are some examples from user of the same english level as the one you are goingto mimic.
  ## Example1:
  {example1}
  ## Example2:
  {example2}
  ## Example3:
  {example3}
  """, 'user': """
  # task detail
  The user's english level is:
  {cefr_level}
  The paragraph you need to predict on:
  {paragraph_text}
  The tokens in the paragraph:
  {tokens}
  Respond only with valid JSON.
  –
  """ }
```

	Space	Info	Longformer (ROC*)	Longformer (BCE)	LLaMA (ROC*)	LLaMA (BCE)
Epochs	[1, 30]		25	14	30	14
Learning Rate	$[10^{-9}, 10^{-2}]$	log space	3.6×10^{-6}	6.7×10^{-5}	3.7×10^{-5}	2.4×10^{-4}
Pos. Weight	[0.8, 30]	BCE only	-	0.81	-	29
γ	[0.05, 0.75]	ROC* only	0.59	-	0.05	-
Sample Size	[300, 10000]	ROC*, step size=100	6600	-	300	-
Batch Size (p.D.)	[4, 14]	step size = 2	12	8	4	12

Table 15: Hyperparameter space and selected hyperparameters for DLU prediction models. We report the per device batch size. The number of devices was always set to 4.

	Space	Info	Longformer (ROC*)	Longformer (BCE)	LLaMA (ROC*)	LLaMA (BCE)
			Models finetuned only on CWI			
Epochs	[1, 30]		8	11	22	11
Learning Rate	$[10^{-9}, 10^{-2}]$	log space	7.0×10^{-5}	4.6×10^{-5}	1.1×10^{-4}	2.3×10^{-5}
Pos. Weight	[0.8, 30]	BCE only	-	29.9	-	26.5
γ	[0.05, 0.75]	ROC* only	0.69	-	0.45	-
Sample size	[300, 10000]	ROC*, step size=100	3400	-	4200	-
Batch size (p.D.)	[8, 80]	step size = 2	48	10	50	72
			Models finetuned on DLU and then on CWI			
Epochs	[1, 30]		27	10	26	25
Learning Rate	$[10^{-9}, 10^{-2}]$	log space	3.6×10^{-6}	5.4×10^{-5}	7.2×10^{-5}	5.2×10^{-5}
Pos. Weight	[0.8, 30]	BCE only	-	23.78	-	15.46
γ	[0.05, 0.75]	ROC* only	0.66	-	0.23	-
Sample size	[300, 10000]	ROC*, step size=100	3300	-	3800	-
Batch size (p.D)	[8, 80]	step size = 2	8	42	16	30

Table 16: Hyperparameter space and selected hyperparameters for CWI prediction models. We report the per device batch size. The number of devices was always set to 4.

	Space	Info	Longformer (ROC*)	Longformer (BCE)	LLaMA (ROC*)	LLaMA (BCE)
			Models finetuned only on SEP task			
Epochs	[1, 30]		24	10	10	6
Learning Rate	$[10^{-9}, 10^{-2}]$	log space	3.1×10^{-5}	1.0×10^{-5}	8.6×10^{-6}	2.3×10^{-5}
Pos. Weight	[0.8, 30]	BCE only	-	15.08	-	16.90
γ	[0.05, 0.75]	ROC* only	0.34	-	0.65	-
Sample size	[300, 10000]	ROC*, step size=100	2600	-	9100	-
Batch size (p.D.)	[4, 44]	step size = 2	36	34	38	18
			Models finetuned on DLU and then on SEP task			
Epochs	[1, 30]		17	2	8	3
Learning Rate	$[10^{-9}, 10^{-2}]$	log space	9.0×10^{-5}	1.8×10^{-4}	3.1×10^{-6}	2.9×10^{-4}
Pos. Weight	[0.8, 30]	BCE only	-	17.99	-	12.33
γ	[0.05, 0.75]	ROC* only	0.05	-	0.55	-
Sample size	[300, 10000]	ROC*, step size=100	300	-	4200	-
Batch size (p.D)	[4, 44]	step size = 2	16	30	40	20

Table 17: Hyperparameter space and selected hyperparameters for SEP prediction models. We report the per device batch size. The number of devices was always set to 4.

H Significance Tests

We perform a two-sided permutation test using SciPy (Virtanen et al., 2020). We set `permutation_type='samples'` and `random_state='1848'`. The number of permutations is left at the default 9999. The test statistics and associated p-values can be found in tables tables 18 to 20.

The Bonferroni-correct p-value is 0.0027. We rounded the digits of the threshold using the floor, as this makes the significance test more restrictive.

I Processing of CWI

The CWI dataset we used (Yimam et al., 2017, 2018) provides one data row for each labelled word, even if these words occur in the same sentences. To reduce training time and make the processing more similar to DLU, we treated these words as occuring together during training. For evaluation, we again made one prediction per input, as in the original CWI dataset for comparability. This might have affected our performance negatively, explaining some of the difference to the results reported by

	Metric	Statistic	p-Value	
Longformer	AUC	compare	2.8×10^{-2}	2.6×10^{-1}
LLaMA	AUC	compare	1.5×10^{-2}	5.2×10^{-1}

Table 18: Significance tests for DLU. The tests concern whether using the ROC* vs. the BEC loss changes the AUC.

	Metric	Loss	Statistic	p-Value
Longformer	AUC	roc	7.8×10^{-5}	9.6×10^{-1}
Longformer	F1	roc	7.5×10^{-3}	6.0×10^{-2}
Longformer	AUC	bce	4.9×10^{-3}	1.8×10^{-1}
Longformer	F1	bce	2.8×10^{-2}	2.0×10^{-4}
LLaMA	AUC	roc	7.2×10^{-3}	1.5×10^{-1}
LLaMA	F1	roc	3.2×10^{-2}	2.0×10^{-4}
LLaMA	AUC	bce	1.6×10^{-2}	8.0×10^{-3}
LLaMA	F1	bce	6.4×10^{-3}	2.1×10^{-1}

Table 19: Significance tests for CWI task, testing whether models finetuned on DLU first perform differently on F_1 or AUC.

	Metric	Loss	Statistic	p-Value
Longformer	AUC	roc	9.2×10^{-3}	2.7×10^{-1}
Longformer	F1	roc	1.1×10^{-1}	2.0×10^{-4}
Longformer	AUC	bce	1.8×10^{-1}	2.0×10^{-4}
Longformer	F1	bce	1.1×10^{-1}	2.0×10^{-4}
LLaMA	AUC	roc	3.9×10^{-2}	4.0×10^{-4}
LLaMA	F1	roc	6.8×10^{-2}	2.0×10^{-4}
LLaMA	AUC	bce	8.1×10^{-3}	6.3×10^{-1}
LLaMA	F1	bce	3.1×10^{-2}	2.2×10^{-3}

Table 20: Significance tests for SEP task, testing whether models finetuned on DLU first perform differently on F_1 or AUC.

Smădu et al. (2024).

J Further Discussion of Results

Using an adaptive threshold for the F_2 (aF_2) consistently improves the performance of the baseline further, which is not always the case for the transformer models. This suggests that the decision threshold for transformer models is context dependent and cannot be transferred between splits. Furthermore, it shows that the simple frequency baseline can be further improved with simple.

As a result of the different effect of the adaptive threshold, the highest F_2 value (23.4%) by a transformer model (Longformer ROC*) is higher than the aF_2 (21%) of the frequency baseline, even though the baseline achieves the highest aF_2.

K Additional Results

In Section 7 we report results on the DLU train split, but as we release only the dev split with this paper, we report the results on this split in Table 22. The training method was the same as for the results on the test split.

The results might be affected by the same documents being repeated in the evaluation split (dev or test) because more than one user interacted with it. To investigate this effect, we also evaluated on these splits after removing all but one randomly selected version of each document, i.e. the look-up data for one random user per document. The results are shown in tables 24 and 25. The adaptive threshold for the aF_2 is the same as for the original evaluation.

		A				B				C				unk				All				D-read			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	10.3	15.1	-	56.5	14.1	18.0	-	57.6	**13.4**	20.9	-	64.1	9.1	14.3	-	57.3	12.2	17.0	-	57.7	2.2	4.6	-	57.4
	fewshot	10.2	16.1	-	57.4	12.5	17.8	-	57.5	12.8	**21.9**	-	67.2	10.4	16.4	-	59.1	11.7	17.6	-	58.4	-	-	-	-
LLaMA-Inst.	zeroshot	8.7	16.4	-	58.6	7.8	12.8	-	53.0	5.3	10.0	-	56.6	6.9	13.0	-	57.2	7.6	13.5	-	55.0	1.0	2.4	-	56.1
	fewshot	8.3	15.1	-	56.7	7.6	12.4	-	52.7	4.5	8.9	-	55.5	3.8	7.1	-	49.7	6.7	11.7	-	53.2	-	-	-	-
LLaMA	ROC*	0.0	0.0	7.2	75.7	0.0	0.0	8.2	65.9	0.0	0.0	0.0	62.1	0.0	0.0	11.2	68.1	0.0	0.0	7.9	67.8	0.0	0.0	3.0	76.7
	BCE	11.0	22.2	21.9	72.7	12.4	21.8	19.4	64.9	5.6	11.8	5.8	63.0	8.0	16.7	19.7	69.4	10.4	20.0	18.9	66.2	2.7	5.6	**5.2**	77.1
Longformer	ROC*	14.4	25.6	18.6	78.5	16.0	25.5	21.7	72.4	5.7	11.0	13.4	64.9	12.1	20.7	18.7	77.6	14.0	23.6	20.0	73.9	**2.7**	**5.8**	2.8	83.4
	BCE	0.0	0.0	12.3	70.9	0.0	0.0	13.2	70.7	0.0	0.0	3.4	60.5	0.0	0.0	8.4	76.6	0.0	0.0	11.6	71.1	0.0	0.0	3.9	74.6
Baseline	freq.	8.7	18.9	24.7	75.8	9.6	20.6	23.1	71.4	4.2	9.9	10.8	72.3	5.7	12.9	16.6	72.2	8.1	17.7	21.0	72.5	0.9	2.2	3.3	**84.9**
	ens.	**22.2**	**32.8**	**31.6**	**85.9**	**17.3**	**26.1**	**28.0**	**76.2**	12.3	18.8	**19.6**	**81.9**	**14.2**	**24.3**	**23.6**	**80.4**	**17.3**	**26.5**	**27.4**	**79.2**	-	-	-	-

Table 21: Prediction results on the DLU test split, but for the prompting model, we take all occurrences of a word listed by the prompted model to be looked-up. (Results on non-prompting models are unchanged.)

		A				B				C				unk				All			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	11.7	14.0	-	54.7	9.9	13.3	-	55.7	12.0	9.5	-	52.2	13.0	15.2	-	55.2	11.2	13.1	-	54.4
	fewshot	10.8	12.5	-	53.9	9.4	12.4	-	55.1	9.7	7.5	-	51.5	12.9	18.0	-	56.3	10.6	12.8	-	54.1
LLaMA-Inst.	zeroshot	8.9	9.4	-	52.6	9.0	14.3	-	56.6	15.1	16.1	-	51.7	6.1	8.5	-	49.7	9.4	12.6	-	53.4
	fewshot	11.2	15.6	-	55.1	6.1	10.4	-	53.1	12.9	12.9	-	51.0	9.1	13.2	-	52.6	8.4	12.4	-	52.7
LLaMA	ROC*	0.0	0.0	13.2	71.6	0.0	0.0	7.1	64.8	0.0	0.0	1.6	51.2	0.0	0.0	9.9	68.5	0.0	0.0	7.9	63.3
	BCE	15.4	**25.4**	20.9	69.0	7.8	13.4	10.1	58.2	15.3	14.6	10.9	62.2	13.8	24.2	21.7	67.4	11.8	18.9	15.5	62.1
Longformer	ROC*	**17.0**	25.4	18.0	71.8	10.2	19.1	16.6	69.5	15.0	17.9	10.0	51.5	15.3	23.8	19.9	71.7	12.8	21.0	16.2	65.6
	BCE	0.0	0.0	22.0	**73.3**	0.0	0.0	16.1	**71.1**	0.0	0.0	9.3	56.8	0.0	0.0	17.8	72.9	0.0	0.0	16.3	68.3
Baseline	freq.	9.8	20.6	22.4	63.2	6.5	14.6	17.0	68.3	**22.9**	**39.7**	**37.7**	62.1	11.4	23.8	27.2	69.8	9.7	20.6	22.7	65.7
	ens.	14.6	23.3	**24.0**	69.0	**11.3**	**20.1**	**19.3**	69.3	22.2	23.9	31.0	**64.9**	**21.6**	**32.9**	**31.3**	**76.7**	**15.0**	**23.8**	**24.2**	**69.0**

Table 22: Prediction results on the DLU dev split. "aF2" stands for F2 with a adaptive threshold, as discussed in Section 5.

		A				B				C				unk				All			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	11.9	14.7	-	55.0	8.8	12.1	-	54.9	15.0	12.3	-	53.3	13.5	17.3	-	56.1	11.3	13.8	-	54.6
	fewshot	11.2	14.8	-	54.8	8.5	11.9	-	54.7	12.7	10.2	-	52.4	12.4	19.8	-	57.3	10.7	14.3	-	54.6
LLaMA-Inst.	zeroshot	10.1	12.4	-	53.6	6.9	12.5	-	55.1	21.4	26.8	-	55.4	6.9	11.1	-	49.5	9.4	14.8	-	54.1
	fewshot	10.3	16.0	-	54.9	4.9	9.3	-	51.1	20.1	23.8	-	54.6	8.7	14.5	-	52.0	8.3	13.8	-	52.7
LLaMA	ROC*	0.0	0.0	13.2	71.6	0.0	0.0	7.1	64.8	0.0	0.0	1.6	51.2	0.0	0.0	9.9	68.5	0.0	0.0	7.9	63.3
	BCE	15.4	**25.4**	20.9	69.0	7.8	13.4	10.1	58.2	15.3	14.6	10.9	62.2	13.8	24.2	21.7	67.4	11.8	18.9	15.5	62.1
Longformer	ROC*	**17.0**	25.4	18.0	71.8	10.2	19.1	16.6	69.5	15.0	17.9	10.0	51.5	15.3	23.8	19.9	71.7	12.8	21.0	16.2	65.6
	BCE	0.0	0.0	22.0	**73.3**	0.0	0.0	16.1	**71.1**	0.0	0.0	9.3	56.8	0.0	0.0	17.8	72.9	0.0	0.0	16.3	68.3
Baseline	freq.	9.8	20.6	22.4	63.2	6.5	14.6	17.0	68.3	**22.9**	**39.7**	**37.7**	62.1	11.4	23.8	27.2	69.8	9.7	20.6	22.7	65.7
	ens.	14.6	23.3	**24.0**	69.0	**11.3**	**20.1**	**19.3**	69.3	22.2	23.9	31.0	**64.9**	**21.6**	**32.9**	**31.3**	**76.7**	**15.0**	**23.8**	**24.2**	**69.0**

Table 23: Prediction results on the DLU dev split, but for the prompting model, we take all occurrences of a word listed by the prompted model to be looked-up. (Results on non-prompting models are unchanged.)

		A				B				C				unk				All			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	9.6	11.6	-	52.7	15.8	20.3	-	59.2	0.0	0.0	-	47.2	1.7	2.9	-	48.0	9.3	13.0	-	55.1
	fewshot	16.1	19.8	-	57.7	14.5	20.6	-	59.7	0.0	0.0	-	46.8	5.3	9.0	-	53.8	11.4	16.6	-	57.7
LLaMA-Inst.	zeroshot	11.0	16.9	-	54.7	10.5	18.5	-	58.8	3.0	6.4	-	56.5	4.4	8.4	-	53.1	8.6	15.2	-	57.1
	fewshot	9.7	15.7	-	53.1	8.5	12.9	-	53.9	3.3	6.8	-	57.0	0.0	0.0	-	42.5	6.1	10.2	-	52.3
LLaMA	ROC*	4.9	3.8	3.8	72.2	21.7	17.9	17.9	70.2	0.0	0.0	0.0	**76.2**	**13.3**	15.2	15.2	58.9	14.9	13.1	13.1	69.5
	BCE	17.1	24.2	24.2	69.5	17.6	24.2	24.2	67.0	0.0	0.0	0.0	58.8	11.9	**22.0**	**22.0**	**74.2**	14.4	22.1	22.1	68.4
Longformer	ROC*	**22.5**	**26.5**	**26.5**	**78.8**	**22.0**	**29.2**	**29.2**	**75.4**	**5.3**	**10.0**	**10.0**	62.8	7.2	10.9	10.9	73.2	**17.3**	**23.7**	**23.7**	**75.5**
	BCE	16.1	16.4	16.4	73.0	12.6	12.5	12.5	72.2	0.0	0.0	0.0	72.7	7.0	9.0	9.0	73.6	11.1	12.2	12.2	72.7

Table 24: Prediction results on test split when for each document only one user was randomly selected.

		A				B				C				unk				All			
		F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC	F_1	F_2	aF_2	AUC
Gemma-Inst.	zeroshot	7.5	11.2	-	53.7	4.1	5.7	-	50.8	**26.2**	21.7	-	58.5	0.0	0.0	-	46.6	8.8	11.2	-	53.7
	fewshot	6.7	10.4	-	52.9	7.5	10.9	-	54.5	19.2	17.9	-	55.9	0.0	0.0	-	47.7	9.2	12.3	-	54.3
LLaMA-Inst.	zeroshot	8.3	14.3	-	56.2	6.3	12.1	-	56.0	14.8	19.4	-	53.2	2.6	6.1	-	59.0	6.9	12.8	-	53.3
	fewshot	8.3	15.8	-	58.2	3.7	6.8	-	49.6	22.7	**35.3**	-	**63.1**	0.0	0.0	-	41.3	9.3	16.8	-	57.9
LLaMA	ROC*	5.7	5.6	5.6	64.7	6.1	5.6	5.6	67.6	0.0	0.0	0.0	50.6	**16.7**	**20.8**	**20.8**	76.2	5.0	4.3	4.3	61.0
	BCE	4.5	7.0	7.0	61.8	4.4	6.0	6.0	53.8	7.5	5.7	5.7	54.2	8.0	16.1	16.1	**83.7**	5.5	7.1	7.1	52.1
Longformer	ROC*	**17.4**	**20.0**	**20.0**	71.3	13.3	20.3	20.3	68.5	13.7	15.6	**15.6**	54.5	10.8	20.4	20.4	70.6	13.7	**18.7**	**18.7**	64.9
	BCE	13.0	15.0	15.0	**71.7**	**18.0**	**22.2**	**22.2**	**69.6**	12.8	12.4	12.4	55.0	15.4	20.0	20.0	72.3	**15.3**	17.2	17.2	**66.6**

Table 25: Prediction results on dev split when for each document only one user was randomly selected.

IPA CHILDES & G2P+: Feature-Rich Resources for Cross-Lingual Phonology and Phonemic Language Modeling

Zébulon Goriely 🍊 **Paula Buttery** 🍊🍋
🍊 Department of Computer Science & Technology, University of Cambridge, U.K.
🍋 ALTA Institute, University of Cambridge, U.K.
🍊 `firstname.secondname@cl.cam.ac.uk`

Abstract

In this paper, we introduce two resources: (i) `G2P+`, a tool for converting orthographic datasets to a consistent phonemic representation; and (ii) IPA CHILDES, a phonemic dataset of child-directed and child-produced speech across 31 languages. Prior tools for grapheme-to-phoneme conversion result in phonemic vocabularies that are inconsistent with established phonemic inventories, an issue which `G2P+` addresses by leveraging the inventories in the Phoible database (Moran and McCloy, 2019). Using this tool, we augment CHILDES (MacWhinney and Snow, 1985) with phonemic transcriptions to produce IPA CHILDES. This new resource fills several gaps in existing phonemic datasets, which often lack multilingual coverage, spontaneous speech, and a focus on child-directed language. We demonstrate the utility of this dataset for phonological research by training phoneme language models on 11 languages and probing them for distinctive features, finding that the distributional properties of phonemes are sufficient to learn major class and place features cross-lingually.

phonemetransformers/ipa-childes
(CC BY 4.0)

codebyzeb/g2p-plus (MIT)

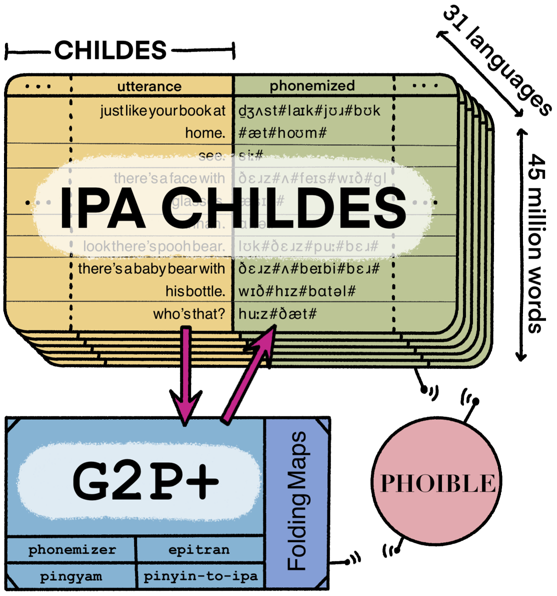


Figure 1: An overview of IPA CHILDES and `G2P+`, which are introduced in this paper.

1 Introduction

Phonological research can be enriched by large-scale data-oriented studies that investigate phoneme function across the globe's languages. However, while written text is plentiful and easily accessible across hundreds of languages, phonemic data is much more limited in availability. Phonemic datasets can be created by employing expert phoneticians to carefully transcribe speech, but this is a time-consuming process and completely infeasible for creating large datasets. Instead, the typical approach is to use grapheme-to-phoneme (G2P) conversion tools, which use statistical rules and pronunciation dictionaries to convert orthographic text to a phonemic representation. Open-source G2P tools have been used to create large and multilingual phonemic datasets with domains ranging from telephone conversations to legal proceedings. However, the fact that these tools are open-sourced and use a variety of statistical approaches and transcription schemes means that phonemic corpora vary considerably according to their phonemic vocabularies and level of phonetic detail, making it difficult to compare findings and incorporate other linguistic resources into analysis.

There is also a lack of phonemic data for certain domains, preventing phonological research in these areas. In particular, we note that it is difficult to find phonemic data for child-centered speech[1] and,

---

[1] Child-centered speech is speech occurring within a child's environment and includes child-directed and child-produced utterances.

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 502–521
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

in general, spontaneous speech across several languages. The *de-facto* repository for child-centered data is the Child Language Data Exchange System (CHILDES), which currently contains over 1.4TB of transcript data in over 40 languages (MacWhinney and Snow, 1985; MacWhinney, 2019). The impact of CHILDES across clinical and linguistic research has been profound (Ratner, 2024) but the largely orthographic nature of the data has prevented phonological experimentation.[2]

We thus identify two major challenges impeding phonological research. First, the lack of consistent G2P conversion, which we address by developing `G2P+`, a tool for converting orthographic text to a phonemic representation. `G2P+` leverages existing G2P tools for conversion but carefully maps the output to established phonemic inventories in Phoible, a database of cross-linguistic phonological inventory data. Using Phoible inventories not only ensures consistency for each language regardless of the G2P backend used, but the database also contains phonological feature information, supporting fine-grained phonological analysis. Second, we address the lack of a multilingual phonemic dataset of child-centered speech by using `G2P+` to convert the majority of the CHILDES database to phonemes. The resulting dataset, IPA CHILDES, contains phonemic transcriptions of 31 languages in CHILDES, totaling 45 million words. We illustrate these resources in fig. 1.

We exemplify how to use these resources by training cross-lingual phoneme language models. Phoneme LMs have a wide variety of applications in NLP, including lyric generation (Ding et al., 2024), text-to-speech (Li et al., 2023), and low-resource language modeling (Leong and Whitenack, 2022). Developmentally plausible training corpora also provide a means of studying emergent phonology, but past work has been limited by the availability of training and evaluation resources in languages besides English. Here, after establishing the scaling conditions of phoneme LMs, we train monolingual models on the 11 largest languages in IPA CHILDES. Using the fact that `G2P+` maintains a correspondence with Phoible during conversion, we use linear probes to predict an input phoneme's phonological features from its contextual embedding. We evaluate this approach against the phoneme's feature description in Phoible and find that the probes consistently correctly predict the 'syllabic' and 'consonantal' features, indicating the broad separation of vowels and consonants across languages and demonstrating the utility of phoneme LMs for studying emergent phonology.

These experiments demonstrate the utility of our tools for phonological analysis. We release `G2P+`, IPA CHILDES, and all trained models to support future work.

2 Related Work

2.1 Phonemic Datasets

Phonemic data is required to investigate a range of linguistic phenomena. Recently, researchers have used data-driven approaches to study morphological theories of acquisition (Kirov and Cotterell, 2018), explore the role of distributional information in phonology (Mayer, 2020), calculate cross-language phonological distance (Eden, 2018) and simulate early lexicon learning (Goriely et al., 2023). Despite the benefits of phonemic data, few such datasets exist.

Written text and audio datasets are far more plentiful than phonemic datasets. Written text, being widely distributed and easy to collect through practices such as web-scraping (Bansal et al., 2022), has steered years of NLP research, ranging from the parsers trained on the Penn Treebank (Taylor et al., 2003) to the large language models trained on billion-word datasets like the Pile (Gao et al., 2020). Despite the availability of written text, it is often inappropriate for speech technology and phonological research. Instead, since tape recorders became widely available, researchers have created datasets of human speech. These now include **elicited** speech corpora such as TIMIT, (Garofolo et al., 1993), FLEURS (Conneau et al., 2023), the MSWC (Mazumder et al., 2021), GlobalPhone (Schultz, 2002) and CommonVoice (Ardila et al., 2020); **audio book** corpora such as LibriSpeech (Panayotov et al., 2015), MLS (Pratap et al., 2020) and the CMU Wilderness Corpus (Black, 2019); and **naturalistic** speech corpora such as Switchboard (Godfrey et al., 1992), the Fisher corpus (Cieri et al., 2004), the British National Corpus (Consortium, 2007), the Buckeye corpus (Pitt et al., 2007), Babel (Harper, 2011) and VoxLingua107 (Valk and Alumäe, 2021). Of these datasets, only TIMIT, MLS and Switchboard include phonemic annotations, limiting their use in phonological analysis.

---

[2] CHILDES does contain phonetic transcriptions for some languages as part of the PhonBank project, but only for a select few corpora and only for child-produced utterances, impeding the phonological analysis of child-*directed* speech.

Later work augmented these datasets with phonemic transcriptions. These include Audio BNC derived from the British National Corpus (Coleman et al., 2011), LibriLight derived from LibriSpeech (Kahn et al., 2020), VoxClamantis derived from the CMU Wilderness Corpus (Salesky et al., 2020), VoxCommunis derived from CommonVoice (Ahn and Chodroff, 2022) and IPAPACK derived from FLEURS and MSWC (Zhu et al., 2024).

These datasets and their phonemically-annotated successors all vary considerably according to the language coverage, number of words, domain and the presence of text-based transcriptions. We provide a summary of these properties in appendix C. Our dataset, IPA CHILDES, is the first phonemic dataset for *child-centered* speech and the first *multilingual* phonemic dataset for spontaneous speech.

2.2 Grapheme to Phoneme Conversion

Ideally, phonemic transcriptions of speech would originate from expert human annotators, but such annotation is incredibly slow. For instance, it was estimated that it would take 120 person-years to transcribe and align the 1200 hours of speech in the Audio BNC corpus (Coleman et al., 2011). Of the phonemic datasets described above, only the smallest, TIMIT, was fully transcribed by human experts, at a rate of only 100 sentences per week (Zue and Seneff, 1996; Lamel et al., 1989). Switchboard also provides human-annotated phonemic transcriptions but only for 5,000 utterances (Greenberg et al., 1996).

In practice, phonemic transcriptions are produced using G2P. In the simplest case, this involves the use of pronunciation dictionaries such as the Carnegie Mellon University (CMU) Pronouncing Dictionary[3] or the English Pronouncing Dictionary (Jones, 2011). These were used to create the phonemic transcriptions for the Buckeye Corpus, Audio BNC and Babel, but pronunciation dictionaries are limited by the items included in the dictionary and so may fail to convert part-words, interruptions or rare proper nouns, which frequently occur in spontaneous speech. More sophisticated G2P methods combine pronunciation dictionaries with statistical models. These systems have been developed for many languages using rules or finite-state transducers to generalize to unseen words (Mortensen et al., 2018; Hasegawa-Johnson et al., 2020; Bernard and Titeux, 2021). Other G2P systems have applied neural networks to automatically learn these rules and generalize to new languages (Novak et al., 2016; Zhu et al., 2022).

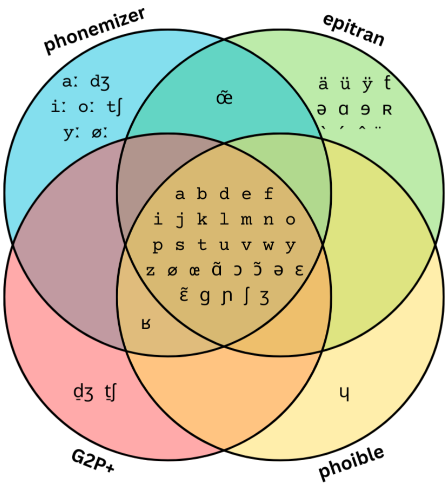


Figure 2: Venn diagram of the inventories produced by `phonemizer`, `epitran` and `G2P+` compared to Phoible inventory 2269 for French.

As G2P systems operate only from text, they may fail to capture accents and the variation found in natural speech (see appendix A for a discussion). Nevertheless, G2P systems provide a useful method for producing phonemic transcriptions at scale, and were used to produce the transcriptions for LibriSpeech, VoxClamantis and IPAPACK. The fact that transcription errors may occur is often acknowledged as a limitation, but rarely are the outputs of different G2P systems compared to each other or to established inventories. For instance, `epitran` and `phonemizer`, two popular tools described in section 3.1, produce very different inventories for French, as demonstrated in fig. 2.

In this work, we leverage existing statistical G2P tools, validate their outputs using maps to Phoible inventories, and use our resulting tool to produce phonemic transcriptions for the utterances in the CHILDES database.

2.3 Phoneme LMs and Child-Centered Data

In this work, we illustrate one use of our dataset by training small monolingual LMs on 11 languages and examining the representations they learn for individual phonemes.

Training models on such little data (here, only 500 thousand words) may be considered atypical in the modern NLP landscape, but questions of developmental plausibility have led to an increased

---

[3] `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`

interest in pretraining with limited data. For instance, the BabyLM workshop series challenges participants to train smaller models on data that is limited by both scale, 10–100 million words, and by domain, with the pre-training corpus including data from CHILDES, among other child-centered corpora (Warstadt et al., 2023; Hu et al., 2024b). Such limitations have led to the development of new architectures (Georges Gabriel Charpentier and Samuel, 2023; Charpentier and Samuel, 2024), motivated cognitively-inspired pre-training strategies (Huebner et al., 2021; Diehl Martinez et al., 2023) and allowed for gaining insights into human learning (Yedetore et al., 2023). The majority of this work has centered on English. Exceptions include Capone et al. (2024); Shen et al. (2024), who train Italian monolingual and bilingual models, respectively, Yadavalli et al. (2023) who use data from five language in CHILDES to explore second language acquisition theories (but only train an English LM) and Salhan et al. (2024), who use age-ordered data from four languages in CHILDES to explore fine-grained curricula inspired by language acquisition.

However, these BabyLMs are typically trained on orthographic text, limiting their ability to be studied at the phonological level, and generally use subword tokens, which do not generally correspond to cognitively plausible units (Beinborn and Pinter, 2023) limiting their value for psycholinguistic research (Giulianelli et al., 2024). Bunzeck et al. (2024) and Goriely et al. (2024) both establish phoneme-based training of BabyLMs (where tokens consist of individual phonemes, with word boundaries removed) but only train on English text. Here, we use IPA CHILDES to demonstrate phoneme-based training for 11 languages and leverage the fact that `G2P+` maintains a correspondence to Phoible in order to probe our BabyLMs for knowledge of distinctive features.

3 G2P+

We introduce `G2P+` as a tool for converting datasets from an orthographic representation to a phonemic representation. It operates either as a python library or as a command-line program; the user selects one of four backends and the language to use for conversion. Each backend supports a different set of languages as described in section 3.1. The recommended backends for each of the languages in IPA CHILDES are given in appendix B and example usage of the tool is given in appendix D.

Each line of orthographic text is converted to phonemes, represented using the International Phonetic Alphabet (IPA). Regardless of the backend selected, the representation is consistent, with phonemes separated by whitespace (for convenient tokenization) and unique delimiters used to separate words and utterances (see appendix E for details).

The output representation is also consistent in terms of the set of phonemes types produced, using *folding*, as described in section 3.2. Without folding, each backend produces a different set of phonemes (as demonstrated in fig. 2) which may not align with established phoneme inventories. Our folding maps not only ensure the output is consistent regardless of the backend chosen, but also makes it easy to leverage information in Phoible in analysis, as demonstrated in section 5.2.

3.1 G2P Backends

In order to support a wide variety of languages, we implement wrappers around four backend G2P tools:

`phonemizer:` Phonemizer (Bernard and Titeux, 2021) is a python library for G2P in various languages based on `eSpeak`[4], an open-source speech synthesizer which supports over one hundred languages and accents (Dunn and Vitolins, 2022).

`epitran:` Epitran (Mortensen et al., 2018) supports the automatic grapheme-to-phoneme conversion of text across many languages, accents and scripts, with a particular focus on low-resource languages. For the majority of the 92 languages supported,[5] it uses greedily-interpreted grapheme-to-phoneme maps augmented with context-sensitive pre-processor and post-processor rewrite rules.

`pinyin-to-ipa:` Pinyin-to-ipa (Taubert, 2024) is a python library for converting Mandarin written in pinyin to IPA using a few contextual grapheme-to-phoneme maps. The phoneme inventory is based on the phonology of Mandarin as described by (Lin, 2007) and (Duanmu, 2007) and tone markers are attached to the vowel of the syllable, rather than the

---

[4] For Japanese text written in Romanji, as is the case in CHILDES, we use phonemizer with the the Segments backend (Forkel et al., 2019).

[5] For English, Epitran uses the Flite Speech Sythesis System (Black and Lenzo, 2001) and for Simplified and Traditional Chinese it uses the CC-CEDict dictionary (`https://cc-cedict.org`).

end of the syllable. The tool only converts individual pinyin syllables, so our wrapper first splits the input into syllables before using the tool to convert each syllable to IPA.

`pingyam`: Pingyam[6] is a table storing conversion information between the various romanization systems of Cantonese (including IPA) based on data from the Open Cantonese Dictionary.[7] Our wrapper converts from the Jyutping system to IPA by first splitting the input text into syllables before using the table to convert each syllable to IPA. For consistency with `pinyin-to-ipa`, we move tone markers to the vowel of each syllable.

Although `pinyin-to-ipa` and `pingyam` only support one Chinese language each, we include them as backends because `epitran` and `phonemizer` have relatively poor G2P quality for these languages. This has prevented Chinese languages from being included in previous cross-lingual phonemic datasets (Ahn and Chodroff, 2022) and has led to them being disregarded in cross-lingual analysis (Pimentel et al., 2020). We hope that by including these backends, we address this gap. We also combine tone markers with their preceding phoneme to create a unique token (e.g., `a˥` is a single token, not two). We thus treat tone markers as phonological features rather than as individual phonemes, similar to how diphthongs are unique phonemes. However, this decision is still debatable and does lead to a comparatively larger phonemic vocabulary, so we provide an option to disable this merging (see appendix D).

3.2 Phoneme inventory validation

In order to validate the set of phonemes produced by each choice of backend and language, we compare the output to the phoneme inventories for that language listed in Phoible, a database containing phoneme inventories extracted from source documents and tertiary databases for 2186 distinct languages (Moran and McCloy, 2019).

Phoible also contains typological data and phonological feature information for each phoneme, a useful resource for phonological analysis. As there are often multiple inventories in Phoible for each language, we choose the inventory that best matches the output phoneme of all backends that supports that language, according to the number of phoneme types, the number of consonants, the number of vowels and the number of diphthongs.

Once the best inventory has been found, we use a process called *folding* to align the output phoneme set with the inventory and correct errors in the output. This is achieved a manually-crafted look-up table (a *folding map*) which is applied to the output of the G2P wrapper. These maps are primarily used to solve surface-level errors, instances where the G2P tool outputs a specific Unicode string for a specific phoneme but the inventory lists a different string. For example, the `phonemizer` backend with the `ja` language code (Japanese) outputs the tied characters t͡s as one of the phonemes, but the Japanese inventory lists ts instead. These errors can be solved with a simple one-to-one mapping. These mappings will not affect the information-theoretic properties of the output but do allow the output symbols to be matched with entries in Phoible.

Besides these surface-level errors, other transcription errors can also be solved with folding maps. For example, the `epitran` backend for Serbian always outputs d ʒ as two phonemes instead of the single phoneme dʒ, which can also be solved with a single mapping. The construction of the folding maps and these additional error types are discussed further in appendix F.

3.3 Qualitative Analysis

In fig. 2, we compare the matching Phoible inventory for French to the output of `G2P+` (using `phonemizer` as a backend) and the outputs produced by `phonemizer` and `epitran` when applied to the French section of CHILDES. The outputs of `phonemizer` and `epitran` both differ considerably from the inventory and from each other whereas the `G2P+` only fails to produce a single phoneme, ɥ, and produces two additional phonemes dʒ and tʃ, which we allow as they come from loanwords such as “pizza” and “sandwich”.

4 IPA CHILDES

IPA CHILDES contains 45 million words of monolingual child-centered speech for 31 languages. The data is sorted by child age in order to support curriculum learning experiments, such as in the work of Huebner et al. (2021), and we also provide an ‘is_child’ feature to allow for filtering child or adult utterances.

In order to create the dataset, we first download all monolingual and non-SLI corpora in CHILDES.

---

[6] `https://github.com/kfcd/pingyam`

[7] `https://www.kaifangcidian.com/han/yue/`

CHILDES has 48 languages but only 31 are supported by a backend in `G2P+` (either because the language is not supported, or because they have been transcribed using an irregular script). For languages supported by multiple backends, we produce a sample transcription using each backend and carefully examine the output. The 'best-fitting' backend (the one that produces a phonemic vocabulary closest to one of the inventories in Phoible) is selected and is the backend for which we produce a folding map, as described in section 3.2. Having selected the best backend, we use `G2P+` to convert all orthographic utterances for each language to a phonemic representation, producing a CSV containing the original representation, the phonemic representation as well as additional data stored in CHILDES (such as target child age, morpheme count, part of speech information, and the IDs of each utterance, transcript, corpus and collection).

An illustration of the dataset is given in fig. 1 and a description of each language section is given in appendix B, detailing the matching Phoible inventory and CHILDES section for each language. Note that English is divided into British English (EnglishUK) and North American English (EnglishNA) to mirror the split present in CHILDES and Portuguese is also split into European and Brazilian varieties, following previous work (Caines et al., 2019; Goriely et al., 2023). For these splits, we use different `phonemizer` accents. Data is not uniformly distributed across languages. EnglishNA is the most represented, with close to 10 million words, and Farsi is the least represented, with only 43 thousand words. We discuss limitations of the dataset in appendix A.

5 Cross-Lingual Phoneme LMs

Phoneme LMs trained on developmentally plausible corpora allow for the testing of phonological representations but recent work has only explored English models trained on 10 – 100 million words (see section 2.3). Here, we establish the size requirements for models trained on data available in IPA CHILDES and then demonstrate how models trained on the 11 largest languages in our dataset can be used to explore emergent phonology.

Each of our models are auto-regressive, trained to predict phonemes in a sequence. This is similar to how standard auto-regressive models are trained, except that each token represents a single phoneme, rather than a word or subword. We refer to the suite of models as "cross-lingual" as each individual model is monolingual, only trained on data from a single language. This is in contrast to "multilingual" models that are trained on multiple languages at once.

5.1 Size Requirements of Phoneme LMs

We use the BabySLM benchmark (Lavechin et al., 2023) to evaluate syntactic and phonological knowledge. The *syntactic* score is calculated using a preference task over pairs of grammatical and ungrammatical sentences across six syntactic phenomena commonly seen in naturalistic speech. For example, models should assign ð ə ɡ ʊ d k ɪ t i ("the good kitty") a higher likelihood than ð ə k ɪ t i ɡ ʊ d ("the kitty good"). The *lexical* score is similarly calculated using minimal pairs of words and pseudowords, such as ɹ uː l ə ɹ z ("rulers") compared to the pseudo-word m uː k ə ɹ z ("mukers"). Lavechin et al. (2023) demonstrated that an LSTM model trained on 1.2 million words from Providence (one of the corpora in CHILDES) achieved a lexical score of 75.2 and a syntactic score of 55.1[8]. Goriely et al. (2024) later achieved lexical and syntactic scores of 87.8 and 83.9 when training a larger transformer-based model on the 100-million-word BabyLM challenge dataset (Hu et al., 2024a).

Here, we use IPA CHILDES and BabySLM to establish the scaling laws of phoneme LMs in terms of data size and model size. We subsample the EnglishNA portion of the dataset, remove word boundaries and child-produced utterances and train a suite of GPT-2 models ranging from 400 thousand to 19 million non-embedding parameters. To prevent overfitting, we train three models for each combination of model size and data size using dropouts of 0.1, 0.3 and 0.5, selecting the model with the lowest perplexity for each. Model parameters, training configurations and scripts are provided in appendix G.

The scaling graphs for the lexical and syntactic scores are given in fig. 3. For every model size, performance increases with more training data but for a particular data size the largest model is not always the best. For instance, the second smallest model is the best choice for the lexical task if only 300 thousand tokens of data are available, likely due to larger models overfitting with a sample this small (even with high dropout). It is also clear that

---

[8] Chance performance for both BabySLM scores is 50 and 100 indicates perfect performance

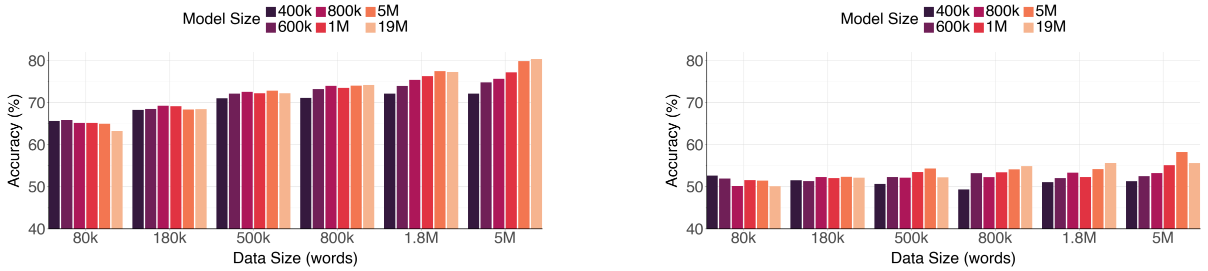


Figure 3: BabySLM lexical score (left) and syntactic score (right) achieved by a phoneme-based GPT-2 model trained on the EnglishNA portion of IPA CHILDES across model sizes and subsample sizes.

although small models with very little data seem to acquire phonological knowledge (as measured by the lexical score), much more data is required to achieve syntactic scores past 60, in line with the results of Lavechin et al. (2023) and Goriely et al. (2024). The best model parameters for each score and data size are given in appendix H.

5.2 Probing for Phonological Features

As the phonemic utterances in IPA CHILDES maintain a correspondence with Phoible, we can use the **distinctive feature** information in Phoible to probe cross-lingual phoneme LMs for phonological knowledge.

We select the 11 largest languages in the dataset and train a GPT-2 model on each, subsampling 500 thousand words[9] and using the best-fitting model for this data size according to the previous experiment (the 5-million-parameter model with a dropout of 0.3). The training configuration remains the same (see appendix G). These models allow us to compute contextual embeddings $c(x)$ for phonemes.

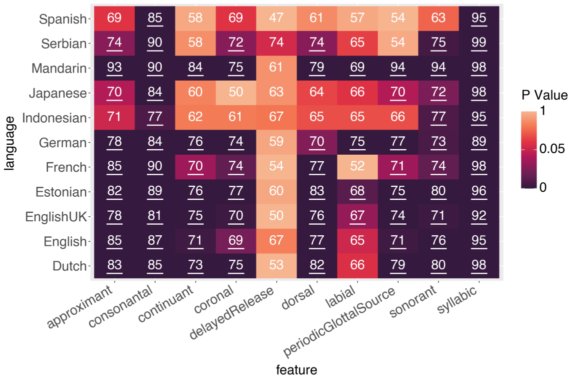


Figure 4: Accuracy of the phonological distinctive feature probe across 11 languages in IPA CHILDES and 9 distinctive features from Phoible.

We then look up the distinctive features of each phoneme in each language using the matching inventories in Phoible (see table 1). We find the set of features for which, in all 11 languages, there are at least 4 phonemes that exhibit the feature and 4 that do not. For each feature f, we train a linear probe p_f to predict that feature from the contextual embeddings $c(x)$ of phonemes. Each probe is trained with an equal number of positive and negative examples and is evaluated using leave-one-group-out cross-validation (i.e for each phoneme x in the phoneme inventory V, the probe is trained on the contextual embeddings of all other phonemes $\{c(y) | y \in V \setminus \{x\}\}$, then evaluated by predicting the feature from contextual embeddings of the left-out phoneme $p_f(c(x))$, and the final score is a macro-average across all phonemes $x \in V$).

The results of each probe are provided in fig. 4. The majority of the probes achieve accuracies significantly[10] higher than chance (50%), indicating that the models learn representations that encode distinctive features. While the scores for each feature are broadly consistent across languages, some notable differences emerge. For example, nearly all feature probes achieve statistically significant results in Mandarin, whereas only two do so in Spanish. This disparity can be partly attributed to the number of unique phonemes in each language. Because we treat each combination of vowel and tone as a distinct phoneme, Mandarin has 99 phoneme types, compared to just 24 in Spanish. The smaller phoneme inventory in Spanish greatly reduces n for each probe, making it more challenging to obtain

---

[9] As the number of phonemes per word varies across these languages, we actually subsample 1.8 million tokens (phonemes) for each language, which is roughly 500 thousand words.

[10] Statistical significance was assessed using a binomial test, where the null hypothesis assumes a probability of success $p_0 = 0.5$ and the number of trials n is equal to the number of phonemes tested by the probe. A result was considered significant if the computed p-value was less than 0.05.

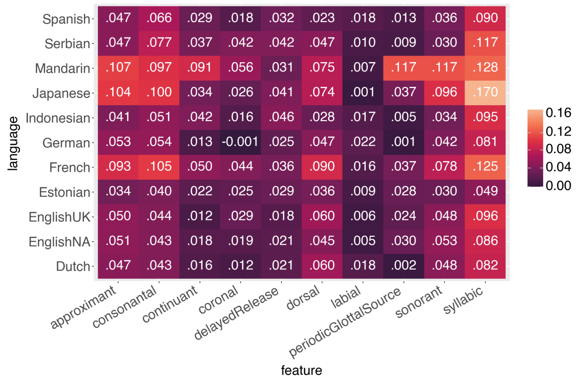


Figure 5: Average silhouette scores when using each distinctive feature to cluster contextual embeddings of the phonemes in each language.

statistically significant results.

In all 11 languages, the highest result is achieved by the probe for the ‘syllabic’ feature which generally[11] separates vowels from consonants. As these models only learn to predict phonemes and have no concept of how each phoneme is pronounced, the fact that this separation is learned clearly indicates that vowels and consonants provide a strong distributional signal across languages. The `consonantal` feature similarly separates vowels from consonants[12] and is learned by a probe in every language. However, not every feature can be learned by these probes. For instance, the `delayedRelease` feature, which distinguishes stops from affricates, is not learned by any probe. Our models do not encode the rate of phoneme delivery, so it is unsurprising that a feature that relates to the temporal properties of phonemes is difficult to probe.

Distributional Phoneme Clusters

To better understand why the probes capture certain phonological features, we examine whether contextual embeddings cluster according to these features. For each language, we sample 50 contextual embeddings per phoneme and label them with their associated phonological features. For each labeling, we then compute the **silhouette score** for each embedding — a metric ranging from –1 to 1, where higher values indicate that an embedding is more similar to others in its assigned cluster than to those in neighboring clusters (Rousseeuw, 1987). Averaging these scores across all embeddings allows us to compare how well different features cluster the phoneme representations, as shown in fig. 5.

The scores are all relatively close to zero, likely due to the curse of dimensionality — our embeddings have 256 dimensions, far exceeding the number of distinct phonemes in each language. Despite this, the results are consistent with the probe findings: the syllabic feature yields the highest clustering quality.

We further visualize this clustering using dendrograms, created by averaging the contextual embeddings for each phoneme and applying an incremental clustering algorithm. Figure 6 shows examples for Japanese and French, with the syllabic feature marked for each phoneme. In both cases, vowels are almost entirely separated from consonants, with one notable exception: n in Japanese. We also observe some alignment with traditional phoneme groupings (e.g., b and p), though overall the dendrograms diverge from standard phonological classifications. This suggests that the distributional behavior of phonemes in context may not neatly align with their articulatory or categorical properties.

6 Discussion

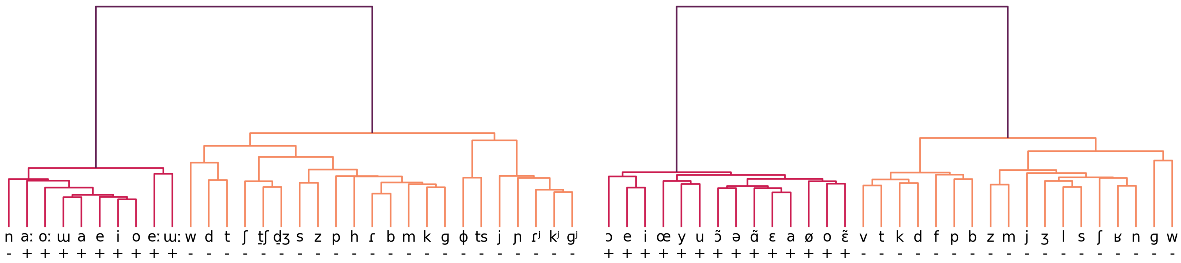
IPA CHILDES addresses several limitations of past datasets, as the first large multilingual corpus of child-centered phonemic speech. In this study we demonstrate how this data can be used to train phoneme LMs, but this dataset could also support information-theoretic studies of language processing and acquisition, which have previously based their calculations on word types (Piantadosi et al., 2011; Dautriche et al., 2017a; Pimentel et al., 2020) or orthographic text (Mahowald et al., 2013; Dautriche et al., 2017b; Futrell et al., 2020), often citing a lack of phonemic data as a limiting factor. The child-centered domain of our dataset could also be beneficial for studying the ‘Goldilocks’ hypothesis (Kidd et al., 2014) and the properties of ‘Parentese’ (Ramírez-Esparza et al., 2017). We provide an example of an experiment investigating the later in appendix I, where we compute the average information of utterances directed to children aged 0–6 across 10 languages and find a general trend of increasing informative content.

Our `G2P+` tool also provides new avenues for linguistic analysis by ensuring that phonemes pro-

---

[11] In some languages there are also syllabic consonants, which like vowels can act as the nucleus of a syllable.

[12] This feature indicates an audible constriction of the vocal tract, separating obstruents, nasals, liquids, and trills from vowels, glides and laryngeal segments (Gussenhoven and Jacobs, 2017).


n aː oː ɯ a e i o eː ɯː w d t ʃ t̠ʃ d̠ʒ s z p h ɾ b m k g ɸ ts j ɲ ɾʲ kʲ gʲ
- + + + + + + + + + -
ɔ e i œ y u ɔ̃ ə ɑ̃ ɛ a ø o ɛ̃ v t k d f p b z m j ʒ l s ʃ ʁ n g w
+ + + + + + + + + + + + + + - - - - - - - - - - - - - - - - - -

(a) Japanese

(b) French

Figure 6: Similarity of the contextual embeddings for each phoneme learned by the Japanese and French phoneme LMs. Similarities are computed using Euclidean distance considering the average of 50 contextual embeddings for each phoneme and linkages are created using the incremental algorithm. The 'syllabic' distinctive feature is marked below each phoneme.

duced for each language are consistent with established inventories in Phoible. This not only addresses transcription errors, but also allows for the use of distinctive feature information provided by Phoible in analysis. We demonstrate this by training linear probes to extract distinctive features from the contextual embeddings of phonemes learned by our monolingual models. We find that certain features (e.g. `consonantal`) emerge solely from the distributional properties across all 11 languages, while others (e.g. `delayedRelease`) do not.

Our resources could also support the training of self-supervised speech models (e.g. Hsu et al., 2021). These models are trained directly on audio and lag behind phoneme or text-based models, often requiring several orders of magnitude more data to learn semantic representations (Cuervo and Marxer, 2024), but recent work has found that fine-tuning on phoneme classification can reduce this gap (Feng et al., 2023; Poli et al., 2024). Our work is closely related to recent efforts in low-resource cross-lingual language modeling — for example, the Goldfish suite of monolingual models spanning 350 languages, some trained on as little as 5MB of orthographic text (Chang et al., 2024). IPA is also a more universal representation than orthographic text, which varies considerably across languages, with multilingual IPA models proving to be effective for force-alignment (Zhu et al., 2024) and zero-shot cross-lingual NER (Sohn et al., 2024). In this study we only train monolingual models, but future work could extend this to the multilingual setting.

7 Conclusion

This work introduces `G2P+` and IPA CHILDES, two new resources for phonological research. `G2P+` improves open-source G2P tools by ensuring phonemic vocabularies align with the established inventories in the Phoible database. Using this tool, we create IPA CHILDES by converting the orthographic transcriptions in CHILDES into phonemic representations, resulting in a large corpus of child-centered spontaneous speech across 31 languages.

We demonstrate the utility of these resources for phonological analysis using phoneme LMs by extending prior work to the cross-lingual setting. Our results establish the corpus size requirements for phoneme LMs trained on developmentally plausible corpora and we show that models trained on 11 languages effectively implicitely encode distinctive features. These findings support the role of phoneme LMs in studying emergent phonology. We anticipate that `G2P+` and IPA CHILDES will enable a wide range of future studies in linguistics and NLP, particularly in phonological acquisition, cross-linguistic analysis, and speech processing.

Acknowledgments

We thank Lisa Beinborn for her advice on an an early draft of this article. We are also grateful to Pietro Lesci and Suchir Salhan for their feedback.

Our experiments were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. Zébulon Goriely is supported by an EPSRC DTP Studentship.

References

Emily Ahn and Eleanor Chodroff. 2022. VoxCommunis: A corpus for cross-linguistic phonetic analysis. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5286–5294, Marseille, France. European Language Resources Association.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. 2022. Data scaling laws in NMT: The effect of noise and architecture. In *International Conference on Machine Learning*, pages 1466–1482. PMLR.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing cognitive plausibility of subword tokenization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4478–4486, Singapore. Association for Computational Linguistics.

Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to phones transcription for multiple languages in python. *Journal of Open Source Software*, 6(68):3958.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430.

Alan W Black. 2019. Cmu wilderness multilingual speech dataset. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5971–5975.

Alan W Black and Kevin A Lenzo. 2001. Flite: a small fast run-time synthesis engine. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: Battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64.

Andrew Caines, Emma Altmann-Richer, and Paula Buttery. 2019. The cross-linguistic performance of word segmentation models over time. *Journal of child language*, 46(6):1169–1201.

Luca Capone, Alice Suozzi, Gianluca E Lebani, Alessandro Lenci, et al. 2024. Babies: A benchmark for the linguistic evaluation of italian baby language models.

Tyler A Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K Bergen. 2024. Goldfish: Monolingual language models for 350 languages. *arXiv preprint arXiv:2408.10441*.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. Gpt or bert: why not both? *arXiv preprint arXiv:2410.24159*.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. Call for papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Christopher Cieri, David Miller, and Kevin Walker. 2004. The Fisher corpus: A resource for the next generations of speech-to-text. In *LREC*, volume 4, pages 69–71.

John Coleman, Ladan Baghai-Ravary, John Pybus, and Sergio Grau. 2012. Audio BNC: the audio edition of the spoken british national corpus. *Phonetics Laboratory, University of Oxford*.

John Coleman, Mark Liberman, Greg Kochanski, Lou Burnard, and Jiahong Yuan. 2011. Mining a year of speech. *VLSP 2011: New tools and methods for very-large-scale phonetics research*, pages 16–19.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2023. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 798–805.

BNC Consortium. 2007. British National Corpus, XML edition. Literary and Linguistic Data Service.

Santiago Cuervo and Ricard Marxer. 2024. Scaling properties of speech language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 351–361.

Isabelle Dautriche, Kyle Mahowald, Edward Gibson, Anne Christophe, and Steven T Piantadosi. 2017a. Words cluster phonetically beyond phonotactic regularities. *Cognition*, 163:128–145.

Isabelle Dautriche, Kyle Mahowald, Edward Gibson, and Steven T Piantadosi. 2017b. Wordform similarity increases with semantic similarity: An analysis of 100 languages. *Cognitive science*, 41(8):2149–2169.

Richard Diehl Martinez, Hope McGovern, Zebulon Goriely, Christopher Davis, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2023. CLIMB – curriculum learning for infant-inspired model building. In

Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning, pages 84–99, Singapore. Association for Computational Linguistics.

Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. 2024. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*.

San Duanmu. 2007. *The phonology of standard Chinese*. Oxford University Press.

Ewan Dunbar, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux. 2021. The zero resource speech challenge 2021: Spoken language modelling. In *Proc. Interspeech 2021*, pages 1574–1578.

R.H. Dunn and V. Vitolins. 2022. eSpeak NG speech synthesizer. In GitHub respository (Version 1.51).

S Elizabeth Eden. 2018. *Measuring phonological distance between languages*. Ph.D. thesis, UCL (University College London).

Siyuan Feng, Ming Tu, Rui Xia, Chuanzeng Huang, and Yuxuan Wang. 2023. Language-universal phonetic representation in multilingual speech pretraining for low-resource speech recognition. In *INTERSPEECH 2023*, Dublin, Ireland. ISCA.

Robert Forkel, Steven Moran, Johann-Mattis List, Simon J Greenhill, Lucas Ashby, Kyle Gorman, and Gereon Kaiping. 2019. cldf/segments: Unicode standard tokenization.

Richard Futrell, Edward Gibson, and Roger P Levy. 2020. Lossy-context surprisal: An information-theoretic model of memory effects in sentence processing. *Cognitive science*, 44(3):e12814.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. 1993. DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. *NASA STI/Recon technical report n*, 93:27403.

Lucas Georges Gabriel Charpentier and David Samuel. 2023. Not all layers are equally as important: Every layer counts BERT. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 238–252, Singapore. Association for Computational Linguistics.

Mario Giulianelli, Luca Malagutti, Juan Luis Gastaldi, Brian DuSell, Tim Vieira, and Ryan Cotterell. 2024. On the proper treatment of tokenization in psycholinguistics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18556–18572, Miami, Florida, USA. Association for Computational Linguistics.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Zébulon Goriely, Andrew Caines, and Paula Buttery. 2023. Word segmentation from transcriptions of child-directed speech using lexical and sub-lexical cues. *Journal of Child Language*, pages 1–41.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Steven Greenberg, Joy Hollenback, and Dan Ellis. 1996. Insights into spoken language gleaned from phonetic transcription of the switchboard corpus. In *Proc. ICSLP*, volume 96, pages 24–27.

Carlos Gussenhoven and Haike Jacobs. 2017. *Understanding phonology*. Routledge.

M. P Harper. 2011. The IARPA Babel multilingual speech database. Accessed: 2020-05-01.

Mark Hasegawa-Johnson, Leanne Rolston, Camille Goudeseune, Gina-Anne Levow, and Katrin Kirchhoff. 2020. Grapheme-to-phoneme transduction for cross-language asr. In *Statistical Language and Speech Processing*, pages 3–19, Cham. Springer International Publishing.

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

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Leshem Choshen, Ryan Cotterell, Alex Warstadt, and Ethan Gotlieb Wilcox, editors. 2024a. *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Miami, FL, USA.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and

Ethan Gotlieb Wilcox. 2024b. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Philip A. Huebner, Elior Sulem, Fisher Cynthia, and Dan Roth. 2021. BabyBERTa: Learning more grammar with small-scale child-directed language. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 624–646, Online. Association for Computational Linguistics.

Daniel Jones. 2011. *Cambridge English pronouncing dictionary with CD-ROM*. Cambridge University Press.

J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazare, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Libri-Light: A benchmark for ASR with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, page 7669–7673. IEEE.

Herman Kamper, Aren Jansen, and Sharon Goldwater. 2017. A segmental framework for fully-unsupervised large-vocabulary speech recognition. *Computer Speech & Language*, 46:154–174.

Celeste Kidd, Steven T Piantadosi, and Richard N Aslin. 2014. The goldilocks effect in infant auditory attention. *Child development*, 85(5):1795–1804.

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.

Lori F Lamel, Robert H Kassel, and Stephanie Seneff. 1989. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *Proc. SIOA 1989*, pages Vol–2.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. BabySLM: language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*, pages 4588–4592, Dublin, Ireland. ISCA.

Colin Leong and Daniel Whitenack. 2022. Phone-ing it in: Towards flexible multi-modal language model training by phonetic representations of data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5306–5315, Dublin, Ireland. Association for Computational Linguistics.

Yinghao Aaron Li, Cong Han, Xilin Jiang, and Nima Mesgarani. 2023. Phoneme-level BERT for enhanced prosody of text-to-speech with grapheme predictions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Yen-Hwei Lin. 2007. *The Sounds of Chinese*. Cambridge University Press.

Brian MacWhinney. 2019. Understanding spoken language through TalkBank. *Behavior Research Methods*, 51(4):1919–1927.

Brian MacWhinney and Catherine Snow. 1985. The Child Language Data Exchange System. *Journal of Child Language*, 12(2):271–295.

Kyle Mahowald, Evelina Fedorenko, Steven T Piantadosi, and Edward Gibson. 2013. Info/information theory: Speakers choose shorter words in predictive contexts. *Cognition*, 126(2):313–318.

Connor Mayer. 2020. An algorithm for learning phonological classes from distributional similarity. *Phonology*, 37(1):91–131.

Mark Mazumder, Sharad Chitlangia, Colby Banbury, Yiping Kang, Juan Manuel Ciro, Keith Achorn, Daniel Galvez, Mark Sabini, Peter Mattson, David Kanter, Greg Diamos, Pete Warden, Josh Meyer, and Vijay Janapa Reddi. 2021. Multilingual spoken words corpus. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

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. In *NeuRIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*.

Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose. 2016. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938.

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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Steven T Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences*, 108(9):3526–3529.

Tiago Pimentel, Brian Roark, and Ryan Cotterell. 2020. Phonotactic complexity and its trade-offs. *Transactions of the Association for Computational Linguistics*, 8:1–18.

Mark A Pitt, Laura Dilley, Keith Johnson, Scott Kiesling, William Raymond, Elizabeth Hume, and Eric Fosler-Lussier. 2007. Buckeye corpus of conversational speech (2nd release). *Columbus, OH: Department of Psychology, Ohio State University*, pages 265–270.

Mark A. Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond. 2005. The buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability. *Speech Communication*, 45(1):89–95.

Maxime Poli, Emmanuel Chemla, and Emmanuel Dupoux. 2024. Improving spoken language modeling with phoneme classification: A simple fine-tuning approach. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5284–5292.

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

Nairán Ramírez-Esparza, Adrián García-Sierra, and Patricia K Kuhl. 2017. Look who's talking NOW! Parentese speech, social context, and language development across time. *Frontiers in psychology*, 8:1008.

Nan Bernstein Ratner. 2024. Augmenting clinical insights with computing: How talkbank has impacted assessment and treatment of speech and language disorders. *Language Teaching Research Quarterly*, 44:31–40.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

Elizabeth Salesky, Eleanor Chodroff, Tiago Pimentel, Matthew Wiesner, Ryan Cotterell, Alan W Black, and Jason Eisner. 2020. A corpus for large-scale phonetic typology. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4526–4546, Online. Association for Computational Linguistics.

Suchir Salhan, Richard Diehl Martinez, Zébulon Goriely, and Paula Buttery. 2024. Less is more: Pre-training cross-lingual small-scale language models with cognitively-plausible curriculum learning strategies. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 174–188, Miami, FL, USA. Association for Computational Linguistics.

Tanja Schultz. 2002. Globalphone: a multilingual speech and text database developed at karlsruhe university. In *Interspeech*, volume 2, pages 345–348.

Zhewen Shen, Aditya Joshi, and Ruey-Cheng Chen. 2024. Bambino-lm:(bilingual-) human-inspired continual pre-training of babylm. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 1–7.

Jimin Sohn, Haeji Jung, Alex Cheng, Jooeon Kang, Yilin Du, and David R Mortensen. 2024. Zero-shot cross-lingual NER using phonemic representations for low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13595–13602.

Stefan Taubert. 2024. pinyin-to-ipa.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The penn treebank: an overview. *Treebanks: Building and using parsed corpora*, pages 5–22.

Jörgen Valk and Tanel Alumäe. 2021. Voxlingua107: A dataset for spoken language recognition. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 652–658.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Aditya Yadavalli, Alekhya Yadavalli, and Vera Tobin. 2023. SLABERT talk pretty one day: Modeling second language acquisition with BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11763–11777.

Aditya Yedetore, Tal Linzen, Robert Frank, and R. Thomas McCoy. 2023. How poor is the stimulus? evaluating hierarchical generalization in neural networks trained on child-directed speech. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9370–9393, Toronto, Canada. Association for Computational Linguistics.

J. Zhu, C. Zhang, and D. Jurgens. 2022. ByT5 model for massively multilingual grapheme-to-phoneme conversion. In *Proceedings of INTERSPEECH 2022*, pages 446–450.

Jian Zhu, Changbing Yang, Farhan Samir, and Jahurul Islam. 2024. The taste of IPA: Towards open-vocabulary keyword spotting and forced alignment in any language. 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 750–772, Mexico City, Mexico. Association for Computational Linguistics.

Victor W Zue and Stephanie Seneff. 1996. Transcription and alignment of the timit database. In *Recent Research Towards Advanced Man-Machine Interface Through Spoken Language*, pages 515–525. Elsevier.

A Limitations

We consider the following limitations of our work.

Phonemes as a representation of speech: While phonemic data more closely represents how words are pronounced compared to orthographic text (the degree of this difference varies between languages), phonemes are still abstract symbolic units which do not contain many of the detailed and continuous features of speech, such as prosody. They also abstract away from phones, which are detailed realizations of phonemes, representing the physical sound produced rather than a language-specific meaningful unit. When comparing modalities that may be close to the sensory signal available to infants for developmentally plausible language modeling, some researchers consider phonemic data to be as implausible as orthographic data (Lavechin et al., 2023) and instead create language models that can be trained directly on audio (Kamper et al., 2017; Nguyen et al., 2020; Hsu et al., 2021; Dunbar et al., 2021). Nevertheless, phonemes still provide a useful unit of analysis and are necessary for certain linguistic theories and information-theoretic calculations. While phones could offer another useful representation, they are even harder to source than phonemes.

G2P conversion inaccuracies: Despite improving G2P conversion by mapping to inventories in the Phoible database, there are still limitations with `G2P+`. Firstly, our method integrates existing G2P tools, which may vary in quality between languages. When converting each language in CHILDES, we selected the most appropriate backend for each language, in particular adding two backends to support G2P for Mandarin and Cantonese, but the quality may still vary. Many of the G2P tools for certain languages convert words individually, so we do not capture vowel reduction, allophonic variation or other differences found in natural speech. We also use a single accent for each language, losing inter-speaker variability. The `phonemizer` backend supports multiple accents for certain languages (here we use a different accent for EnglishNA and EnglishUK) and future work could try to maintain accent differences during grapheme-to-phoneme conversion, but this would require speaker information or audio, as was done during the creation of Audio BNC (Coleman et al., 2012). Finally, we note that G2P methods may not produce correct transcriptions for child-produced utterances, which are often corrected by the transcriber, especially for young infants. Initially we intended to distribute IPA CHILDES without child-produced utterances (and in this study only train models with the child-directed utterances) but as they might be useful in future research, we instead note this limitation.

Phoible inventories: Although the Phoible database collects established phonemic inventories and provides distinctive feature vectors, there are still often multiple phoneme inventories for a single language. This the exact phonemic inventory for a particular language is still a matter of debate among expert phonologists. When creating folding maps we choose the ‘best-fitting’ inventory to map to, as detailed in table 1, but we acknowledge that these inventories may not be exact.

Phoneme LMs: We train phoneme LMs on 11 languages from IPA CHILDES but the specific architecture we use is based on our scaling experiment for the EnglishNA model. Although we do not directly compare these LMs, we note the possibility that other parameters may have better suited the non-English languages. We were only able to conduct the scaling experiments for English due to the lack of phonological benchmarks for other languages but we hope that the release of IPA CHILDES facilitates further work in multilingual phonological evaluation of phoneme LMs.

Languages: Although our dataset is multilingual, there are still limitations in terms of language coverage. The languages are predominantly European and Asian, with no languages indigenous to the Americas, Australia or Africa. English is also still the dominant language of the dataset and the Farsi section is very small, only containing 43 thousand words. In creating this dataset, we were limited by the languages available in CHILDES. The languages in CHILDES we were not able to convert were Greek, Arabic, Hebrew, Thai, Georgian, Tamil, Taiwanese, Jamaican, Sesotho, Berber, Cree and Slovenian and Russian due to missing G2P backends or unsupported orthographies.

B Breakdown of IPA CHILDES

IPA CHILDES contains transcriptions of child-centered speech for 31 languages. Details of each language section are provided in table 1.

Language	CHILDES Collection	Backend	Inventory ID	Words	Phonemes	% Child
EnglishNA	EnglishNA (49)	`phonemizer`	2175	9,993,744	30,986,218	36
EnglishUK	EnglishUK (16)	`phonemizer`	2252	7,147,541	21,589,842	39
German	German (10)	`epitran`	2398	5,825,166	21,442,576	44
Japanese	Japanese (11)	`phonemizer`	2196	2,970,674	11,985,729	44
Indonesian	EastAsian/Indonesian (1)	`epitran`	1690	2,347,642	9,370,983	34
French	French (15)	`phonemizer`	2269	2,973,318	8,203,649	40
Spanish	Spanish (18)	`epitran`	164	2,183,992	7,742,550	46
Mandarin	Chinese/Mandarin (16)	`pinyin_to_ipa`	2457	2,264,518	6,605,913	39
Dutch	DutchAfricaans/Dutch (5)	`phonemizer`	2405	1,475,174	4,786,803	35
Polish	Slavic/Polish (2)	`phonemizer`	1046	1,042,841	4,361,797	63
Serbian	Slavic/Serbian (1)	`epitran`	2499	1,052,337	3,841,600	29
Estonian	Other/Estonian (9)	`phonemizer`	2181	843,189	3,429,228	45
Welsh	Celtic/Welsh (2)	`phonemizer`	2406	666,350	1,939,286	69
Cantonese	Chinese/Cantonese (2)	`pingyam`	2309	777,997	1,864,771	34
Swedish	Scandinavian/Swedish (3)	`phonemizer`	1150	581,451	1,782,692	45
PortuguesePt	Romance/Portuguese (4)	`phonemizer`	2206	499,522	1,538,408	39
Korean	EastAsian/Korean (3)	`phonemizer`	423	263,030	1,345,276	37
Italian	Romance/Italian (5)	`phonemizer`	1145	352,861	1,309,489	39
Croatian	Slavic/Croatian (1)	`epitran`	1139	305,112	1,109,696	39
Catalan	Romance/Catalan (6)	`phonemizer`	2555	319,726	1,084,594	36
Icelandic	Scandinavian/Icelandic (2)	`phonemizer`	2568	279,939	1,057,235	35
Basque	Other/Basque (2)	`phonemizer`	2161	230,500	942,725	49
Hungarian	Other/Hungarian (3)	`epitran`	2191	237,062	918,002	48
Danish	Scandinavian/Danish (1)	`phonemizer`	2265	275,170	824,314	42
Norwegian	Scandinavian/Norwegian (2)	`phonemizer`	499	227,856	729,649	43
PortugueseBr	Romance/Portuguese (2)	`phonemizer`	2207	174,845	577,865	44
Romanian	Romanian (3)	`phonemizer`	2443	152,465	537,669	43
Turkish	Other/Turkish (2)	`phonemizer`	2217	79,404	421,129	51
Irish	Celtic/Irish (2)	`phonemizer`	2521	105,867	338,425	34
Quechua	Other/Quechua (2)	`phonemizer`	104	46,848	281,478	40
Farsi	Other/Farsi (2)	`phonemizer`	516	43,432	178,523	40

Table 1: A breakdown of each language available in IPA CHILDES. The bracketed number in the **CHILDES Collection column** refers to the number of corpora downloaded from that collection. The **Backend**, **Lang Code** and **Phoneme Inventory** columns refer to the `G2P+` configuration used to convert utterances for that language to phonemes and the Phoible inventory used for that language in folding. The **Words** and **Phonemes** columns refer to the number of words and tokens in each subset and **% Child** refers to the percentage of the data that is spoken by a child.

C Dataset comparison

In section 2.1 we discuss previous phonemic datasets in relation to IPA CHILDES. We provide a full comparison of these datasets in table 2.

D `G2P+` Usage

`G2P+` is a python library that can be used as an API or as a command-line tool in order to convert orthographic text to a phonemic representation. The tool allows the user to select the backend and language code to use for G2P with text provided through filepaths or standard input. Additional options include `--keep_word_boundaries` to output a dedicated `WORD_BOUNDARY` token between words and `--uncorrected` to skip the folding process and output the phonemes exactly as produced by the backend tool. Each backend also supports individual options. For instance, `--split-tones` outputs tones as individual tokens instead of merging them with the syllabic phoneme for our two Chinese language backends. See the repository's `README.txt` for further details.

E Phoneme Stream Representation

In order to ensure that phonemes are output using a consistent representation, we define the **phoneme stream representation** as follows:

- Each phoneme is represented using the International Phonetic Alphabet (IPA).
- Each phoneme is separated by a space.
- Word boundaries and utterance boundaries are represented using unique symbols.

IPA is used to represent each phoneme due to being the most widely used and comprehensive phonetic alphabet. It is important to separate phonemes by spaces because IPA symbols may be represented

Dataset	Modality	Scale (words)	Domain	Languages
The Pile (Gao et al., 2020)	Orth	100B[†]	Web-scraped written text	English only
GlobalPhone (Schultz, 2002)	Orth, Phon, Audio	5M[†]	Read speech	22
CommonVoice (Ardila et al., 2020)	Orth, Audio	30M[†]	Read speech	38
VoxCommunis (Ahn and Chodroff, 2022)	Orth, Phon, Audio	23M[†]	Read speech	40
CMU Wilderness (Black, 2019)	Orth, Audio	170M[†]	Read speech	699
VoxClamantis (Salesky et al., 2020)	Orth, Audio, Phon	152M[†]	Read speech	635
TIMIT (Garofolo et al., 1993)	Orth, Phon, Audio	40k	Read speech	English only
FLEURS (Conneau et al., 2023)	Orth, Audio	15M[†]	Read speech	102
MSWC (Mazumder et al., 2021)	Orth, Audio	20M	Read speech	102
IPAPACK (Zhu et al., 2024)	Orth, Phon	15M[†]	Read speech	115
LibriSpeech (Panayotov et al., 2015)	Orth, Audio	10M[†]	Audio books	English only
Libri-Light (Kahn et al., 2020)	Orth,[*] Phon,[*] Audio	700M[†]	Audio books	English only
MLS (Pratap et al., 2020)	Orth,[*] Phon,[*] Audio	600M[†]	Audio books	8
Switchboard (Godfrey et al., 1992)	Orth, Phon, Audio	3M[†]	Telephone conversations	English only
Fisher (Cieri et al., 2004)	Orth, Audio	12M[†]	Telephone conversations	English only
Buckeye (Pitt et al., 2005)	Orth, Phon, Audio	300k	Spontaneous speech	English only
British National Corpus (Consortium, 2007)	Orth, Audio	100M	Written & spontaneous speech	English only
Audio BNC (Coleman et al., 2012)	Orth, Phon, Audio	7M	Spontaneous speech	English only
VoxLingua107 (Valk and Alumäe, 2021)	Audio	80M	Spontaneous speech	107
Babel (Harper, 2011)	Orth, Audio	60M	Telephone conversations	25
CHILDES (MacWhinney and Snow, 1985)	Orth	59M	Child-centered speech	45
BabyLM (Choshen et al., 2024)	Orth	100M	Speech and text[**]	English only
IPA CHILDES	Orth, Phon	45M	Child-centered speech	31

Table 2: A comparative summary of the datasets discussed in section 2.1. The datasets are described in terms of their modality, scale, domain and languages. IPA CHILDES is the first multilingual phonemic dataset of spontaneous speech and the first phonemic dataset of child-centered speech.

[†] *Word counts estimated from the size in bytes or the hours of audio in the dataset, using a heuristic based on the size of Switchboard of 5 bytes per word and 12,000 words per hour.*

[*] *Libri-Light and MLS only have orthographic and phonemic transcriptions for 10 hours of audio per language..*

[**] *BabyLM contains a mix of speech and text data from a mix of adult-directed and child-directed sources, only 29% is child-directed speech.*

using multiple Unicode characters. For instance, the word “enjoy” can be transcribed in IPA as ɛndʒɔɪ which uses six characters but only contains four phonemes, since dʒ is a single consonant and ɔɪ is a diphthong. By instead representing the word as ɛ n dʒ ɔɪ, it is much easier to split the word into individual phonemes by using whitespace as a delimiter. Similarly, word boundaries and utterance boundaries are represented using the unique symbols `WORD_BOUNDARY` and `UTT_BOUNDARY`.

F Folding Maps

Folding maps are primarily used to make surface-level adjustments, but they can also be used to solve several other error types in order to create a better alignment with a Phoible inventory. These errors are detailed in table 3.

The many-to-one mappings and those that split or merge tokens may alter the number of output tokens or types. Since such a mapping will change the information-theoretic properties of the output, it is important that they are linguistically motivated and carefully implemented.

In order to construct the folding map for each backend-language pair, we run `G2P+` on orthographic text for that language and compare the output set of phonemes P_O to the phonemes in the closest inventory in Phoible P_I. We call the set of phonemes present in P_O but not P_I the “unknown phonemes” U_K where $U_K = P_O \setminus P_I$ and the set of phonemes present in P_I but not P_O the “unseen phonemes” U_S where $U_S = P_I \setminus P_O$. We then construct the folding map as follows:

1. Find pairs $(k, s) \in U_K \times U_S$ that differ according to an accent or diacritic and obviously represent the same phoneme (determined by ruling out alternatives or examining where k is produced in the output). Create a one-to-one mapping $k : s$ for each such pair, e.g. $\mathrm{t} : \mathrm{t}^{\mathrm{h}}$.

2. Find pairs $(k, s) \in U_K \times U_S$ that clearly repre-

Error type	**Consequence**	**Example**
One-to-one: The backend uses one symbol for a phoneme but the inventory lists a different symbol for that phoneme.	The one-to-one mapping does not change the number of types or tokens in the output.	`phonemizer` with language code `sv` (Swedish) outputs n but the matching inventory uses n̪.
Many-to-one: The backend produces two different phonemes that should only map to a single phoneme in the inventory.	The many-to-one mapping reduces the number of phoneme types.	`phonemizer` with language code `pt` (Portuguese) outputs both ɹ and r but the matching inventory only lists ʁ.
Consonant merging: The backend outputs two symbols for a consonant that should be written as a single phoneme.	The mapping merges the pair of consonants, reducing the number of phoneme tokens produced.	`epitran` with language code `srp-Latn` (Serbian) outputs the sequence d ʒ but these are should be written as a single phoneme dʒ.
Vowel merging: The backend outputs a pair of vowels as separate phonemes but they are typically analysed as a single diphthong.	The mapping merges the pair of vowels, reducing the number of phoneme tokens produced.	`pingyam` with language code `cantonese` outputs the sequence o u but these are should be treated as a diphthong ou.
Vowel splitting: The backend outputs a diphthong that is not listed in the inventory and should be split into individual phonemes.	The mapping splits the pair of vowels, increasing the number of phoneme tokens produced.	`phonemizer` with language code `en-us` (North American English) outputs aɪʊ as a single phoneme but this should be aɪ ʊ.
Phoneme duplication: The backend outputs duplicate phonemes to represent long vowels or consonants or because of an error.	The mapping replaces the pair of phonemes with just one, reducing the number of phoneme tokens.	`phonemizer` with language code `et` (Estonian) outputs d d but should output the long consonant dː.
Diacritic error: The backend incorrectly outputs the diacritic as a separate symbol instead of attaching it to the phoneme.	The mapping may change the number of phoneme types or tokens.	`phonemizer` with language code `ko` (Korean) outputs the diacritic for aspiration as h instead of ^{h} so sequences kh and ph are mapped to k^{h} and p^{h}.
Orthographic error: Due to an invalid symbol in the orthographic text, the backend outputs an incorrect phoneme.	The contextual mapping changes the frequency statistics for the resulting phoneme, possibly reducing the number of phoneme types.	`epitran` with language code `hun-Latn` (Hungarian) outputs ô when the orthographic letter ő is incorrectly written as ô and so the phoneme is mapped to øː.

Table 3: A list of errors that can occur during grapheme-to-phoneme conversion that can be fixed with a folding map but that may change the information-theoretic properties of the output.

sent the same phoneme (determined as above) but may use entirely different symbols, possibly due to an alternative transcription scheme. Create a one-to-one mapping for each pair, e.g. a : æ.

3. For remaining items $k \in U_K$, determine whether these result from one of the other errors in table 3. Carefully examine instances where k is produced in the output and create a suitable mapping $k : p$ for some $p \in P_I$ to solve the error (the mapping may need to be contextual or include several characters, e.g. ɚ : ə ɹ or ʊ ɔ : w ɔ).

4. For remaining items $s \in U_S$, determine whether these result from one of the other errors in table 3. Carefully examine instances where s should be produced in the output and create a suitable mapping $k : s$ for some $k \in P_O$ to solve the error (the mapping may need to be contextual or include several characters).

5. Examine the output for cases of **phoneme duplication** and other errors that may not contain phonemes in U_K or U_S but could still be solved with the phoneme map and create suitable mappings.

The goal is for $U_K = \{\} = U_S$ or equivalently $P_I = P_O$, i.e the set of phonemes produced by the tool perfectly aligns with the phoneme inventory in Phoible. This is not always possible, often there are a few remaining phonemes in U_K and/or U_S. This can occur when no obvious mappings could be found in steps 1–4 above. For example, the `epitran` backend for German does not produce the phoneme ʒ (it is "unseen") and none of the unknown phonemes seem to be a good match. Another possibility is that the output set of phonemes P_O may not align well with any of the Phoible phoneme inventories and so the closest match may not include some of the unknown phonemes $k \in U_K$ despite being valid phonemes for that language and listed in other inventories. For example, the `epitran` backend for German produce the phonemes x and ɐ which are not listed in the matching inventory but are listed in other established inventories for German. In other cases, the unknown phonemes may come from loan words (e.g. ts for "pizza" in Portuguese). Finally, there are some cases where the output considerably disagrees with all of the Phoible inventories but is a valid phonemic analysis of the language according to other sources.

See section 3.3 for an example of using `G2P+` for French, using the `phonemizer` backend with a folding map to approach Phoible inventory 2269.

G Implementation Details

We conduct our experiments using the `PyTorch` framework (Paszke et al., 2019) and the `Transformers` library (Wolf et al., 2020).

G.1 Hardware Details

We use a server with one NVIDIA A100 80GB PCIe GPU, 32 CPUs, and 32 GB of RAM for all experiments. Below, we report a subset of the output of the *lscpu* command:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical,
                       48 bits virtual
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R)
                       Silver 4210R CPU
                       @ 2.40GHz
CPU family:            6
Model:                 85
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             8
Stepping:              7
BogoMIPS:              4800.11
```

G.2 Model Parameters and Training Procedure

Parameter	Value
Max Example Length	128
Learning Rate	0.001
Optimizer	AdamW
Scheduler Type	Linear
Max Steps	200k
Warm-up Steps	60k
Per Device Batch Size	32

Table 4: Hyperparameter settings for training the GPT-2 architecture. Where values are not reported, they may be assumed to be default values.

We describe training parameters in table 4 and model sizes in table 5. Following the conventions of the Pythia suite of models (Biderman et al., 2023), we report the number of non-embedding parameters. Unlike their suite, where models are named according to the number of parameters, we name our models according to the number of non-embedding parameters. This is because we use the same architecture for multiple languages, each of which has a different vocabulary size according to the number of phoneme types in that language, which alters the total number of parameters. Our 1M, 19M and 85M models are equivalent to Pythia-14M, Pythia-70M and Pythia-160M, respectively. Our training scripts are available here.

Model Size	Layers	Heads	Embd	Inner
400k	2	4	128	512
600k	3	4	128	512
800k	4	4	128	512
1M	6	4	128	512
5M	6	8	256	1024
19M	6	8	512	2048
25M	8	8	512	2048
85M	12	12	768	3072

Table 5: GPT-2 model sizes used in the size requirement experiment. Where values are not reported, they may be assumed to be default values.

Data is prepared into batches by first tokenizing the entire dataset, combining all tokens into one long vector, and then splitting the vector into chunks of 128 tokens. Only the very last example is padded, if required. At each step during training, random chunks are selected and combined into batches.

Checkpoints are taken every 20,000 steps during training. At each checkpoint, the perplexity is evaluated on the held-back evaluation set, and at the end of training the checkpoint with the lowest perplexity is returned as the best model. For the smallest models, many of the best models were from the very first checkpoint, since due to the small training dataset and small model, the model had already fit the data by this point.

In our size requirement experiment (see section 5.1), we train each model in table 5 using a dropout of 0.1, 0.3 and 0.5 on each subset size of the EnglishNA portion of IPA CHILDES.

H Best Phoneme LM Parameters Across Data Scales

Following the size experiment in section 5.1, we report the model size and dropout values that achieved the highest BabySLM scores for each subsample size of the EnglishNA portion of IPA CHILDES in table 6.

I Average Information Density of Phonemized Child-Directed Speech Increases with Age Cross-Lingually

The phonemic representation of the utterances in our dataset open up new avenues for exploring

Data Size (words)	BabySLM Lexical Model Size	Dropout	Score	BabySLM Syntactic Model Size	Dropout	Score
80k	600k	0.3	65.8	400k	0.5	52.6
180k	800k	0.3	69.3	5M	0.5	52.3
500k	5M	0.3	72.9	5M	0.3	54.3
800k	19M	0.5	74.2	19M	0.1	54.9
1.8M	5M	0.3	77.4	19M	0.1	55.6
5M	19M	0.1	80.3	5M	0.3	58.3

Table 6: Best model sizes and dropout values for the BabySLM Lexical and Syntactic scores for each subset size of the EnglishNA corpus of IPA CHILDES.

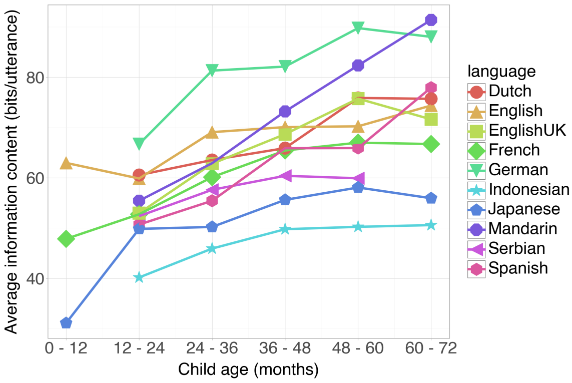


Figure 7: Average information of child-directed utterances in CHILDES

the phonotactic properties of languages and the information-theoretic properties of child-directed speech.

Here, we demonstrate one information-theoretic experiment, comparing the average information content of child-directed utterances to the age of the child being spoken to (this information is also available in CHILDES and is preserved in our dataset). We group child ages in years (0-12 months, 12-24 months, etc.) and calculate the average information content of a sample of child-directed utterances using a unigram language model. The information I_U of each utterance consisting of a sequence of phonemes $p_1, p_2, \ldots, p_n$ is given by

$$I_U = -\sum_{i=0}^{n} log_2 P(p_i),$$

where $P(p_i)$ is the probability of phoneme p_i given by its frequency in the data. We plot the average information of utterances in each age category for the largest 10 languages in the dataset in fig. 7. We find that across all 10 languages the average information of utterances increases with the age of the child, indicating that speakers of ‘Parentese’ may adjust the complexity of their speech according to the learner’s age.

BabyLM's First Words:
Word Segmentation as a Phonological Probing Task

Zébulon Goriely   **Paula Buttery**
Department of Computer Science & Technology, University of Cambridge, U.K.
ALTA Institute, University of Cambridge, U.K.
`firstname.secondname@cl.cam.ac.uk`

Abstract

Language models provide a key framework for studying linguistic theories based on prediction, but phonological analysis using large language models (LLMs) is difficult; there are few phonological benchmarks beyond English and the standard input representation used in LLMs (subwords of graphemes) is not suitable for analyzing the representation of phonemes. In this work, we demonstrate how **word segmentation** can be used as a phonological probing task, allowing us to study the representations learned by phoneme-based language models trained on child-directed speech across 31 languages. Following computational models of word segmentation, we present unsupervised methods for extracting word boundaries from a trained model using the observation that prediction-error peaks at the start of words. We also use linear probes to identify that these models implicitly track word boundaries, even when they do not appear in training. This cross-lingual work corroborates statistical learning theories of acquisition and empirically motivates new methods for training subword tokenizers.

phonemetransformers/first-words
(CC BY 4.0)
codebyzeb/phonemetransformers
(CC BY 4.0)

1 Introduction

Small models trained on developmentally plausible data have led to numerous advancements across pre-training strategies, architectures and tools for linguistic analysis (Hu et al., 2024). Yet most of this work involves training on English orthographic data with subword tokenization, restricting the ability to study phonological representations and word learning. A few recent studies have demonstrated that these so-called "BabyLMs" can be trained on individual phonemes (Goriely et al., 2024; Bunzeck et al., 2024), supporting phoneme-based phonological analysis. However, the majority of this work continues to center on English, in part due to the lack of phonological benchmarks for other languages.

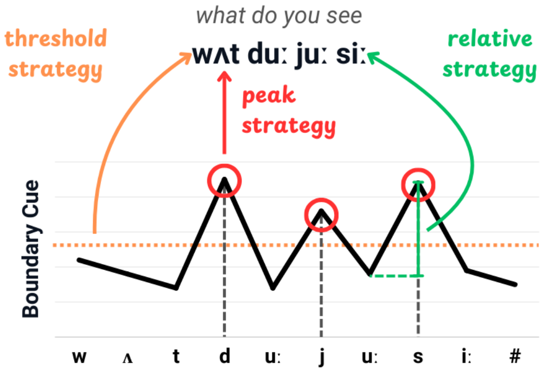


Figure 1: Three strategies for unsupervised word segmentation using cues extracted from an auto-regressive language model trained to predict phonemes.

In this work, we explore the phonological capabilities of phoneme-based BabyLMs across 31 languages using the **word segmentation task**. Following computational models of word segmentation studies in the acquisition literature, we investigate models by assessing their ability to correctly place word boundaries in a sequence of phonemes when word boundaries are not provided during training. Successful segmentation indicates implicit phonological knowledge and when performed zero-shot on developmentally plausible data, contributes to statistical learning theories of language acquisition.

In some of the earliest sequential models, it was noted that *prediction-error* (the degree to which the model struggles to predict the next token) often corresponded with word boundaries (Elman, 1990). Using this observation, we identify four word boundary cues that can be extracted from trained models and three unsupervised strategies for placing boundaries using these cues, as illustrated in fig. 1. We additionally follow the supervised approach of Hahn and Baroni (2019), training linear probes on final layer embeddings to deter-

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 522–539
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

mine if word boundaries are implicitly tracked in order to improve phoneme prediction.

We train phoneme-based BabyLMs on the phonemic transcriptions of child-centered speech comprising the IPA CHILDES dataset (Goriely and Buttery, 2025). We find that these models implicitly encode word boundaries across all 31 languages and identify two factors that may provide useful priors depending on the language: the length of words and the distribution of phonemes at the end of words.

We discuss the validity of orthographic word boundaries as gold labels and note the similarities between our results and recent work that uses byte-level prediction entropy to improve the tokenization step in large language model (LLM) pretraining (Pagnoni et al., 2024). We conclude that this framework not only supports the study of distributional phonology and acquisition, but could also have implications for improving the efficiency and robustness of LLMs.

Finally, we release our code and pre-trained models to facilitate future work.

2 Related Work and Motivations

Since their inception, language models have been used to study the structures of language and explore mechanisms that humans may use to learn them.

Early "connectionist" language models were trained on sequences of letters or phonemes, often using developmentally plausible data in order to explore theories of word learning and phonology (Seidenberg and McClelland, 1989; Norris, 1994; Coltheart et al., 2001). Modern *large* language models (LLMs) are still probed for grammatical information, but standard benchmarks are generally based on higher-order structures: syntax and semantics rather than morphology and phonology. This is due to LLM design being optimized for downstream tasks, not linguistic analysis. For instance, LLMs are typically trained on graphemic text using subword tokens. While this representation is practical for large-scale training, these tokens are not very cognitively plausible (Beinborn and Pinter, 2023), are less effective than character-based tokens for learning word structure (Bunzeck and Zarrieß, 2025) and cannot be used to explore representations of phonological units. Additionally, modern LLMs are inappropriate for theories of acquisition, due to the scales of data they are trained on (Warstadt et al., 2023).

Here, we are interested in evaluating models that train directly on individual phonemes, without word boundaries. When trained on individual words, phoneme LMs have been used to study the acquisition of morphological rules (Kirov and Cotterell, 2018) and compare phonotactic complexity across languages (Pimentel et al., 2020). When trained on running text, phoneme LMs have been used for text-to-speech (Li et al., 2023) and lyric generation (Ding et al., 2024). When compared to grapheme-based models on standard linguistic benchmarks, phoneme models slightly underperform (Nguyen et al., 2022; Bunzeck et al., 2024) but this could be attributed to pre-processing, punctuation and the fact that LLM architectures and evaluation sets have been optimized for written text (Goriely et al., 2024). Despite the benefits of phoneme-based training, phonological evaluation is limited, and few phoneme LMs exist beyond English. Goriely and Buttery (2025) trained phoneme LMs on child-directed speech across 11 languages, but were only able to use an English benchmark for studying how phonological and syntactic knowledge scales in phoneme LMs.

In this work, we propose the word segmentation task as a language-independent method for probing the representations learned by phoneme LMs. Below, we summarize past approaches for investigating the phonological capabilities of language models. We then give historical background on the word segmentation task. Finally, we discuss past examples of word segmentation being used as a probing task.

2.1 Phonological Evaluation of LLMs

While many studies have explored the representations learned by phoneme LMs trained on individual words, there are very few benchmarks for phoneme LMs trained on running text.

One method for testing phonology is to use minimal pairs of words and pseudowords as a lexical decision task. One benchmark that uses this approach is BabySLM (Lavechin et al., 2023), which provides a lexical decision metric for phoneme LMs or speech LMs (which learn directly from audio) using a vocabulary based on child-directed speech. Bunzeck et al. (2025) use a similar approach in order to compare grapheme LMs to phoneme LMs. They also use two probing tasks to examine the representations of sentences; age prediction and rhyme prediction.

PhonologyBench (Suvarna et al., 2024) is a benchmark that uses prompts to test chat-based English LLMs. However, by using prompts, they treat phonology as an emergent ability tested through metalinguistic judgment, an evaluation strategy which Hu and Levy (2023) argues is inferior to using quantities directly derived from a model's representations.

These benchmarks also only test English models, in part due to the lack of phoneme LMs in other languages, but also due to a lack of resources for constructing phonological tasks. For example, pseudowords are typically generated using `wuggy` (Keuleers and Brysbaert, 2010), which only supports three languages for phonetic pseudoword generation. An example of language-independent evaluation of phoneme LMs is the phonetic feature probe used in Goriely and Buttery (2025), which only requires feature vectors for each IPA symbol. The word segmentation task requires no language-specific data, only utterances labeled with word boundaries.

2.2 Computational Models of Segmentation

Unlike in written text, where lexical units are separated by spaces and punctuation, spoken communication consists of continuous utterances with no clear demarcation of words (see, e.g. Cole and Jakimik, 1980). Somehow, without a lexicon to consult, children are able to segment speech into words and phrasal units by the age of six months (Jusczyk, 1999). How children learn to segment words and bootstrap their lexicon is known in psycholinguistics as the *word segmentation problem*, and statistical learning experiments have established a wide variety of statistical cues which children may use to segment speech (Cutler and Carter, 1987; Gleitman et al., 1988; Jusczyk et al., 1993; Saffran et al., 1996b; Jusczyk et al., 1999a; Suomi et al., 1997).

Particularly influential were the experiments of Saffran et al. (1996a), who established that 8-month-old children use distributional information to segment speech, specifically noting that low conditional probability between two adjacent syllables often indicated a word boundary. These experiments inspired the development of computational models proposing cognitively plausible learning mechanisms for word segmentation, most of which are based on the principle that units within words are far more predictable than units across word boundaries (Harris, 1955). Many models draw on Brent (1999), who use unigram statistics to segment speech, with later models using higher-order n-grams (Venkataraman, 2001), incorporating phonological constraints (Blanchard et al., 2010) or leveraging prior distributions over word frequencies and phonological shapes (Goldwater et al., 2009). Other models explicitly calculate several statistical cues at each potential word boundary and combine cues using a majority voting framework (Çöltekin and Nerbonne, 2014; Çöltekin, 2017; Goriely et al., 2023). Each cue provides a signal over the utterance (as illustrated in fig. 1) with peaks in each cue indicating a potential boundary.

Peaks in predictability can also be observed in neural language models. In the foundational work of Elman (1990), a simple recurrent network (SRN) is trained to predict letters in an unsegmented sequence (one of the first examples of auto-regressive language modeling). Elman observes that the prediction-error increases at the onset of each new word, concluding that "there is information in the signal that could serve as a cue to the boundaries of linguistic units which must be learned".

Christiansen et al. (1998) later used an SRN to segment speech by using the probability of an *utterance* boundary, rather than prediction-error, to place word boundaries. This followed previous work suggesting that children could use utterance boundaries to bootstrap their lexicon (Aslin et al., 1996) and is a cue used in the models of Çöltekin and Nerbonne (2014); Goriely et al. (2023).

In this study, we combine ideas from past computational models for word segmentation. Rather than explicitly calculate n-gram statistics, our cues are based on prediction-error and utterance boundary probability extracted from LLMs trained on the next-phoneme prediction task. As these cues are based on the language model's prediction of phonemes, successful segmentation indicates that implicit phonological knowledge of word-like units in these models.

While our experimental setup draws on previous computational work in word segmentation, we do not claim that our phoneme-level language models simulate child language acquisition (see section 6). Rather, we use the segmentation task — with phoneme-level input — as a diagnostic tool that allows us to characterize the cross-linguistic distributional structure of speech sounds and test whether language models naturally cluster sequences into units that coincide with our notion of word-hood. Although our findings may support aspects of statistical learning theories, we acknowledge the limitations of using phoneme-based representations in appendix A.

2.3 Probing for Word Boundaries

Previous work has explored the representations of word boundaries in LLMs. Sanabria et al. (2021) explored methods for extracting word boundaries from attention weights in an LSTM, finding that attention had limited value for segmentation. Hahn and Baroni (2019) trained character-level RNNs and LSTMs without word boundaries, finding that individual activations correlated with word boundaries and that a linear probe trained on all activations also identified boundaries. They claimed that removing word boundaries resulted in a 'near tabula rasa' training paradigm but trained on billions of graphemic words Wikipedia, which is not developmentally plausible. Here, we use this probe on the final layer of phoneme LMs trained on developmentally plausible data, a more 'tabula rasa' paradigm.

Other studies have verified Elman's observations that prediction-error corresponds with word boundaries. For instance, Al-Rfou et al. (2019) train a 64-layer character-level transformer and in qualitative analysis note that three measures of prediction-error sharply increase at the start of words. However, their model is trained on graphemic text from Wikipedia without removing the word boundaries and they do not explicitly use these measures to evaluate word segmentation performance. Here, we use their three measures to propose an unsupervised word segmentation algorithm using phoneme LMs trained without word boundaries.

3 Word Segmentation Task

We use the *word segmentation task* as a zero-shot method for studying the phonological properties of language models trained on phoneme sequences. Given a list of utterances, each of which consists of a non-delimited phoneme sequence, the task is to produce a *segmentation* of each utterance by using an unsupervised method for placing word boundaries. For instance, given the utterance "what do you see", represented phonemically as wʌtduːyuːsiː, successful segmentation would return wʌt duː yuː siː, as demonstrated in fig. 1. Note that phonemes are individual tokens (e.g. uː is a single token, not two) and, crucially, word boundaries are removed during training, although utterance boundaries are present.

Our method for unsupervised word segmentation is based on the observation made by Elman (1990), that cues for word boundaries can be extracted from a sequence prediction model. Given a language model that at each position i provides the probability of a phoneme x given a context $x_1 \dots x_{i-1}$, we extract the following four cues at each potential boundary position:

- **Entropy:** The entropy (in bits) across the probabilities for all items in the vocabulary.
- **Loss:** The cross-entropy loss (bits) calculated as the negative log probability of the subsequent phoneme p_i.
- **Rank:** The rank of x_i in the list of possible tokens at position i sorted by likelihood.
- **Utterance Boundary Probability (UBP):** The probability assigned to the utterance boundary token.

The first three cues are put forward by Al-Rfou et al. (2019), where they are used to qualitatively examine the error rate of their character-based language model. Our use of these cues for word segmentation is novel. The fourth cue, UBP, relates to the model of Christiansen et al. (1998), who found that the prediction of the utterance boundary marker in a SRN increased at word boundaries. All four cues are utilized in the segmentation models of Çöltekin and Nerbonne (2014); Goriely et al. (2023) but rather than being explicitly calculated using n-gram frequencies, we calculate them using the probability distribution produced by a language model.

For each of these cues, we have three methods for placing word boundaries. The first is to identify peaks in each cue: placing word boundaries whenever the cue's value is higher at position i than at position $i-1$ or $i+1$ in the sequence. The second is to learn a single threshold value, placing word boundaries when the cue exceeds it. The third combines both strategies, placing word boundaries when the relative increase of the cue's value from position $i-1$ to i exceeds a learned threshold. We call these the **peak**, **threshold** and **relative** strategies, respectively, as illustrated in fig. 1. We acknowledge that the threshold and relative strategies

Suite Size	Model Parameters	Tokens (words)	Languages
Tiny	400k	100k (~20k)	31
Small	600k	700k (~180k)	17
Medium	5M	1.8M (~500k)	11
Large	19M	18M (~5M)	1

Table 1: The model size in number of (non-embedding) parameters and data size used for each suite of models. Languages are sub-sampled according to the token count for consistency, as word length varies across languages.

are not fully unsupervised, using a single learned parameter.

Finally, in order to explore whether word boundary information is present in the model's representations, we follow Hahn and Baroni (2019) and train a linear probe to predict word boundaries from the final layer embeddings. We implement their 'balanced' probe, training on embeddings taken from an equal number of word-final and word-internal positions, and ensure that no words in the training set are contained in the test set.

4 Experimental Setup

We train a suite of GPT-2 models on each of the 31 languages in the IPA CHILDES corpus. As the size of each subset varies considerably,[1] for a fair comparison we must subsample our training data to the size of the smallest subset and use a very small model to prevent over-fitting. In order to explore the use of larger models and more training data, we train four suites of models, each using a different sample size and model size, setting model parameters according to the scaling experiments of Goriely and Buttery (2025). These suites are detailed in table 1 with parameter configurations and training parameters given in appendix B. The smallest model (only 2 layers) is trained on 100k tokens from all 31 languages, and the largest model (6 layers) is trained on 18M tokens of English.

For the linear probes, we follow Hahn and Baroni (2019) and report accuracy. They claim that chance performance is 50% due to the balanced training data, but our results suggest otherwise. In order to evaluate our unsupervised strategies, we follow past work (see section 2.2) compute the F1 score of boundary placement, excluding boundaries placed at the start and end of utterances (as these are 'free' from the utterance boundaries).

5 Results

We present the results of the word boundary probe in fig. 2 and the maximum boundary F1 scores of our unsupervised segmentation strategies in fig. 3. The individual scores for each combination of language, suite size, boundary cue and segmentation strategy are provided in appendix C.

Overall, both the word boundary probe and the unsupervised strategies successfully identify word boundaries — all probes achieve accuracies significantly higher than the untrained baseline, as do the unsupervised strategies (see appendix D for details on significance tests). The probe accuracies show that models implicitly track word boundaries in their contextual embeddings, suggesting that they are learning phonological rules to aid in next-phoneme prediction. The unsupervised segmentation results indicate that word boundaries can be extracted through prediction across many languages, corroborating previous statistical learning results about the role of distributional cues in language acquisition.

Below, we analyze these results in more detail.

180k words are sufficient for learning word boundaries. We note that across all languages, the accuracy of the word boundary probes increases from the Tiny suite to the Small suite (where models are trained on about 180k words, as seen in table 1), but improvements are minimal for models in the larger suites. This also occurs with the unsupervised approach, despite receiving several orders of magnitude more training data and training with many more parameters. We conclude that 180k words is sufficient for a model to learn word-like units in our framework, but other models may require more or less data.

Utterance boundaries are better predictors of word boundaries than prediction-error. Figure 3 provides the maximum boundary F1 score achieved for each model in each suite across the four boundary cues and three segmentation strategies, for a total of 12 combinations. In table 2 we summarize the cue and strategy combinations that achieved these scores. The UBP cue is the most effective in each suite, out-performing the three cues based on prediction-error, and the relative strategy out-performs the other two strategies. For reference, we give the best combinations for each language in appendix C. Generally, the best cue stays consistent across suites for a particular

---

[1] The North American English section contains 10M words but Farsi only contains 40k.

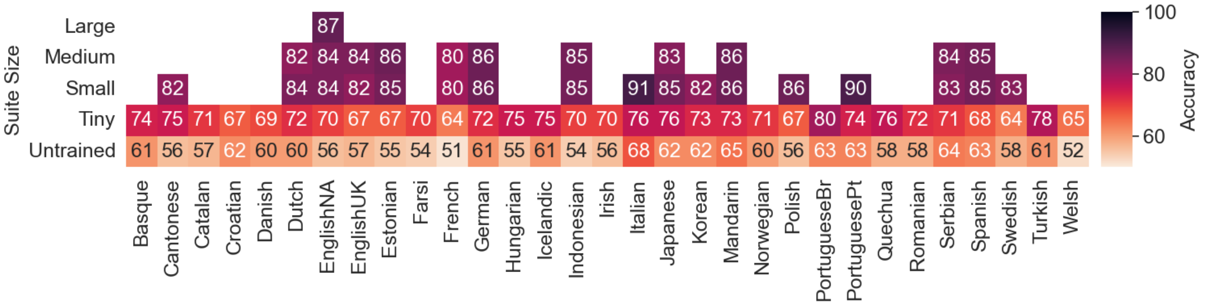


Figure 2: Accuracy scores for the word boundary probe trained on the contextual embeddings of phonemes across models in each suite. Training and test instances are balanced and each word used for training embeddings is removed from the test set. Probe results for each untrained model in the Tiny suite are included as a baseline.

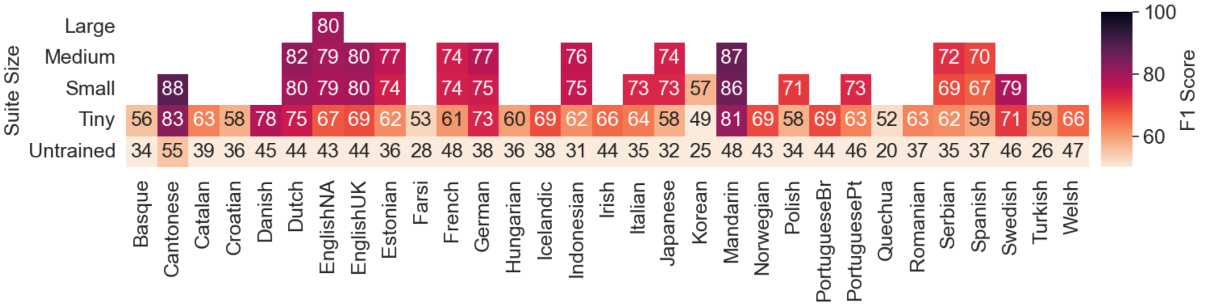


Figure 3: Boundary placement F1 scores achieved using the unsupervised segmentation strategies across models in each suite. For each score, we report the maximum across the 4 cues and 3 segmentation strategies. The Untrained row give the maximum scores achieved by each model in the Tiny suite before training.

Cue & Strategy	Tiny	Small	Medium	Large
UBP (threshold)	3	2	1	-
UBP (relative)	3	6	4	-
UBP (peak)	11	4	3	1
Entropy (threshold)	1	-	1	-
Entropy (relative)	-	4	2	-
Entropy (peak)	-	1	-	-
Loss (relative)	9	-	-	-
Rank (relative)	3	-	-	-
Rank (peak)	1	-	-	-

Table 2: Counts of the word boundary cues and segmentation strategies that achieved the highest F1 scores in each suite.

language (e.g. Entropy is the best cue for Italian), but this is not always the case, and the best strategy also varies.

The peak segmentation strategy fails to capture subsequent boundaries. We compare the four segmentation cues using the peak strategy segment utterances from the EnglishNA section of IPA CHILDES in fig. 4. We identify two failure modes for this strategy. The first is that since two peaks cannot directly follow one another, subsequent boundaries cannot both be successfully placed. In this example, the h in “help” is incorrectly placed by all four cues. A second failure case is that the relative size of peaks is not considered; three cues incorrectly place a boundary within the word “fingers” due to a very small peak at ə. The threshold and relative segmentation strategies address both of these issues but for English the peak strategy is still best overall.

Italian has a strong prior for learning word boundaries. Hahn and Baroni (2019) claim that since the probes are trained on balanced examples, chance accuracy should be 50%. However, we find that the probes trained on completely untrained models (see fig. 2) achieve accuracies ranging from 51% for French up to 68% for Italian. This is because the balancing procedure does not account for the fact that phonemes have different probability distributions depending on their position within words. For example, in fig. 5 we find that at the end of Italian words, a small number of phonemes have particularly high frequencies (the vowels ɐ, o, e and i end 84% of words) whereas the distribution of French word-final phonemes is not as skewed. This skewed distribution provides a useful prior for the Italian probe, which can achieve high accuracies

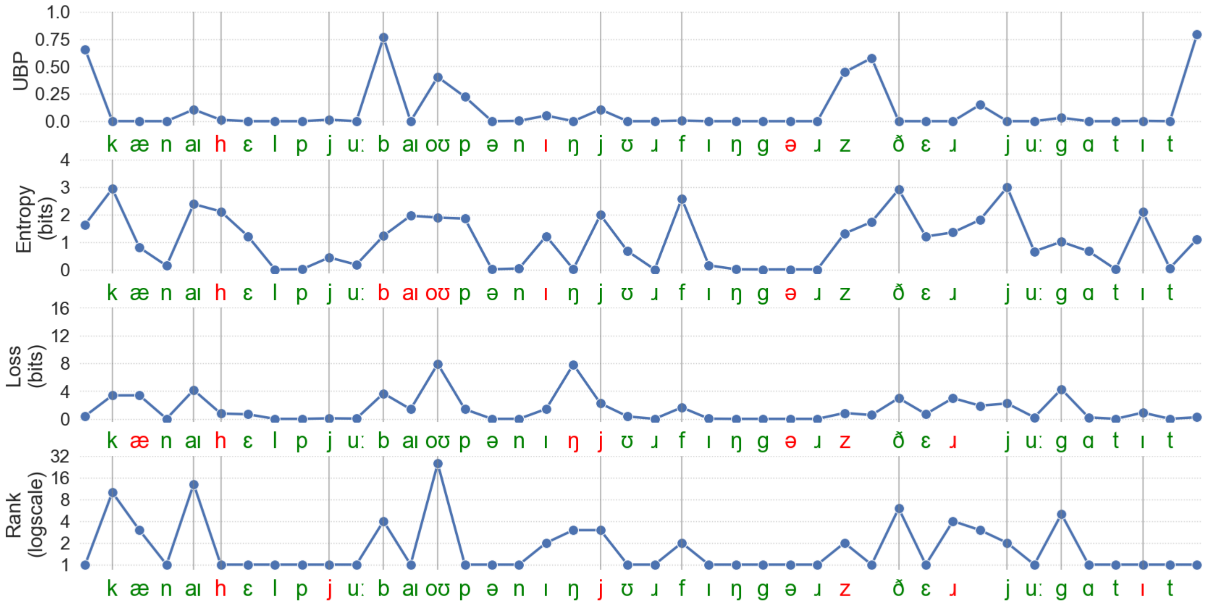


Figure 4: Per-phoneme boundary probability, entropy, loss and rank assigned by the Medium English model for the sequence of utterances “can I help you by opening your fingers”, “there”, “you got it”. Spaces indicate utterance boundaries, vertical lines indicate gold word boundaries and phonemes are marked as green if they are correctly identified as word boundaries using the **peak** strategy or if they follow an utterance boundary (red otherwise).

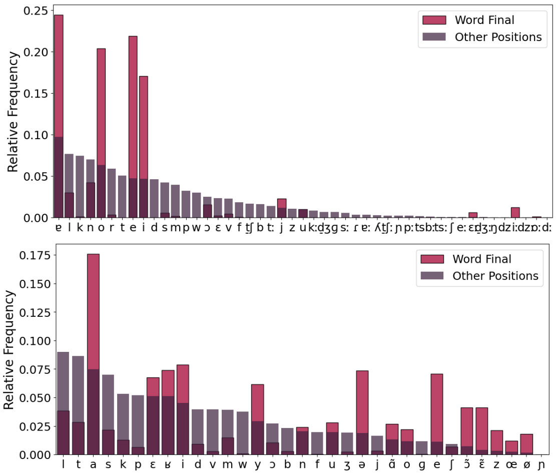


Figure 5: Relative frequencies of phonemes appearing in word-final positions and all other positions for Italian (top) and French (bottom).

by relying on these phoneme frequencies (the only signal available when using embeddings from an unsupervised model). To measure the relative benefit of each prior, we can compute the **normalized entropy** of the word-final phoneme distributions in each language,

$$H_{\text{norm}} = \frac{H(P)}{H_{\max}} = \frac{\sum_{i=1}^{n} p_i \log_i p_i}{log_2(n)},$$

which ranges from 0 (deterministic distribution) to 1 (uniform distribution). We find that not only do Italian and French have the lowest and highest normalized entropies with 0.51 and 0.84, respectively, but in general, this normalized entropy has a high negative correlation with probe accuracy for the untrained models (Pearson $\rho = -0.69$). This correlation is still present for the Tiny suite (Pearson $\rho = -0.52$) but is not significant for the Small and Medium suites, indicating that although the word-final phoneme distribution prior is useful, the embeddings do still encode information about word boundaries that the probes can detect.

Word length is a confounding factor for unsupervised segmentation. Just as with the probes, using our unsupervised methods on untrained models can reveal confounding factors, as shown in fig. 3. The F1 scores for the untrained models range from 20 for Quechua up to 55 for Cantonese. For 25 of the 31 languages, this score comes from the UBP cue with the relative strategy; since the probability of an utterance boundary from an untrained model will randomly vary over the phoneme sequence, boundary placement using the relative strategy essentially places boundaries randomly, which can still yield relatively high F1 scores if words are short. This seems to be the case here; Quechua has the highest average word length in IPA CHILDES and Cantonese has the lowest, with

6.2 and 2.4 phonemes per word, respectively. Generally, we find that word length has a high negative correlation with the F1 scores with Pearson $\rho = -0.94, -0.71, -0.79, -0.42$ for the Untrained, Tiny, Small and Medium suites, respectively (although the final correlation is not significant).

This confounding factor means that we cannot easily compare word segmentation scores between languages, only scores for each language across suite sizes. Compared to the untrained models, the unsupervised word segmentation strategy still achieves significantly higher F1 scores for every language, demonstrating that distributional information is a useful cue for bootstrapping a lexicon.

6 Discussion

In this work, we train BabyLMs on phonemic transcriptions of 31 languages in IPA CHILDES and explore the word segmentation task as a method for probing these models for phonological knowledge. Our results indicate that prediction-error and utterance boundary probability can be used as cues for unsupervised word segmentation. Our study is the first to use prediction-error extracted from LLMs for unsupervised word segmentation, extending previous work that explicitly calculated these cues using n-gram models (Çöltekin and Nerbonne, 2014; Çöltekin, 2017; Goriely et al., 2023). We also update previous neural models of word recognition (Elman, 1990; Christiansen et al., 1998) by using modern architectures and evaluating cross-lingually. We now turn to the broader implications of our findings.

Statistical learning. Viewing our models as statistical learners, we find that no single cue or strategy consistently yields the best segmentation performance across different model sizes and languages. This is perhaps unsurprising, as many of the cues are highly interrelated (for example, entropy and surprisal often correlate) and all segmentation strategies are grounded in the same underlying principle: identifying boundaries at points of high prediction uncertainty. It is this general principle, rather than any specific cue or strategy, that proves sufficient for segmenting utterances into word-like units. Nevertheless, most cues and strategies perform reasonably well on their own. Previous segmentation models have explored combining multiple distributional cues through unsupervised majority voting (Çöltekin, 2017; Goriely et al., 2023), an approach that could be fruitfully applied to the cues investigated here in future work.

Cross-lingual comparison. Comparing models across languages is a challenge. Our study is the first cross-lingual study using the word segmentation task to compare 31 languages, but we identify two confounding factors that inhibit cross-lingual comparison. Firstly, we find that the distribution of phonemes in word-final slots provides a prior not previously accounted for in studies that probed contextual embeddings for word boundary information. Secondly, we find that word length provides a prior for the unsupervised strategies, since randomly placing boundaries yields a higher F1 score when words are shorter, which has not previously been accounted for in cross-lingual word segmentation studies. Nevertheless, both the probes and the unsupervised strategies achieve significant scores for all 31 languages, indicating the importance of the distributional cue for learning to segment speech in any language. These findings also highlight the importance of accounting for frequency information as a prior when training probes or comparing models trained on different datasets.

Simulating acquisition. Our results focus on the performance of our models at the end of training, whereas past work has compared the learning dynamics of phoneme-based models to developmental patterns observed in human acquisition (Kirov and Cotterell, 2018). Although our findings indicate the utility of the distributional cue for identifying word-like units, we do not claim that our models simulate language acquisition. In particular, given recent advances in models that operate directly on raw audio, the use of phoneme-level representations may be insufficient for capturing the full complexity of language learning, as discussed in appendix A.

Rather, we use this framework to investigate the distributional patterns of phonemes across languages and whether language models trained to predict upcoming phonemes implicitly track meaningful sub-sequences that align with words. While many computational models of word segmentation treat segmentation as a necessary precursor for language understanding, this assumption has been questioned. For example, Baayen et al. (2016) show that a tri-phone model, operating on unsegmented utterances can make predictions consistent with infants' sensitivity to linguistic structure. Likewise, recent phoneme-level language models perform well on both linguistic benchmarks and downstream tasks without explicit segmentation (Goriely et al., 2024) — although our results suggest that some degree of implicit segmentation may be occurring to enhance these models' predictive performance.

Word boundaries as gold labels. Throughout this work, we have used word boundaries from orthographic text as the gold labels for evaluation, but these boundaries may not correspond with lexical units in speech. In early stages of acquisition, children may treat both predictable multi-word phrases as single lexical units (MacWhinney, 1978) and unsupervised word segmentation strategies may be segmenting morphemes, rather than words (Fleck, 2008). From an information-theoretic angle, word boundaries may only exist to optimize the trade-off between syntax and morphology across languages (Koplenig et al., 2017; Mosteiro and Blasi, 2025) and in general, what exactly defines a 'word' is still up for debate (Dixon and Aikhenvald, 2003; Haspelmath, 2023).

Unsupervised segmentation for tokenization. Instead of evaluating against word boundaries, we can treat our cues as *graded* measures of co-occurrence statistics, as noted by Elman (1990). This idea can be leveraged to improve the tokenization step in modern LLM pre-training. Instead of forming subwords by merging frequently occurring byte pairs, token sequences that are highly predictable can be combined. Pagnoni et al. (2024) apply this concept to a "token-free" model, where bytes are joined into 'patches' according to the entropy of the probability distribution for each byte (probabilities are computed using a byte-level LLM). They use two constraints for merging bytes which exactly correspond to our threshold and relative segmentation strategies, but only use entropy as a cue. In our experiments, entropy was less effective than utterance boundary probability (UBP) for unsupervised word segmentation and in an initial investigation (see appendix E) we found that creating a subword tokenizer using both cues improves the linguistic abilities of models trained on phonemes compared to regular BPE and that the UBP cue is more effective than entropy. This creates a parallel between word segmentation research and practical applications for tokenization in NLP and we encourage further work in this area.

7 Conclusion

Phoneme-level language models trained on developmentally plausible corpora are a valuable tool for studying cross-lingual phonology and theories of acquisition. In this study, we demonstrate how the **word segmentation task** can be used to probe these models for phonological knowledge and introduce novel unsupervised methods leveraging prediction-error and utterance boundary probability to identify words. Our findings show that models trained on 31 languages can all detect word boundaries; however, cross-linguistic comparisons are influenced by confounding factors such as word length and word-final phoneme distribution. These factors, while positing challenges, also offer new avenues for understanding the role of distributional cues in language processing cross-lingually. Finally, we explore the connection between word segmentation and information-driven tokenization schemes, highlighting how this research can inform and improve practical applications in natural language processing.

Acknowledgments

We are grateful to Pietro Lesci and Julius Cheng for their careful reading of this article and their insightful feedback, which greatly contributed to its improvement.

Our experiments were performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council. Zébulon Goriely is supported by an EPSRC DTP Studentship.

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3159–3166.

Richard N Aslin, Julide Z Woodward, Nicholas P LaMendola, and Thomas G Bever. 1996. Models of word segmentation in fluent maternal speech to infants. In *Signal to syntax*, pages 117–134. Psychology Press.

R. Harald Baayen, Cyrus Shaoul, Jon Willits, and Michael Ramscar and. 2016. Comprehension without segmentation: a proof of concept with naive discriminative learning. *Language, Cognition and Neuroscience*, 31(1):106–128.

Lisa Beinborn and Yuval Pinter. 2023. Analyzing cognitive plausibility of subword tokenization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4478–4486, Singapore. Association for Computational Linguistics.

Daniel Blanchard, Jeffrey Heinz, and Roberta Golinkoff. 2010. Modeling the contribution of phonotactic cues to the problem of word segmentation. *Journal of Child Language*, 37(3):487–511.

Michael R. Brent. 1999. Efficient, probabilistically sound algorithm for segmentation and word discovery. *Machine Learning*, 34(1):71–105.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: Battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2025. Small language models also work with small vocabularies: Probing the linguistic abilities of grapheme- and phoneme-based baby llamas. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6039–6048, Abu Dhabi, UAE. Association for Computational Linguistics.

Bastian Bunzeck and Sina Zarrieß. 2025. Subword models struggle with word learning, but surprisal hides it.

Leshem Choshen, Ryan Cotterell, Michael Y Hu, Tal Linzen, Aaron Mueller, Candace Ross, Alex Warstadt, Ethan Wilcox, Adina Williams, and Chengxu Zhuang. 2024. Call for papers – The BabyLM Challenge: Sample-efficient pretraining on a developmentally plausible corpus. *arXiv preprint arXiv:2404.06214*.

Morten H Christiansen, Joseph Allen, and Mark S Seidenberg. 1998. Learning to segment speech using multiple cues: A connectionist model. *Language and Cognitive Processes*, 13(2-3):221–268.

Ronald A Cole and Jola Jakimik. 1980. A model of speech perception. *Perception and production of fluent speech*, 133(64):133–42.

Çağrı Çöltekin and John Nerbonne. 2014. An explicit statistical model of learning lexical segmentation using multiple cues. In *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*, pages 19–28.

Çağrı Çöltekin. 2017. Using Predictability for Lexical Segmentation. *Cognitive Science*, 41(7):1988–2021.

Max Coltheart, Kathleen Rastle, Conrad Perry, Robyn Langdon, and Johannes Ziegler. 2001. Drc: a dual route cascaded model of visual word recognition and reading aloud. *Psychological review*, 108(1):204.

Anne Cutler and David M. Carter. 1987. The predominance of strong initial syllables in the English vocabulary. *Computer Speech and Language*, 2(3-4):133–142.

Maureen de Seyssel, Marvin Lavechin, and Emmanuel Dupoux. 2023. Realistic and broad-scope learning simulations: first results and challenges. *Journal of Child Language*, 50(6):1294–1317.

Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. 2024. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*.

R. M. W. Dixon and Alexandra Y. Aikhenvald. 2003. *Word: a typological framework*, page 1–41. Cambridge University Press.

Ewan Dunbar, Mathieu Bernard, Nicolas Hamilakis, Tu Anh Nguyen, Maureen de Seyssel, Patricia Rozé, Morgane Rivière, Eugene Kharitonov, and Emmanuel Dupoux. 2021. The zero resource speech challenge 2021: Spoken language modelling. In *Proc. Interspeech 2021*, pages 1574–1578.

Emmanuel Dupoux. 2018. Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner. *Cognition*, 173:43–59.

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

Naomi H. Feldman, Sharon Goldwater, Emmanuel Dupoux, and Thomas Schatz. 2021. Do Infants Really Learn Phonetic Categories? *Open Mind*, 5:113–131.

Margaret M Fleck. 2008. Lexicalized phonotactic word segmentation. In *Proceedings of ACL-08: HLT*, pages 130–138.

Lila R Gleitman, Henry Gleitman, Barbara Landau, and Eric Wanner. 1988. Where learning begins: Initial representations for language learning. *Linguistics: The Cambridge Survey: Volume 3, Language: Psychological and Biological Aspects*, pages 150–193.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54.

Zébulon Goriely, Andrew Caines, and Paula Buttery. 2023. Word segmentation from transcriptions of child-directed speech using lexical and sub-lexical cues. *Journal of Child Language*, pages 1–41.

Zébulon Goriely, Richard Diehl Martinez, Andrew Caines, Paula Buttery, and Lisa Beinborn. 2024. From babble to words: Pre-training language models on continuous streams of phonemes. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 37–53, Miami, FL, USA. Association for Computational Linguistics.

Zébulon Goriely and Paula Buttery. 2025. Ipa-childes & g2p+: Feature-rich resources for cross-lingual phonology and phonemic language modeling.

Michael Hahn and Marco Baroni. 2019. Tabula nearly rasa: Probing the linguistic knowledge of character-level neural language models trained on unsegmented text. *Transactions of the Association for Computational Linguistics*, 7:467–484.

Zellig S. Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Martin Haspelmath. 2023. Defining the word. *Word*, 69(3):283–297.

Jennifer Hu and Roger P Levy. 2023. Prompting is not a substitute for probability measurements in large language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the second BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 1–21, Miami, FL, USA. Association for Computational Linguistics.

Peter W. Jusczyk. 1999. How infants begin to extract words from speech.

Peter W. Jusczyk, Anne Cutler, and Nancy J. Redanz. 1993. Infants' Preference for the Predominant Stress Patterns of English Words. *Child Development*, 64(3):675–687.

Peter W. Jusczyk, Elizabeth A. Hohne, and Angela Bauman. 1999a. Infants' sensitivity to allophonic cues for word segmentation. *Perception and Psychophysics*, 61(8):1465–1476.

Peter W Jusczyk, Derek M Houston, and Mary Newsome. 1999b. The beginnings of word segmentation in english-learning infants. *Cognitive psychology*, 39(3-4):159–207.

Emmanuel Keuleers and Marc Brysbaert. 2010. Wuggy: A multilingual pseudoword generator. *Behavior research methods*, 42:627–633.

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.

Alexander Koplenig, Peter Meyer, Sascha Wolfer, and Carolin Müller-Spitzer. 2017. The statistical trade-off between word order and word structure–large-scale evidence for the principle of least effort. *PloS one*, 12(3):e0173614.

Marvin Lavechin, Maureen de Seyssel, Marianne Métais, Florian Metze, Abdelrahman Mohamed, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2024. Modeling early phonetic acquisition from child-centered audio data. *Cognition*, 245:105734.

Marvin Lavechin, Maureen De Seyssel, Hadrien Titeux, Hervé Bredin, Guillaume Wisniewski, Alejandrina Cristia, and Emmanuel Dupoux. 2022. Can statistical learning bootstrap early language acquisition? a modeling investigation.

Marvin Lavechin, Yaya Sy, Hadrien Titeux, María Andrea Cruz Blandón, Okko Räsänen, Hervé Bredin, Emmanuel Dupoux, and Alejandrina Cristia. 2023. BabySLM: language-acquisition-friendly benchmark of self-supervised spoken language models. In *INTERSPEECH 2023*, pages 4588–4592, Dublin, Ireland. ISCA.

Yinghao Aaron Li, Cong Han, Xilin Jiang, and Nima Mesgarani. 2023. Phoneme-level BERT for enhanced prosody of text-to-speech with grapheme predictions. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Brian MacWhinney. 1978. The acquisition of morphophonology. *Monographs of the Society for Research in Child Development*, 43(1/2):1–123.

Bob McMurray. 2022. The myth of categorical perception. *The Journal of the Acoustical Society of America*, 152(6):3819–3842. Publisher: Acoustical Society of America.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Pablo Mosteiro and Damián Blasi. 2025. Word boundaries and the morphology-syntax trade-off. In *Proceedings of the New Horizons in Computational Linguistics for Religious Texts*, pages 86–93.

Tu Anh Nguyen, Maureen De Seyssel, Robin Algayres, Patricia Roze, Ewan Dunbar, and Emmanuel Dupoux. 2022. Are word boundaries useful for unsupervised language learning? *arXiv preprint arXiv:2210.02956*.

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. In *NeuRIPS Workshop on Self-Supervised Learning for Speech and Audio Processing*.

Dennis Norris. 1994. Shortlist: A connectionist model of continuous speech recognition. *Cognition*, 52(3):189–234.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, et al. 2024. Byte latent transformer: Patches scale better than tokens. *arXiv preprint arXiv:2412.09871*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32.

Tiago Pimentel, Brian Roark, and Ryan Cotterell. 2020. Phonotactic complexity and its trade-offs. *Transactions of the Association for Computational Linguistics*, 8:1–18.

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

Jenny R. Saffran, Elissa L. Newport, and Richard N. Aslin. 1996b. Word segmentation: The role of distributional cues. *Journal of Memory and Language*, 35(4):606–621.

Ramon Sanabria, Hao Tang, and Sharon Goldwater. 2021. On the difficulty of segmenting words with attention. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 67–73.

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.

Mark S Seidenberg and James L McClelland. 1989. A distributed, developmental model of word recognition and naming. *Psychological review*, 96(4):523.

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

Kari Suomi, James M. McQueen, and Anne Cutler. 1997. Vowel harmony and speech segmentation in Finnish. *Journal of Memory and Language*, 36(3):422–444.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Anand Venkataraman. 2001. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):350–372.

Alex Warstadt and Samuel R Bowman. 2022. What artificial neural networks can tell us about human language acquisition. In *Algebraic structures in natural language*, pages 17–60. CRC Press.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM challenge: Sample-efficient pretraining on developmentally plausible corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for english. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Limitations

We acknowledge the following limitations of our work.

Limitations of phonemic data: Using phonemic data for the word segmentation task is the typical framework for exploring relevant acquisition theories. However, the phonemic transcriptions in IPA CHILDES do have limitations. Having been generated using grapheme-to-phoneme (G2P) conversion, they may have been subject to conversion error, and the original transcriptions may also contain errors. The G2P process also removes natural variation in speech, such as accents and allophonic variation. The symbolic nature of phonemes may also be an unrealistic starting point for acquisition; it is unclear if infants have access to phonetic categories at this stage of acquisition (Feldman et al., 2021; McMurray, 2022). Researchers who advocate for using language models as cognitive models argue that the training data should be as developmentally plausible as possible (Dupoux, 2018; Warstadt and Bowman, 2022), and that phonemes may be as implausible as text for simulating early acquisition (Lavechin et al., 2023).

From this perspective, a more appropriate framework is to learn segmentation directly from raw audio, as pursued in the Zero Resource Speech Challenge (Nguyen et al., 2020; Dunbar et al., 2021). Audio-based models naturally incorporate prosodic cues, which play a key role in language acquisition (Cutler and Carter, 1987; Jusczyk et al., 1993, 1999b). Unsupervised models have demonstrated the ability to perform statistical learning directly from raw speech (Lavechin et al., 2022; de Seyssel et al., 2023), and have found that the resulting units tend to be shorter than phonemes, consistent with early perceptual categories (Schatz et al., 2021). While such models show promising signs of early phonetic learning and perform well on word-level tasks, they currently require significantly more data to match the performance of text-based models (Lavechin et al., 2023). Moreover, training on curated audiobook datasets gives these models a considerable advantage over learning from noisier, long-form audio that better resembles real-world input—but ongoing work is making such realistic simulations increasingly viable (Lavechin et al., 2024).

Distribution of languages: When training models cross-lingually, we were limited by the scale of each language partition of the IPA CHILDES dataset. The dataset has a very skewed distribution: the EnglishNA section contains 18M words but the Farsi section only contains 43k words. We addressed this skew by training four suites of models in order to provide a cross-lingual comparison while also exploring how segmentation performance increased in scale for the languages with more data available.

Language coverage: To the best of our knowledge, our work is the most cross-lingual exploration word segmentation to date, but is still limited in language coverage: the languages we compare are predominantly European and Asian, with no languages indigenous to the Americas, Australia or Africa. Word segmentation of languages that are more globally distributed should be explored in future work.

B Implementation Details

We conduct our experiments using the `PyTorch` framework (Paszke et al., 2019) and the `Transformers` library (Wolf et al., 2020).

B.1 Hardware Details

We use a server with one NVIDIA A100 80GB PCIe GPU, 32 CPUs, and 32 GB of RAM for all experiments. Below, we report a subset of the output of the *lscpu* command:

```
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical,
                       48 bits virtual
Byte Order:            Little Endian
CPU(s):                32
On-line CPU(s) list:   0-31
Vendor ID:             GenuineIntel
Model name:            Intel(R) Xeon(R)
                       Silver 4210R CPU
                       @ 2.40GHz
CPU family:            6
Model:                 85
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s):             8
Stepping:              7
BogoMIPS:              4800.11
```

B.2 Model Parameters and Training Procedure

We describe the model and training parameters in table 3. The model parameters were chosen according to the scaling experiments of Goriely and Buttery (2025), who trained a suite of GPT-2 models

Parameter	Tiny	Small	Medium	Large
Layers	2	3	6	6
Heads	4	4	8	8
Dropout	0.3	0.3	0.3	0.1
Embedding Size	128	128	256	512
Inner Size	512	512	1024	2048
Max Example Length	128			
Learning Rate	0.001			
Optimizer	AdamW			
Scheduler Type	Linear			
Max Steps	200k			
Warm-up Steps	60k			
Per Device Batch Size	32			

Table 3: Hyperparameter settings for training the GPT-2 architecture in each suite. Vocabulary size varies according to the language, but all other parameters are constant across experiments. Where values are not reported, they may be assumed to be default values.

for different subsets of the English section of IPA CHILDES and used the lexical score in BabySLM (Lavechin et al., 2023) to determine the best parameters. We note that since these parameters were optimised for English, there may be better parameters for the other languages, but differences in perplexity between languages were generally larger than the differences in perplexity between models in the scaling experiments we reference.

Data is prepared into batches by first tokenizing the entire dataset, combining all tokens into one long vector, and then splitting the vector into chunks of 128 tokens. Only the very last example is padded, if required. At each step during training, random chunks are selected and combined into batches.

Checkpoints are taken every 20,000 steps during training. At each checkpoint, the perplexity is evaluated on the held-back evaluation set, and at the end of training the checkpoint with the lowest perplexity is returned as the best model. For the Tiny suite, many of the best models were from the very first checkpoint, since due to the small training dataset and small model, the model had already fit the data by this point.

C Full Word Segmentation Results

All boundary placement F1 scores for the Tiny, Small, Medium and Large suites are given in fig. 6, fig. 7, fig. 8 and fig. 9, respectively. The best combination of cue and segmentation strategy for each language is given in table 4.

D Significance Tests

All word boundary probes for a particular language are trained and tested on the same evaluation set. We compute significance between two probes using McNemar's Test (McNemar, 1947) over the predicted word boundaries for the evaluation set, with a significance threshold of $p < 0.05$. The same procedure is used when comparing the unsupervised methods.

E Using Word Segmentation Cues for Subword Tokenization

We briefly explore the use of our unsupervised word boundary cues to create a subword tokenizer. Typically, the vocabularies for these tokenizers are generated using methods like Byte-Pair Encoding (Sennrich et al., 2016), where the vocabulary initially consists of each individual byte, and pairs of bytes that frequently co-occur in a training dataset are 'merged' into a new token, with this process repeated until a fixed vocabulary size is reached. We use the same principle, but base merges on the word boundary cues from a language model trained on the dataset.

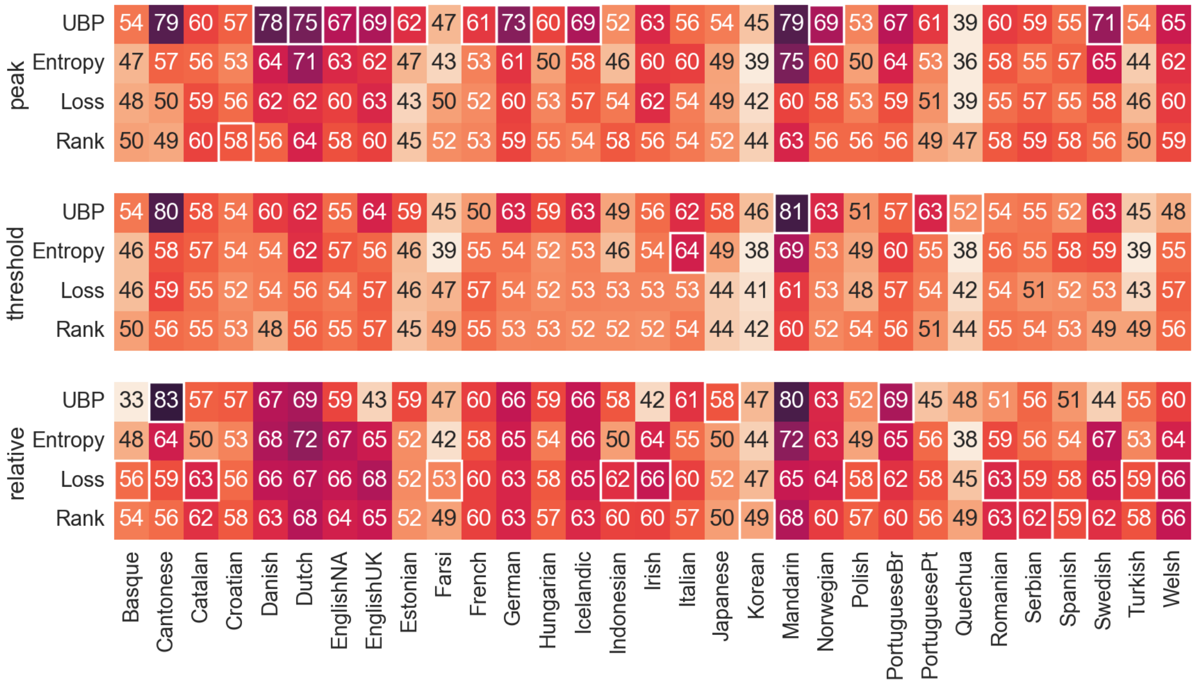
Our method is as follows:

1. We take a trained phoneme-level LM and compute either the UBP cue or the entropy cue at every position in the a given dataset.

2. We initialize our vocabulary V to match the vocabulary of the phoneme LM (so it contains every phoneme plus the utterance boundary token).

3. For every pair of tokens $x_i, x_j \in V$ that co-occur in the dataset, we compute the score for that pair by finding the average value of the word boundary cue at the position of the second token in the pair (e.g. for the pair ð,ɛ, we find the value of the cue at every position where ɛ appears after ð and return the average).

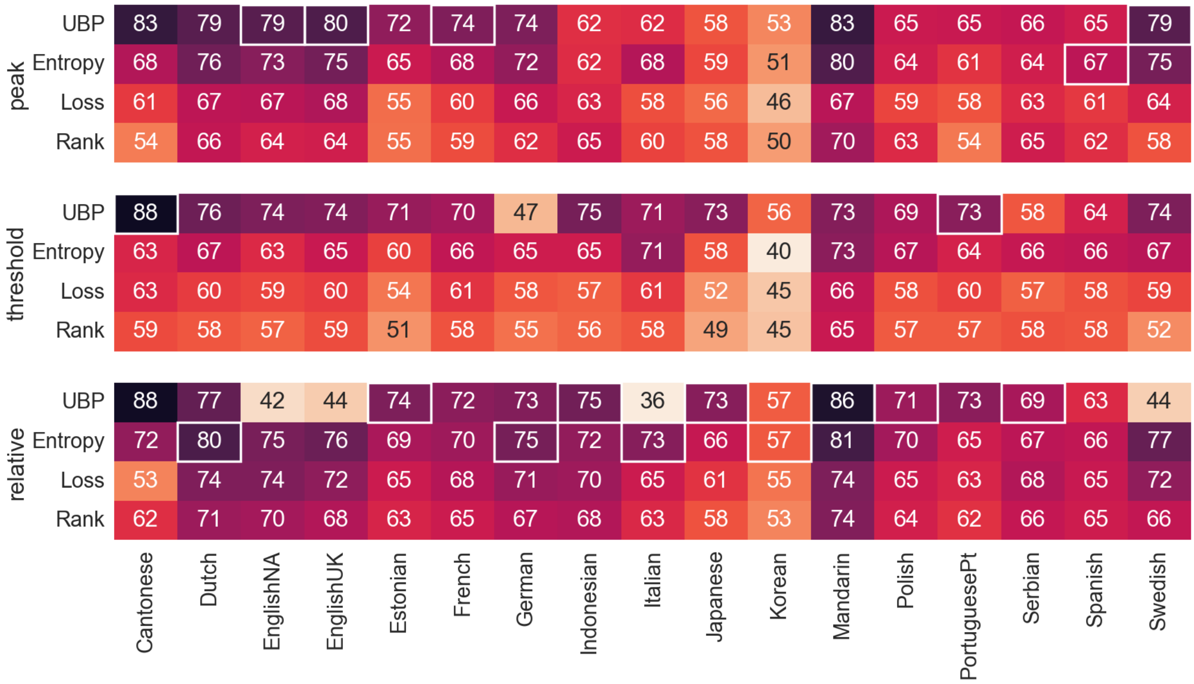
4. We find the pair with the lowest score, create a new token $V_i + V_j$, add it to the vocabulary and apply the merge to every token in the dataset. The cue's value at the newly merged token is set to be the sum of the cue's value of the two tokens before the merging occurs. For the entropy cue this follows from the chain rule and for the UBP cue this results in the



Strategy	Cue	Basque	Cantonese	Catalan	Croatian	Danish	Dutch	EnglishNA	EnglishUK	Estonian	Farsi	French	German	Hungarian	Icelandic	Indonesian	Irish
peak	UBP	54	79	60	57	78	75	67	69	62	47	61	73	60	69	52	63
	Entropy	47	57	56	53	64	71	63	62	47	43	53	61	50	58	46	60
	Loss	48	50	59	56	62	62	60	63	43	50	52	60	53	57	54	62
	Rank	50	49	60	58	56	64	58	60	45	52	53	59	55	54	58	56
threshold	UBP	54	80	58	54	60	62	55	64	59	45	50	63	59	63	49	56
	Entropy	46	58	57	54	54	62	57	56	46	39	55	54	52	53	46	54
	Loss	46	59	55	52	54	56	54	57	46	47	57	54	52	53	53	53
	Rank	50	56	55	53	48	56	55	57	45	49	55	53	53	52	52	52
relative	UBP	33	83	57	57	67	69	59	43	59	47	60	66	59	66	58	42
	Entropy	48	64	50	53	68	72	67	65	52	42	58	65	54	66	50	64
	Loss	56	59	63	56	66	67	66	68	52	53	60	63	58	65	62	66
	Rank	54	56	62	58	63	68	64	65	52	49	60	63	57	63	60	60

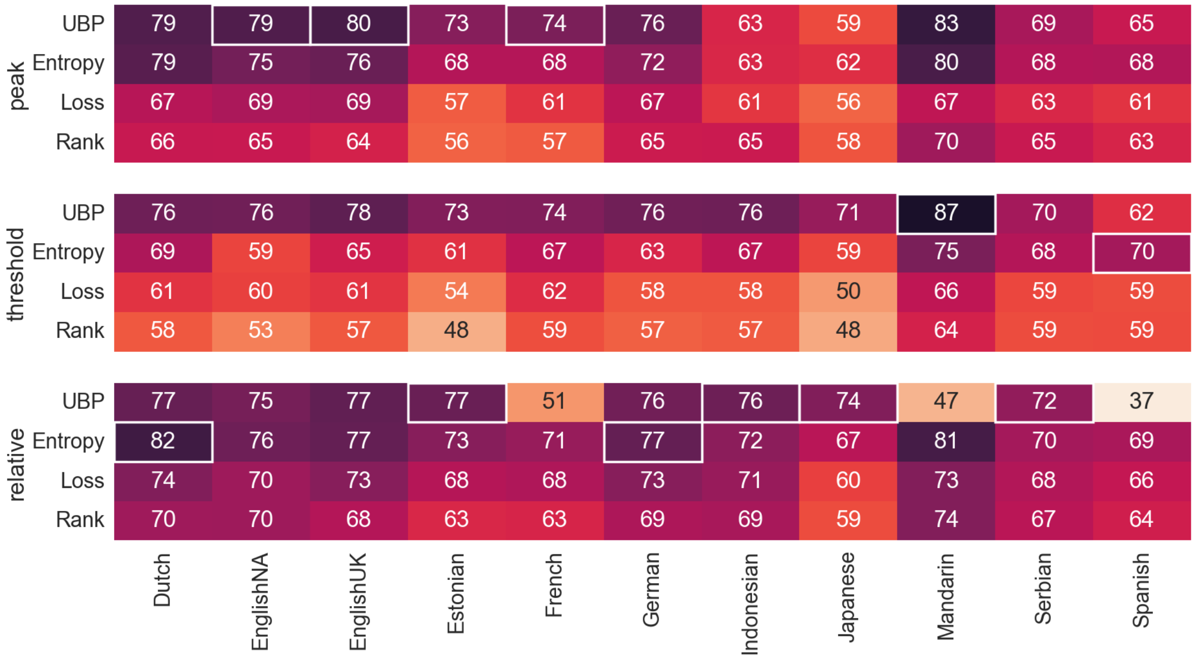
Strategy	Cue	Italian	Japanese	Korean	Mandarin	Norwegian	Polish	PortugueseBr	PortuguesePt	Quechua	Romanian	Serbian	Spanish	Swedish	Turkish	Welsh
peak	UBP	56	54	45	79	69	53	67	61	39	60	59	55	71	54	65
	Entropy	60	49	39	75	60	50	64	53	36	58	55	57	65	44	62
	Loss	54	49	42	60	58	53	59	51	39	55	57	55	58	46	60
	Rank	54	52	44	63	56	56	56	49	47	58	59	58	56	50	59
threshold	UBP	62	58	46	81	63	51	57	63	52	54	55	52	63	45	48
	Entropy	64	49	38	69	53	49	60	55	38	56	55	58	59	39	55
	Loss	53	44	41	61	53	48	57	54	42	54	51	52	53	43	57
	Rank	54	44	42	60	52	54	56	51	44	55	54	53	49	49	56
relative	UBP	61	58	47	80	63	52	69	45	48	51	56	51	44	55	60
	Entropy	55	50	44	72	63	49	65	56	38	59	56	54	67	53	64
	Loss	60	52	47	65	64	58	62	58	45	63	59	58	65	59	66
	Rank	57	50	49	68	60	57	60	56	49	63	62	59	62	58	66

Figure 6: Boundary placement F1 scores achieved by the models in the **Tiny** suite for each cue and segmentation strategy, with the highest score for each language highlighted.



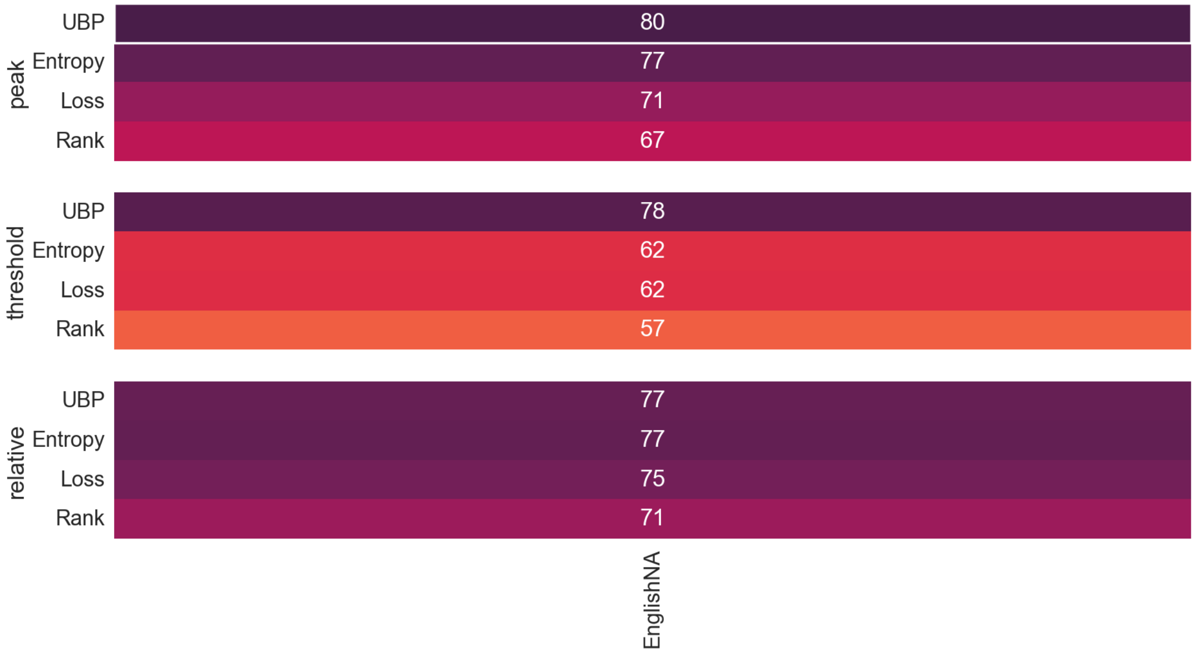
Strategy	Cue	Cantonese	Dutch	EnglishNA	EnglishUK	Estonian	French	German	Indonesian	Italian	Japanese	Korean	Mandarin	Polish	PortuguesePt	Serbian	Spanish	Swedish
peak	UBP	83	79	79	80	72	74	74	62	62	58	53	83	65	65	66	65	79
	Entropy	68	76	73	75	65	68	72	62	68	59	51	80	64	61	64	67	75
	Loss	61	67	67	68	55	60	66	63	58	56	46	67	59	58	63	61	64
	Rank	54	66	64	64	55	59	62	65	60	58	50	70	63	54	65	62	58
threshold	UBP	88	76	74	74	71	70	47	75	71	73	56	73	69	73	58	64	74
	Entropy	63	67	63	65	60	66	65	65	71	58	40	73	67	64	66	66	67
	Loss	63	60	59	60	54	61	58	57	61	52	45	66	58	60	57	58	59
	Rank	59	58	57	59	51	58	55	56	58	49	45	65	57	57	58	58	52
relative	UBP	88	77	42	44	74	72	73	75	36	73	57	86	71	73	69	63	44
	Entropy	72	80	75	76	69	70	75	72	73	66	57	81	70	65	67	66	77
	Loss	53	74	74	72	65	68	71	70	65	61	55	74	65	63	68	65	72
	Rank	62	71	70	68	63	65	67	68	63	58	53	74	64	62	66	65	66

Figure 7: Boundary placement F1 scores achieved by the models in the **Small** suite for each cue and segmentation strategy, with the highest score for each language highlighted.



		Dutch	EnglishNA	EnglishUK	Estonian	French	German	Indonesian	Japanese	Mandarin	Serbian	Spanish
peak	UBP	79	79	80	73	74	76	63	59	83	69	65
	Entropy	79	75	76	68	68	72	63	62	80	68	68
	Loss	67	69	69	57	61	67	61	56	67	63	61
	Rank	66	65	64	56	57	65	65	58	70	65	63
threshold	UBP	76	76	78	73	74	76	76	71	87	70	62
	Entropy	69	59	65	61	67	63	67	59	75	68	70
	Loss	61	60	61	54	62	58	58	50	66	59	59
	Rank	58	53	57	48	59	57	57	48	64	59	59
relative	UBP	77	75	77	77	51	76	76	74	47	72	37
	Entropy	82	76	77	73	71	77	72	67	81	70	69
	Loss	74	70	73	68	68	73	71	60	73	68	66
	Rank	70	70	68	63	63	69	69	59	74	67	64

Figure 8: Boundary placement F1 scores achieved by the models in the **Medium** suite for each cue and segmentation strategy, with the highest score for each language highlighted.



		EnglishNA
peak	UBP	80
	Entropy	77
	Loss	71
	Rank	67
threshold	UBP	78
	Entropy	62
	Loss	62
	Rank	57
relative	UBP	77
	Entropy	77
	Loss	75
	Rank	71

Figure 9: Boundary placement F1 scores achieved by the models in the **Large** suite for each cue and segmentation strategy, with the highest score for each language highlighted.

Language	100k	700k	2M	18M
Basque	Loss (relative)			
Cantonese	UBP (relative)	UBP (threshold)		
Catalan	Loss (relative)			
Croatian	Rank (peak)			
Danish	UBP (peak)			
Dutch	UBP (peak)	Entropy (relative)	Entropy (relative)	
EnglishNA	UBP (peak)	UBP (peak)	UBP (peak)	UBP (peak)
EnglishUK	UBP (peak)	UBP (peak)	UBP (peak)	
Estonian	UBP (peak)	UBP (relative)	UBP (relative)	
Farsi	Loss (relative)			
French	UBP (peak)	UBP (peak)	UBP (peak)	
German	UBP (peak)	Entropy (relative)	Entropy (relative)	
Hungarian	UBP (peak)			
Icelandic	UBP (peak)			
Indonesian	Loss (relative)	UBP (relative)	UBP (relative)	
Irish	Loss (relative)			
Italian	Entropy (threshold)	Entropy (relative)		
Japanese	UBP (relative)	UBP (relative)	UBP (relative)	
Korean	Rank (relative)	Entropy (relative)		
Mandarin	UBP (threshold)	UBP (relative)	UBP (threshold)	
Norwegian	UBP (peak)			
Polish	Loss (relative)	UBP (relative)		
PortugueseBr	UBP (relative)			
PortuguesePt	UBP (threshold)	UBP (threshold)		
Quechua	UBP (threshold)			
Romanian	Loss (relative)			
Serbian	Rank (relative)	UBP (relative)	UBP (relative)	
Spanish	Rank (relative)	Entropy (peak)	Entropy (threshold)	
Swedish	UBP (peak)	UBP (peak)		
Turkish	Loss (relative)			
Welsh	Loss (relative)			

Table 4: Best combination of boundary cue and segmentation strategy for each language and each suite.

probability that *either* original token was an utterance boundary.

5. We repeat (2)-(3), adding new tokens and applying merges until a fixed vocabulary size is reached.

Conceptually, creating merges using minimum average entropy will join highly predictable tokens together and result in tokens with comparable information and a uniformly dense signal that the model can learn from. Creating merges using the minimum average probability of an utterance boundary is similar, but instead tokens are joined according to the model's certainty that they do not cross an utterance boundary.

In order to test this method, we use the phoneme-level LM trained by Goriely et al. (2024) on a phonemized version of the 100-million word BabyLM dataset (Choshen et al., 2024) and train subword tokenizers using a phonemized version of the 10-million word BabyLM dataset. We create two tokenizers with a vocabulary size of 16k using the UBP cue and the entropy cue. We compare these to the BPE tokenizer trained by Goriely et al. (2024) on the same dataset, which also has a vocabulary size of 16k. Note that all three tokenizers are trained on a dataset without word boundaries, so it is possible for tokens to span word boundaries.

Goriely et al. (2024) trained a large model using their BPE tokenizer on the 100-million word BabyLM dataset and evaluated their results on two linguistic benchmarks, BLIMP (Warstadt et al., 2020) and BabySLM (Lavechin et al., 2023). We train and evaluate a model using the same procedure but replace their tokenizer for ours.

The results of this experiment are provided in table 5. We find that our two tokenizers improve all three scores compared to the BPE tbut instead okenizer with the UBP cue leading to a particularly large improvement for the BabySLM syntactic score.

Our method is similar to Pagnoni et al. (2024), who calculate the entropy cue over bytes using a small byte-level LLM, and use either a *global constraint* (corresponding to our threshold segmentation strategy) or a *monotonic constraint* (corresponding to our relative segmentation strategy) in order to group bytes into latent 'patches'. These

Tokenizer	BLIMP	BabySLM Syntactic	BabySLM Lexical
BPE	71.7	74.7	71.2
Entropy	72.7	77.6	81.3
UBP	72.6	85.6	84.4

Table 5: BLIMP and BabySLM scores achieved by a GPT-2 model trained on the BabyLM dataset. We compare BPE to our subword method, where merges are assigned using either entropy or UBP as a cue. BPE results are taken from Goriely et al. (2024).

patches are then fed into the main model, a large transformer, and the encoded patches are 'unpatched' and fed back into the byte-level LLM to predict the next byte. Future work should investigate whether their method is improved by using the cues explored in this study. When training with word boundaries, the prediction of the space character (or other word boundary characters) could also be used to group bytes.

GCG-Based Artificial Languages
for Evaluating Inductive Biases of Neural Language Models

Nadine El-Naggar **Tatsuki Kuribayashi** **Ted Briscoe**
Mohamed bin Zayed University of Artificial Intelligence
`{nadine.naggar, tatsuki.kuribayashi, ted.briscoe}@mbzuai.ac.ae`

Abstract

Recent work has investigated whether extant neural language models (LMs) have an inbuilt inductive bias towards the acquisition of attested typologically-frequent grammatical patterns as opposed to infrequent, unattested, or impossible patterns using artificial languages (White and Cotterell, 2021; Kuribayashi et al., 2024). The use of artificial languages facilitates isolation of specific grammatical properties from other factors such as lexical or real-world knowledge, but also risks oversimplification of the problem.

In this paper, we examine the use of Generalized Categorial Grammars (GCGs) (Wood, 2014) as a general framework to create artificial languages with a wider range of attested word order patterns, including those where the subject intervenes between verb and object (VSO, OSV) and unbounded dependencies in object relative clauses. In our experiments, we exemplify our approach by extending White and Cotterell (2021) and report some significant differences from existing results.

1 Introduction

Attested natural languages (NLs) often have different grammatical properties, such as different word orders, so it is reasonable to ask whether neural language models (LMs) have inductive biases towards specific properties, including different patterns of word order. There are thousands of NLs which differ along multiple semi-independent lexical and grammatical dimensions, so it is difficult to isolate specific properties to evaluate LMs' inductive biases using natural data (Mielke et al., 2019). To remedy this, artificial languages (ALs) have been used in order to create more controlled experiments. Researchers have designed ALs of varying complexities, ranging from lexically-simple but syntactically-complex formal languages, such as the irreducibly context-free Dyck languages or irreducibly indexed (mildly context-sensitive) languages such as cross-serial dependencies ($a^n b^n (c^n)$) (Hewitt et al., 2020), to putatively impossible languages based on permutations of English examples (Kallini et al., 2024).

White and Cotterell (2021) prioritise control of word order in their research. They generate ALs using a Probabilistic Context Free Grammar (PCFG), and use 6 parameters to reorder words and phrases to create 64 ALs with the same lexicon, with the aim of determining whether LMs exhibit an inductive bias towards specific orders. The same dataset of ALs is used by Kuribayashi et al. (2024) to explore a wider range of neural LMs. However, the use of a PCFG precludes the handling of (mildly) context-sensitive NL constructions and does not support a fully general account of unbounded filler-gap dependencies (Steedman, 1996). Furthermore, the use of a VP constituent in the base PCFG means Verb-Subject-Object (VSO) and OSV base orders cannot be represented in the languages created by White and Cotterell (2021).

We create a larger set of ALs that can be used to further test LMs for word order inductive biases covering a wider range of word orders. Specifically, we cover VSO and OSV orders, which represent approximately 8% of attested NLs according to typologists (Dryer and Haspelmath, 2013). Furthermore, we develop an extensible approach to defining ALs that supports the inclusion of mildly context-sensitive (indexed language) constructions, such as cross-serial dependencies, and a general approach to unbounded filler-gap dependencies. We introduce object relative clauses as one exemplar of an unbounded dependency into our extended dataset of ALs. We empirically test LMs on our artificial languages and find significant differences in results compared to existing studies (White and Cotterell, 2021; Kuribayashi et al., 2024), for example, a clearer preference of Transformers for subject-before-verb word orders. This suggests

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 540–556
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

that using more complex, but arguably naturalistic ALs leads to rather different conclusions about the inductive bias of neural LMs

2 Background

2.1 Artificial languages

One line of research has used ALs to evaluate LMs capacity to learn ALs at different levels of the Chomsky hierarchy. Someya et al. (2024) use ALs to determine whether LMs can learn the properties of regular, context-free, and context-sensitive languages, such as nested and long-distance dependencies, and cross-serial dependencies. They find that LSTMs (Hochreiter and Schmidhuber, 1997), Stack-RNNs (Joulin and Mikolov, 2015), and Transformers (Vaswani et al., 2017) struggle to learn nested, long-distance, and cross-serial dependencies, but successfully learn regular languages. Other context-free languages, such as Dyck languages, and mildly context-sensitive languages, like $a^n b^n c^n$, have been used to test recurrent LM learning and generalization to longer sequences (Suzgun et al., 2019; Weiss et al., 2018; El-Naggar et al.) as well as establishing a correspondence between the different LM models and the levels of the Chomsky hierarchy (Delétang et al., 2023). One limitation of this research is that the ALs used diverge from NLs by using minimal vocabulary, many levels of nested dependencies, and so forth.

In another line of research, Chomsky et al. (2023) argued that neural LMs can learn both possible and impossible human languages, so cannot distinguish between them. Kallini et al. (2024) empirically address this claim, by developing putatively impossible AL variants by permutation and modification of an English dataset, following Ravfogel et al. (2019). They find that GPT-2 models struggle to learn the impossible languages, contradicting Chomsky's claim. However, it is difficult to determine precisely what makes the impossible ALs harder to learn because of the multi-dimensional nature of the altered English input.

White and Cotterell (2021) take inspiration from Ravfogel et al. (2019) but use ALs generated by a PCFG to examine the inductive biases of LMs towards different word orders. They use six parameters ('switches') which invert the order of daughter categories within distinct CF productions to determine the structure of their sentences, and evaluate LSTM and Transformer models on the ALs generated by the PCFGs defined by each distinct setting of these parameters. Extending this research, Kuribayashi et al. (2024) evaluate the performance of further cognitively-motivated LMs on the same ALs. However, as a consequence of the use of PCFGs containing a VP constituent, the ALs used by White and Cotterell (2021) and Kuribayashi et al. (2024) do not generate Verb-Subject-Object (VSO) or Object-Subject-Verb (OSV) word orders. In this paper, we generate a wider set of ALs using GCGs and replicate the experiments of Kuribayashi et al. (2024) on this new dataset. Our approach to controlled AL generation is, in principle, expressive enough to generate all attested NL constructions documented by linguists to date, so provides a general framework to support further AL-based investigation of neural LMs. In this paper, we exemplify this by also extending White and Cotterell (2021) dataset to include object relative clauses.

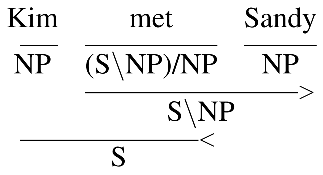
2.2 Categorial Grammar

Classic Categorial Grammar (CG) is a formalism which aims to represent NL syntax isomorphically with compositional semantics (Ajdukiewicz, 1935; Bar-Hillel, 1953). We focus on the syntactic generative properties of extensions to classical CG in this paper. The components of a CG are a lexicon pairing words with basic or functor categories, and a small set of rules defining how functor categories combine with basic categories syntactically and semantically. The "slash" notation is often used to indicate the direction of the arguments relative to the resulting category. For example, X/Y is a functor category looking for an argument basic category Y to the right to create result category X. In classical CG, there are just two rules **forward functional application** (a) or **backward functional application** (b), shown below.

(a) $X/Y \; Y \Rightarrow X$

(b) $Y \; X\backslash Y \Rightarrow X$

In English, a transitive verb like "met" is a functor category $(S\backslash NP)/NP$. The derivation shown below for "Kim met Sandy" shows both forward and backward application.



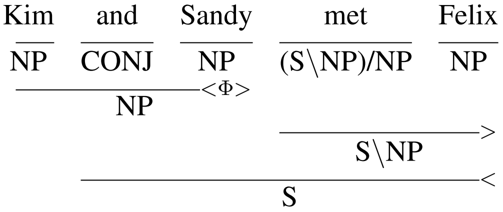
Most if not all of the variation between languages is captured by variation in the set of lexical categories assigned to words.

CG is equivalent to a binary-branching context-free grammar. There are extensions and generalizations of CG, such as Combinatory Categorial Grammar (CCG), (Steedman, 1996), which we refer to generically as Generalized Categorial Grammars (GCGs) (Wood, 2014). In CCG and GCGs, additional operations can be used to combine categories.

One such operation is **coordination**, where 2 constituents of the same category separated by conjunction can be combined into a single constituent of the same type,

$$X \text{ CONJ } X \Rightarrow X$$

Coordination (Φ) is shown in the derivation below.

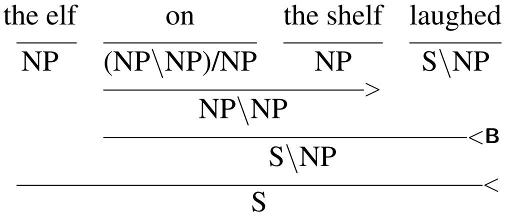


Forward composition and **backward composition** operations are utilized in CCG, where adjacent functions are composed. We show the rules of forward (a) and backward (b) composition below.

(a) $X/Y\ Y/Z \Rightarrow X/Z$

(b) $Y\backslash Z\ X\backslash Y \Rightarrow X\backslash Z$

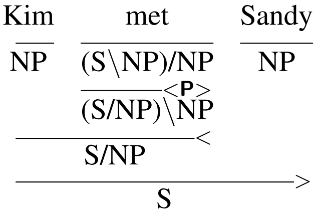
Composition (B) is shown in the derivation below.



Permutation is included in our GCG as a more computationally tractable alternative to type raising in CCG. We use the version from Briscoe (1997, 2000), which allows for a cyclic permutation of the functor arguments without changing their directionality. The definition of permutation is as follows:

$$(X|Y_1)...|Y_n \Rightarrow (X|Y_n)|Y_1$$

Permutation (P) is shown in the derivation below.



We develop our ALs from a GCG utilizing these rules of application, coordination, composition, and permutation.

3 Dataset

As a first case study employing our GCG to create ALs, we mostly reproduce the dataset of White and Cotterell (2021) using GCG but also add some novel word order constructions. Specifically, we adapt the parameters defined by White and Cotterell (2021) to create a GCG for each of the 64 AL configurations they define. We then created lexicons for SOV and VOS languages to create an additional 32 ALs for VSO and OSV languages. We also extend each AL with object relative clauses as an exemplar of a potentially unbounded dependency ('filler-gap') construction.

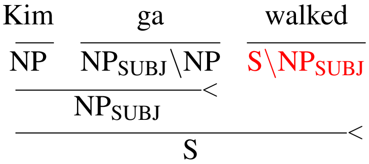
3.1 The Lexicon

We define lexical syntactic categories, e.g., `NP`, first, as listed in Table 1, and then define a set of lexicons. We use a set of mostly English words that is of the same size and has the same categories as White and Cotterell (2021), including singular and plural nouns, and past and present tense verbs, but we ignore subject-verb number agreement, in our initial, simple setting. In addition, following White and Cotterell (2021), we avoid lexical ambiguity, and thus each word in the lexicon is assigned to exactly one category. Following White and Cotterell (2021), we use subject and object markers in all the artificial languages.

3.2 Dataset Generation

Dataset generation involves several steps:

1. **Determining the GCG categories:** We set a GCG lexical syntactic category (e.g., `SCOMP\S`) for each of word types (e.g., `COMP`), as shown in Table 1. These GCG categories are parameterized by seven word order parameters shown in Table 2. For example, if the `S` parameter in Table 2 is set to 0 (head-final), the GCG syntactic type of `VI` (*walked*) should be `S\NP`$_{\mathrm{SUBJ}}$ as follows:



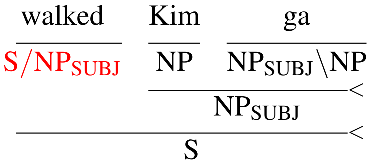
Category	GCG syntactic type	Example
NP (Noun Phrase)	NP	**Kim** ga kissed **Sandy** o
SUBJ (Subject Marker)	$NP_{SUBJ}\backslash NP$	Kim **ga** kissed Sandy o
OBJ (Object Marker)	$NP_{SUBJ}\backslash NP$	Kim ga kissed Sandy **o**
ADJ (Adjective)	$NP \mid NP$	**red** car ga ran
VT (Transitive Verb)	$(S \mid NP_{SUBJ}) \mid NP_{OBJ}$	Kim ga **kissed** Sandy o
VI (Intransitive Verb)	$S \mid NP_{SUBJ}$	red car ga **ran**
VCOMP (Complementary Verb)	$(S \mid NP_{SUBJ}) \mid SCOMP$	Kim ga **believed** that Sandy ga lied
COMP (Verb Complement)	$SCOMP \mid S$	Kim ga believed **that** Sandy ga lied
CONJ (Conjunction)	$var\backslash var/var$	Kim **and** Sandy ga ate
PREP (Preposition)	$(NP \mid NP) \mid NP$	elf **on** shelf ga laughed
REL (Relativizer)	$(NP_{SUBJ} \mid NP_{SUBJ}) \mid (S \mid NP_{OBJ})$	man ga **whom** I ga met laughed

Table 1: Lexical syntactic categories used in our artificial grammar. The bars "|" in the GCG lexical categories indicate either forward- or back-slash, which is controlled by word order parameters in Table 2. The examples in the English grammar are also shown, where the word(s) belonging to the category being described are shown in bold.

Param.	Description	0 (head-final)	1 (head-initial)
`S`	Order of subject and verb	$VI \rightarrow S\backslash NP_{SUBJ}$ $VT \rightarrow (S\backslash NP_{SUBJ}) \mid NP_{OBJ}$ $VCOMP \rightarrow (S\backslash NP_{SUBJ}) \mid SCOMP$	$VI \rightarrow S/NP_{SUBJ}$ $VT \rightarrow (S/NP_{SUBJ}) \mid NP_{OBJ}$ $VCOMP \rightarrow (S/NP_{SUBJ}) \mid SCOMP$
`VP`	Order of object and verb	$VT \rightarrow (S \mid NP_{SUBJ})\backslash NP_{OBJ}$ $VCOMP \rightarrow (S \mid NP_{SUBJ})\backslash SCOMP$ $REL \rightarrow (NP_{SUBJ} \mid NP_{SUBJ}) \mid (S\backslash NP_{OBJ})$	$VT \rightarrow (S \mid NP_{SUBJ})/NP_{OBJ}$ $VCOMP \rightarrow (S \mid NP_{SUBJ})/SCOMP$ $REL \rightarrow (NP_{SUBJ} \mid NP_{SUBJ}) \mid (S/NP_{OBJ})$
`O`	Order of subject and object	Restriction to make an S precede O as canonical word order	Restriction to make an O precede S as canonical word order
`COMP`	Position of complementizer	$COMP \rightarrow SCOMP\backslash S$	$COMP \rightarrow SCOMP/S$
`PP`	Postposition or preposition	$PREP \rightarrow (NP\backslash NP)/NP$	$PREP \rightarrow (NP/NP)\backslash NP$
`ADJ`	Order of adjective and noun	$ADJ \rightarrow NP/NP$	$ADJ \rightarrow NP\backslash NP$
`REL`	Position of relativizer	$REL \rightarrow (NP_{SUBJ}/NP_{SUBJ})\backslash(S \mid NP_{OBJ})$	$REL \rightarrow (NP_{SUBJ}\backslash NP_{SUBJ})/(S \mid NP_{OBJ})$

Table 2: Word order parameters and their associated GCG categories. "A→B" indicates A|B (A is expanded to B) in the GCG derivation.

In contrast, if `S` is set to 1 (head-initial), the possible word order will be like:



Different ALs are generated by different combinations of the seven word-order parameters, which control the directionalities in the lexical categories, resulting in different word orders (Table 2).

2. **Generating the grammars:** We use the seven binary parameters (Table 2) to generate our 96 grammars based on GCG. The parameters, except for `O`, are the same as White and Cotterell (2021), and the `O` parameter biases the S-O order (as a part of postprocessing). This is needed because the permutation operation for the `VT` will eliminate the bias regarding the order of S and O, so to align the experimental settings with White and Cotterell (2021), we add this parameter. The `O` parameter is set to either 0 or 1 only when the subject and object are positioned on the same side of a (transitive) verb (SOV, OSV, VSO, VOS); otherwise, the `O` parameter is automatically determined by the first two parameters of S and VP (SVO and OVS). This process results in 96 grammars – less than the mathematically possible combinations of seven binary parameters (2^7=128). Each language is associated with a specific combination of parameter assignments and denoted, for example, as 0001111 (S=0, VP=0, O=0, COMP=1, PP=1, ADJ=1,

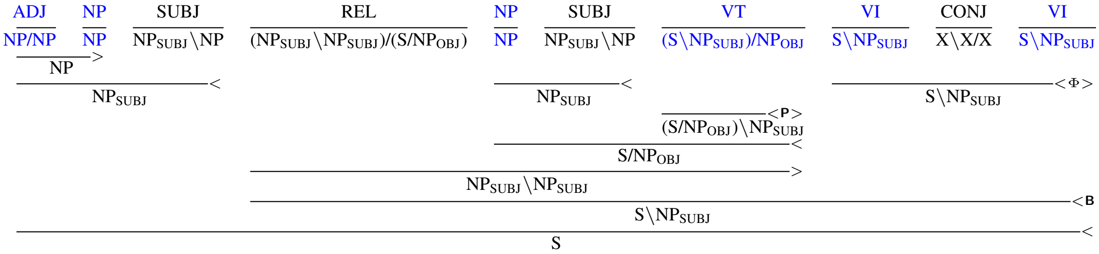


Figure 1: Example of a template and its derivation. The sentence structure is like “Tall man whom she met walked and talked.” The word categories shown in black (e.g., SUBJ) correspond to a single lexical item (e.g., ga). The remaining categories in blue have several candidates of lexical items, and these are uniformly sampled from the predefined dictionary.

Algorithm 1 Template Generation Algorithm

```
Require: Set of word categories $\mathcal{C}$, 96 parsers $[p_1, \cdots, p_{96}]$
  Initialize empty dictionary $ValidTemplates$
  for $length = 3$ to 10 do
      for each sequence of $c \in \mathcal{C}^{length}$ do      ▷ Generate all word category sequences
          if $c$ matches heuristics then
              skip   ▷ Exclude immediately invalid templates
          end if
          for each parser $p_i$ in 96 parsers do
              if $p_i$ successfully parses $c$ then
                  Add $c$ to $ValidTemplates[i]$      ▷ Select grammatically valid templates
              end if
          end for
      end for
  end for
  return $ValidTemplates$
```

Algorithm 2 Generating Sentences from Templates

```
  Input: Valid templates $T$, dictionary $D$ mapping word category $c \in \mathcal{C}$ to lexical items $V_c = D[c]$
  Output: Set of grammatical sentences $S$
  $S \leftarrow \emptyset$
  for each template $t \in T$ do
      for 0 to 500 do
          $s \leftarrow$ dummy string of length $|t|$
          for each category $c_i$ in $t = [c_1, \cdots, c_n]$ do
              Randomly sample $w_i \sim D[c_i]$ (uniform distribution)
              $s[i] = w_i$
          end for
          if $s \notin S$ then
              Add $s$ to $S$
          end if
      end for
  end for
  return $S$
```

REL=1).

3. **Template Generation:** To cover all possible valid syntactic structures in each of our 96 ALs, we first enumerate all possible sequences of word categories (e.g., “`NP ADJ VT CONJ REL...`”), up to length 10, in a brute-force manner. We then parse these sequences with a GCG parser with the corresponding grammar configuration.[1] Word category sequences, and by extension, sentences created from them, are considered grammatically valid if we obtain at least one derivation resulting in S based on the GCG parser. An example of a valid template is shown in Figure 1. This template generation is summarized in Algorithm 1. Note that in order to make this process more efficient, we apply some heuristics (detailed in Appendix A.1) to eliminate templates that cannot result in a valid sentence.

4. **Sentence Generation:** Once we have our templates for each of the 96 grammars, we generate 500 sentences for each template in each grammar by random sampling of the lexicon. We ensure that all of the generated sentences are unique by removing duplicate sentences when they occur. This is shown in Algorithm 2.

5. **Sampling from the Datasets:** Similarly to the dataset size per grammar as White and Cotterell (2021), we randomly sample 50K sentences from the datasets generated for each grammar. We also ensure that all sampled sentences are distinct. These datasets are the ones that we use in our experiments.

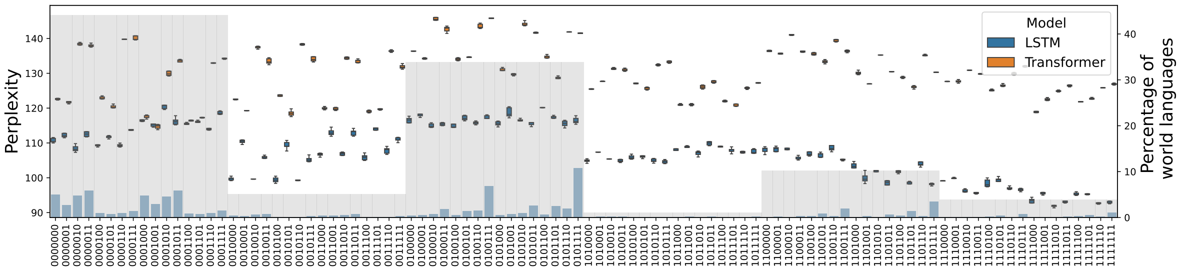
4 Experiments

4.1 Settings

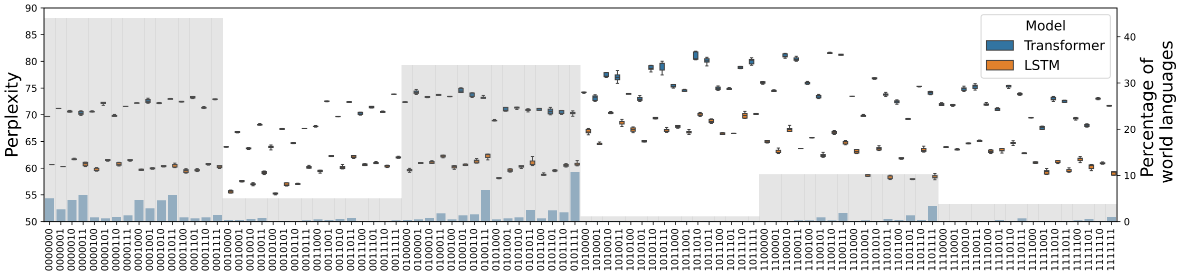
We evaluate the same models as White and Cotterell (2021), which are the LSTM (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani

---

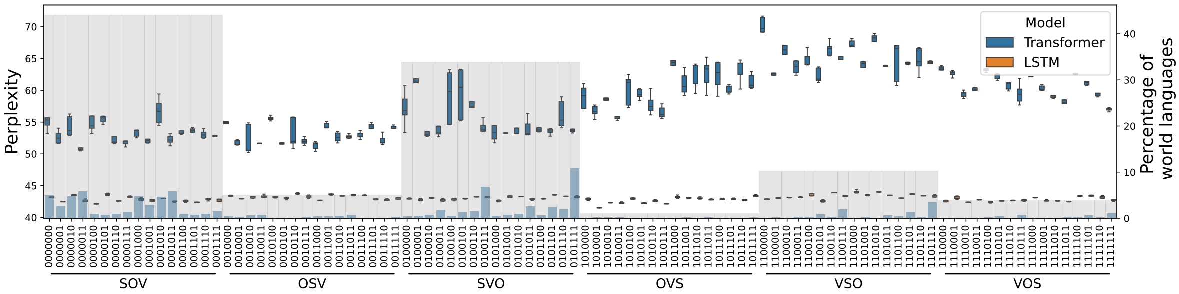
[1] We adapt the NLTK CCGChartParser (Bird et al., 2009), removing type raising and adding the permutation operation as defined by Briscoe (1997, 2000), and use this to parse our templates.



(a) 5 epochs



(b) 10 epochs



(c) Early-stop

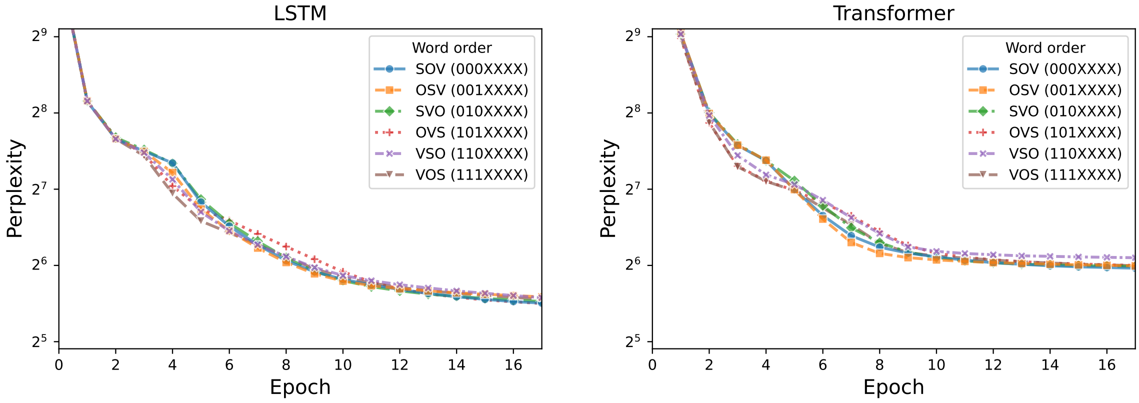
Figure 2: PPLs over 96 grammars. The blue and orange box plots correspond to Transformer and LSTM, respectively. The bars in the graph show the percentage of world languages for each grammar (blue) and word order group, e.g., SOV (gray).

et al., 2017) models. We evaluate perplexity (PPL) over the sentences of the different word orders and investigate the inductive biases that models may have towards specific word order configurations. For each of our 96 languages, similarly to Kuribayashi et al. (2024), the 50K sentences are divided across 5 runs. In each run, the 10K sequences are divided into train/dev/test split with a ratio of 8:1:1. Different random seeds are used in each run. We will basically follow the experimental settings in White and Cotterell (2021) and Kuribayashi et al. (2024) but also extend some analyses focusing on learning dynamics across different training epochs, rather than focusing only on a specific epoch (10 epochs in Kuribayashi et al. (2024)) or the end of learning based on specific criteria (early stopping with patience of 5 in White and Cotterell (2021)).

4.2 Results

What kind of language is harder to learn? Following White and Cotterell (2021); Kuribayashi et al. (2024), we show the PPL distribution across 96 grammars in Figure 2. The distributions at 5 epochs (Figure 2a), 10 epochs (Figure 2b), and the end of training based on early-stopping (consistently longer than 10 epochs; Figure 2c) are reported. Comparing our early-stopping results with those reported in White and Cotterell (2021) with the same stopping criteria, we replicate a high-level trend that Transformers exhibit more PPL variations than LSTMs. At the same time, we observe a somewhat clearer preference of Transformers toward head-final word orders (grammars with many `0`s) than reported in White and Cotterell (2021).

We also observe a dynamic change in word order preference during training. Specifically, at the earlier training phase (5 epochs; Figure 2a), the



(a) LSTM

(b) Transformer

Figure 3: The PPL trajectories for different S-O-V word orders and models (measured on validation data in the early-stopping setting). The y-axis is logarithmic. For better visibility of the preference transition, we cut off large PPLs (y-axis) in the first few epochs and results after the 18th epoch (x-axis), but there is almost no PPL difference across different word order conditions in these epochs.

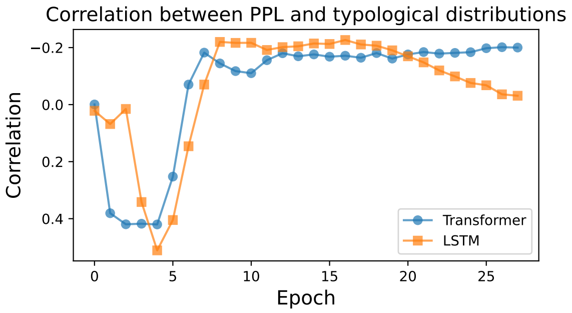
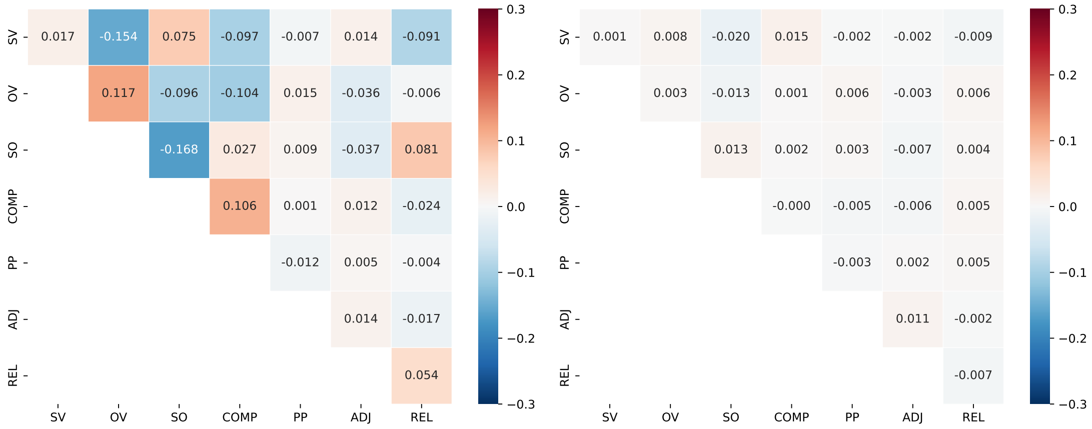


Figure 4: Correlations between PPL and typological distributions, which are measured in each epoch during training (on validation data in the early-stopping setting). The correlations from five runs are averaged. To highlight that a negative correlation is expected, the y-axis is inverted.

PPL tends to be lower in head-initial languages (grammars with many 1s) or more neutral than in the latter phase (early-stopping), which indicates that head-initial languages can be more efficiently learned at first, and then head-final languages outperforms ultimately. Comparing these dynamic preference changes (head-initial→head-final) with the diachronic word order changes in the world's languages, our results, interestingly, contrast with the common view that natural languages have evolved from head-final (SOV) to more neutral (SVO) or head-initial (VSO/VOS) ones (Gell-Mann and Ruhlen, 2011). Figure 3 further summarizes this dynamic change in word order preference.

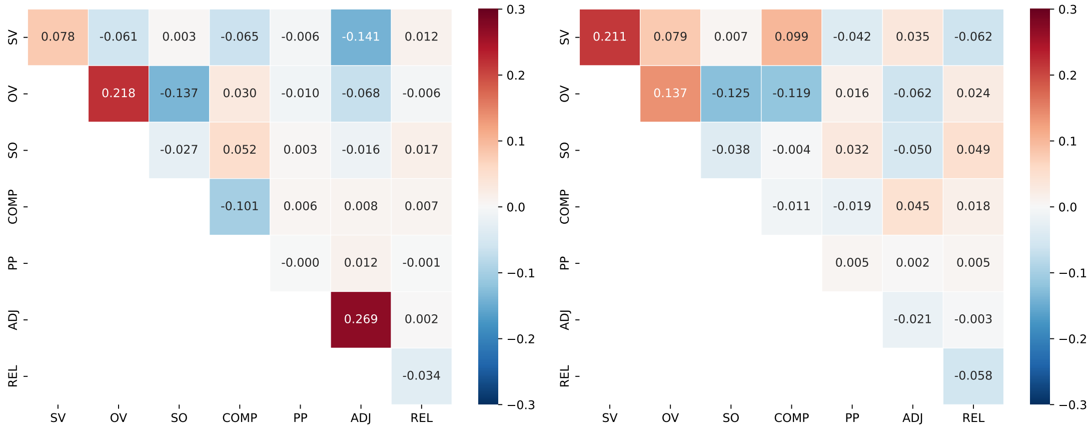
Typological (mis)alignment The percentage of world languages for each grammar and word order group is superimposed on Figure 2 (blue and gray bars). To calculate these typological distributions, we basically adopted the statistics used in Kuribayashi et al. (2024) and enriched them by integrating the S-O order statistics from Dryer and Haspelmath (2013) and complementizer position statistics from Skirgård et al. (2023). The two distributions of PPLs and word order frequencies are compared using Pearson correlation coefficients, following Kuribayashi et al. (2024). After 5 epochs, the correlation between PPLs and typological distributions was 0.40 ($p<0.05$) and 0.25 ($p<0.05$) for LSTM and Transformer, respectively. The positive correlation indicates that the **worse** the PPL is, the **more frequent** the word order is in the world, contrasting with the common claim that natural language is optimized toward better predictability (Gibson et al., 2019; Hahn et al., 2020). After further training in the early-stopping setting, the correlation scores decreased to 0.05 (not significant) and −0.33 ($p<0.05$) for LSTM and Transformer, respectively. These dynamics are shown in Figure 4, where the correlation between typological distributions and PPL distributions for each training epoch is reported. There is a general trend that stable results (i.e., not changing suddenly in adjacent epochs) and better typological correlations are obtained at the later phase of training, but the typological alignment of the LSTM ultimately decreased and lost word order preferences as shown in Figure 2c.

Regression analysis Figure 5 shows quantitative statistics on which word order parameters are asso-



(a) LSTM (5 epochs)

(b) LSTM (early-stop)



(c) Transformer (5 epochs)

(d) Transformer (early-stop)

Figure 5: Coefficients of word order parameters (and their interactions) estimated by the regression models to predict PPL from word order parameters

ciated with the PPL differences. Similarly to White and Cotterell (2021), we train a regression model to predict PPLs from word order parameters and their interaction terms.[2] Positive coefficients for a single word-order parameter (diagonal elements of matrices in Figure 5) indicate that head-initial assignment leads to **worse** PPLs. Positive coefficients for interaction terms indicate that the consistent head-directionality between the two parameters leads to **worse** PPLs, and these are expected to be negative if the common patterns of consistent head-directionalities in natural language are from learners' biases. The coefficients for interaction terms are frequently positive; thus Transformers and LSTMs do not exhibit inductive biases toward typologically plausible, consistent head-directionality, which is consistent with the results in White and Cotterell (2021).

The coefficient matrices also suggest that both training setting differences (e.g., Figures 5a vs. 5b) and model architecture differences (e.g., Figures 5a vs. 5c) had an impact on the results. As for the `REL` parameter, where our inclusion of object relative clauses may impact results, we did not observe previously reported trends, for example, a relatively large positive interaction between `OV` and `REL` reported in White and Cotterell (2021) disappeared.

4.3 Discussion

There are several possible reasons that could explain the differences between our findings and those of White and Cotterell (2021) and Kurib-

---

[2] We used the statsmodels package (Seabold and Perktold, 2010). The formulation is `PPL ∼ SV*OV + SV*SO + SV*COMP + SV*PP + SV*ADJ + SV*REL + OV*SO + OV*COMP + OV*PP + OV*ADJ + OV*REL + SO*COMP + SO*PP + SO*ADJ + SO*REL + COMP*PP + COMP*ADJ + COMP*REL + PP*ADJ + PP*REL + ADJ*REL`, where each parameter is a binary factor with dummy coding (head-final as 0 and head-initial as 1), and `X*Y` represents to both main effects of `X` and `Y` and their interaction effect of `X:Y`. We normalized PPL scores with min-max scaling. In contrast to White and Cotterell (2021), we did not include the sentence-level random effect because our dataset does not have strict alignment between sentences across different grammars.

ayashi et al. (2024). One reason may be that the GCG-generated datasets are potentially more complex than the PCFG-generated datasets used by White and Cotterell (2021) and Kuribayashi et al. (2024). Our datasets include some long-distance dependencies, and in some cases, as a result of permutation, more flexible word orders. Another source of the difference is the addition of 32 grammars (VSO and OSV), which were not included in previous studies. At the same time, we simplified the grammar to omit subject-verb number agreement in this study; thus, the impact of adding such strict agreement rules should be considered in future work.

The dynamic change of word order preference over training epochs emphasizes the effect of inductive biases from training hyperparameters (e.g., training length) beyond model architectures. We have reported experiments only using specific LSTM and Transformer LMs (see Appendix B), but as an orthogonal endeavor to refining ALs, testing a more diverse set of models, including syntactic LMs (Kuribayashi et al., 2024) and more comprehensive exploration of model configurations (e.g., layer numbers, parameter sizes), should yield further insights.

5 Conclusions

In this paper, we extend the work of White and Cotterell (2021) and create a broader set of ALs to evaluate the inductive biases of LMs towards different word orders. This includes the OSV and VSO word orders that were not represented in previous works (White and Cotterell, 2021; Kuribayashi et al., 2024) and permits the inclusion of constructions, which can represent more complex or flexible structures and orders, including longer distance dependencies. We evaluate LSTM and Transformer learning of our ALs and calculate perplexity. We find that the models prefer head initial languages, which contrasts with the findings obtained in previous work. This is intriguing and raises questions that we intend to address and explore further in future work.

We intend to investigate the effects of different training settings and paradigms, on the learning of different language configurations. We also intend to investigate and explore how the models generalize beyond the training data, e.g., to longer sequences. We also intend to investigate and understand model learning and behavior when exposed to different types of long-distance dependencies, such as nested dependencies and cross-serial dependencies, as they occur in NLs. The lexicon we use here disregards verb tenses and number agreement. In future work, we plan to extend our lexicon to contain more detail about the specific features of words and, in general, inject more realistic properties into our ALs.

Limitations

In this work, we use artificial languages to evaluate our LMs' inductive biases. Artificial languages, though controlled, often do not reflect many of the properties and complexities of natural languages, such as subject-verb agreement, lexical ambiguity, and long-distance dependencies. We do not currently distinguish between nouns of different pluralities or verbs of different tenses in our lexicon. More critically, the meaning of sentences in our artificial language is nonsensical in the sense that terminal lexical symbols are randomly sampled, while natural language will have selectional preferences (Hopkins, 2022), or more generally, grounding to events/propositions in the real world. Although our study is a step in the direction of resolving such limitations with GCG, in the future, we plan to extend our lexicon and grammar, including crosslingual perspectives (Xu et al., 2025; Yang et al., 2025), to include more detail and more realistic properties of natural language step-by-step. There is also room to explore the design of typologically impossible/implausible features (Hunter, 2025). Our artificial languages go beyond context-free, and allow us to evaluate the different types of longer-distance dependencies, which we have not explored in detail in this work, but plan to address in the future.

Such future work should also include more in-depth ablations on what kind of additional complexity, compared to the existing PCFG data, affected the results. The evaluation framework also has room to be extended; for example, we can evaluate the compositional generalization of LMs using out-of-domain, longer sequences in evaluation. It will also be fruitful to integrate the perspective of interpretability research to answer how and why LMs struggle with specific word order languages internally.

From an engineering perspective, our dataset generation pipeline can be improved. We first generated possible word sequences in a brute-force manner, and then these were filtered with some heuristic rules and a CCG parser. This brute-force process will limit generation of a corpus with longer sentence lengths, and should be replaced with a more efficient method.

Lastly, while the training paradigms we use in this work are very commonly used, our tested LMs are limited with respect to, e.g., their parameter size, types, and training procedures. In the future, we would like to develop a better understanding of the learning dynamics and explore LM learning of our ALs using different learning paradigms.

Ethical Statement

The data used in this paper is artificial data based mostly on English words. It does not contain any sensitive information or any information that poses any risks. We have no ethical concerns with the contents of this paper.

Acknowledgment

We appreciate constructive feedback from anonymous reviewers.

References

Kazimierz Ajdukiewicz. 1935. Die syntaktische konnexitat. *Studia philosophica*, pages 1–27.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29(1):47–58.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Ted Briscoe. 1997. Co-evolution of language and of the language acquisition device. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain*, pages 418–427.

Ted Briscoe. 2000. Grammatical acquisition: Inductive bias and coevolution of language and the language acquisition device. *Language*, 76(2):245–296.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam chomsky: The false promise of chatgpt. *The New York Times*, 8.

Grégoire Delétang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, and Pedro A. Ortega. 2023. Neural networks and the chomsky hierarchy. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online (v2020.4)*. Zenodo.

Nadine El-Naggar, Pranava Madhyastha, and Tillman Weyde. Exploring the long-term generalization of counting behavior in rnns. In *I Can't Believe It's Not Better Workshop: Understanding Deep Learning Through Empirical Falsification*.

Murray Gell-Mann and Merritt Ruhlen. 2011. The origin and evolution of word order. *Proceedings of the National Academy of Sciences*, 108(42):17290–17295.

Edward Gibson, Richard Futrell, Steven P Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends Cogn. Sci.*, 23(5):389–407.

Michael Hahn, Dan Jurafsky, and Richard Futrell. 2020. Universals of word order reflect optimization of grammars for efficient communication. *Proc. Natl. Acad. Sci. U. S. A.*, 117(5):2347–2353.

John Hewitt, Michael Hahn, Surya Ganguli, Percy Liang, and Christopher D. Manning. 2020. Rnns can generate bounded hierarchical languages with optimal memory. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1978–2010.

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

Mark Hopkins. 2022. Towards more natural artificial languages. In *Proceedings of the 26th Conference on Computational Natural Language Learning (CoNLL)*, pages 85–94, Abu Dhabi, United Arab Emirates (Hybrid).

Tim Hunter. 2025. Kallini et al. (2024) do not compare impossible languages with constituency-based ones. *Computational Linguistics*, pages 1–10.

Armand Joulin and Tomás Mikolov. 2015. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 190–198.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 14691–14714.

Tatsuki Kuribayashi, Ryo Ueda, Ryo Yoshida, Yohei Oseki, Ted Briscoe, and Timothy Baldwin. 2024. Emergent word order universals from cognitively-motivated language models. In *Proceedings of the*

62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 14522–14543.

S. J. Mielke, Ryan Cotterell, Kyle Gorman, Brian Roark, and Jason Eisner. 2019. What kind of language is hard to language-model? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4975–4989.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of rnns with synthetic variations of natural languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3532–3542.

Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference 2010 (SciPy 2010), Austin, Texas, June 28 - July 3, 2010*, pages 92–96.

Hedvig Skirgård, Hannah J. Haynie, Damián E. Blasi, Harald Hammarström, Jeremy Collins, Jay J. Latarche, Jakob Lesage, Tobias Weber, Alena Witzlack-Makarevich, Sam Passmore, Angela Chira, Luke Maurits, Russell Dinnage, Michael Dunn, Ger Reesink, Ruth Singer, Claire Bowern, Patience Epps, Jane Hill, Outi Vesakoski, Martine Robbeets, Noor Karolin Abbas, Daniel Auer, Nancy A. Bakker, Giulia Barbos, Robert D. Borges, Swintha Danielsen, Luise Dorenbusch, Ella Dorn, John Elliott, Giada Falcone, Jana Fischer, Yustinus Ghanggo Ate, Hannah Gibson, Hans-Philipp Göbel, Jemima A. Goodall, Victoria Gruner, Andrew Harvey, Rebekah Hayes, Leonard Heer, Roberto E. Herrera Miranda, Nataliia Hübler, Biu Huntington-Rainey, Jessica K. Ivani, Marilen Johns, Erika Just, Eri Kashima, Carolina Kipf, Janina V. Klingenberg, Nikita König, Aikaterina Koti, Richard G. A. Kowalik, Olga Krasnoukhova, Nora L.M. Lindvall, Mandy Lorenzen, Hannah Lutzenberger, Tônia R.A. Martins, Celia Mata German, Suzanne van der Meer, Jaime Montoya Samamé, Michael Müller, Saliha Muradoğlu, Kelsey Neely, Johanna Nickel, Miina Norvik, Cheryl Akinyi Oluoch, Jesse Peacock, India O.C. Pearey, Naomi Peck, Stephanie Petit, Sören Pieper, Mariana Poblete, Daniel Prestipino, Linda Raabe, Amna Raja, Janis Reimringer, Sydney C. Rey, Julia Rizaew, Eloisa Ruppert, Kim K. Salmon, Jill Sammet, Rhiannon Schembri, Lars Schlabbach, Frederick W.P. Schmidt, Amalia Skilton, Wikaliler Daniel Smith, Hilário de Sousa, Kristin Sverredal, Daniel Valle, Javier Vera, Judith Voß, Tim Witte, Henry Wu, Stephanie Yam, Jingting Ye, Maisie Yong, Tessa Yuditha, Roberto Zariquiey, Robert Forkel, Nicholas Evans, Stephen C. Levinson, Martin Haspelmath, Simon J. Greenhill, Quentin D. Atkinson, and Russell D. Gray. 2023. Grambank reveals the importance of genealogical constraints on linguistic diversity and highlights the impact of language loss. *Science Advances*, 9.

Taiga Someya, Ryo Yoshida, and Yohei Oseki. 2024. Targeted syntactic evaluation on the chomsky hierarchy. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 15595–15605.

Mark Steedman. 1996. Surface structure and interpretation.

Mirac Suzgun, Yonatan Belinkov, Stuart M Shieber, and Sebastian Gehrmann. 2019. Lstm networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*, pages 44–54.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

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, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers*, pages 740–745.

Jennifer C. White and Ryan Cotterell. 2021. Examining the inductive bias of neural language models with artificial languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 454–463.

Mary McGee Wood. 2014. *Categorial grammars (RLE linguistics b: Grammar)*. Routledge.

Tianyang Xu, Tatsuki Kuribayashi, Yohei Oseki, Ryan Cotterell, and Alex Warstadt. 2025. Can language models learn typologically implausible languages? *Preprint*, arXiv:2502.12317.

Xiulin Yang, Tatsuya Aoyama, Yuekun Yao, and Ethan Wilcox. 2025. Anything goes? a crosslinguistic study of (im)possible language learning in lms. *Preprint*, arXiv:2502.18795.

A Dataset Details

A.1 Heuristics Used in Template Generation

In order to make the template generation process more efficient, we apply some heuristics to eliminate templates that would not result in valid sentences in any of our artificial languages. We eliminate templates with the following properties:

1. Shorter than 3 words (the shortest valid sentence in all grammars is 3 words),
2. Starting with a conjunction,
3. Ending with a conjunction,
4. Containing 2 consecutive conjunctions,
5. Containing 2 consecutive prepositions,
6. Starting with subject or object markers,
7. The total number of subject and object markers is greater than the number of NPs,
8. A complementizer appears in the template without a complement verb.

A.2 Restrictions Applied to Parser

In order to parse our templates and assign them to the suitable languages, we adapt the NLTK CCGChartParser (Bird et al., 2009) by disabling type raising, which is included in Combinatory Categorial Grammar (CCG) (Steedman, 1996) and implement and integrate the permutation operation as defined by Briscoe (1997, 2000), which is included in Generalized Categorial Grammar (GCG) (Wood, 2014). We disallow crossed composition and restrict the composition operations in the parser to forward and backward composition.

In the NLTK CCGChartParser, restrictions can be applied to prevent composition, crossing, and substitution by adding ",","." or "_", respectively, before the argument when defining the grammar. When we implement permutation, we introduce an additional character "@" that prevents permutation from being applied.

When defining our grammars, we restrict permutation to categories with S functors only, i.e., verbs. Additionally, in order to restrict the subject and object markers to only combine with NP, we restrict composition when defining the NP_{SUBJ} and NP_{OBJ} categories in the grammar.

Using GCGs to create our artificial languages can allow for flexible word orders as a result of permutation. This would result in OSV sentences being present in SOV datasets, VSO sentences being present in VOS datasets and vice versa. We inhibit permutation when parsing templates into OSV, SOV, VOS and OVS languages, except in the sentences where a REL category is present. This way, there is a clearer distinction between these languages.

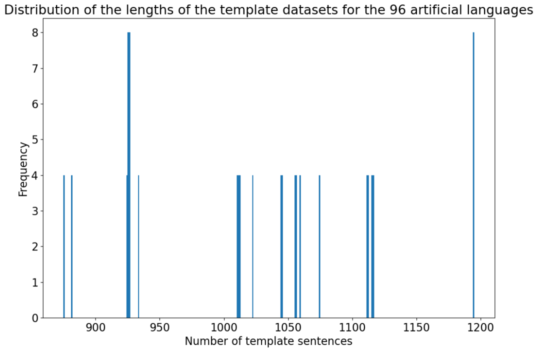


Figure 6: Histogram showing the distribution of the number of templates in the 96 artificial languages

A.3 Dataset Statistics

We calculate statistics for our 96 artificial languages and the templates from which we generate the sentences to provide more insight into the properties of the datasets.

We calculate the average sequence length for the templates and sentences used in evaluation, and they are both approximately 9.42 words long. We count the number of sequences in each template and plot the distribution of them in Figure 6. The smallest and largest template files consist of 875 and 1195 template sequences, respectively. We calculate the average template size as 1022.75 sequences.

We show the number of overlapped sentences and overlapped templates, and the percentage of overlapped sentences and templates in Figures 7,8,9, and 10. As shown in the heatmaps, there is some overlap in the templates for the different languages (Figures 9 and 10). However, there is negligible overlap between the datasets used for experiments (Figures 7 and 8).

B Model Details

Hyperparameters of the Transformer and LSTM LMs are shown in Table 3, which is the same as Kuribayashi et al. (2024). Models are trained with the Fairseq (Ott et al., 2019) toolkit.

Fairseq model	share-decoder-input-output-embed	True
	embed_dim	128
	ffn_embed_dim	512
	layers	2
	heads	2
	dropout	0.3
	attention_dropout	0.1
	#params.	462K
Optimizer	algorithm	AdamW
	learning rates	5e-4
	betas	(0.9, 0.98)
	weight decay	0.01
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	400
	warmup init learning rate	1e-7
Training	batch size	512 tokens
	sample-break-mode	none
	epochs	10

(a) Transformer.

Fairseq model	share-decoder-input-output-embed	True
	embed_dim	128
	hiden_size	512
	layers	2
	dropout	0.1
	#params.	3,547K
Optimizer	algorithm	AdamW
	learning rates	5e-4
	betas	(0.9, 0.98)
	weight decay	0.01
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	400
	warmup init learning rate	1e-7
Training	batch size	512 tokens
	sample-break-mode	none
	epochs	10

(b) LSTM.

Table 3: Model hyperparameters

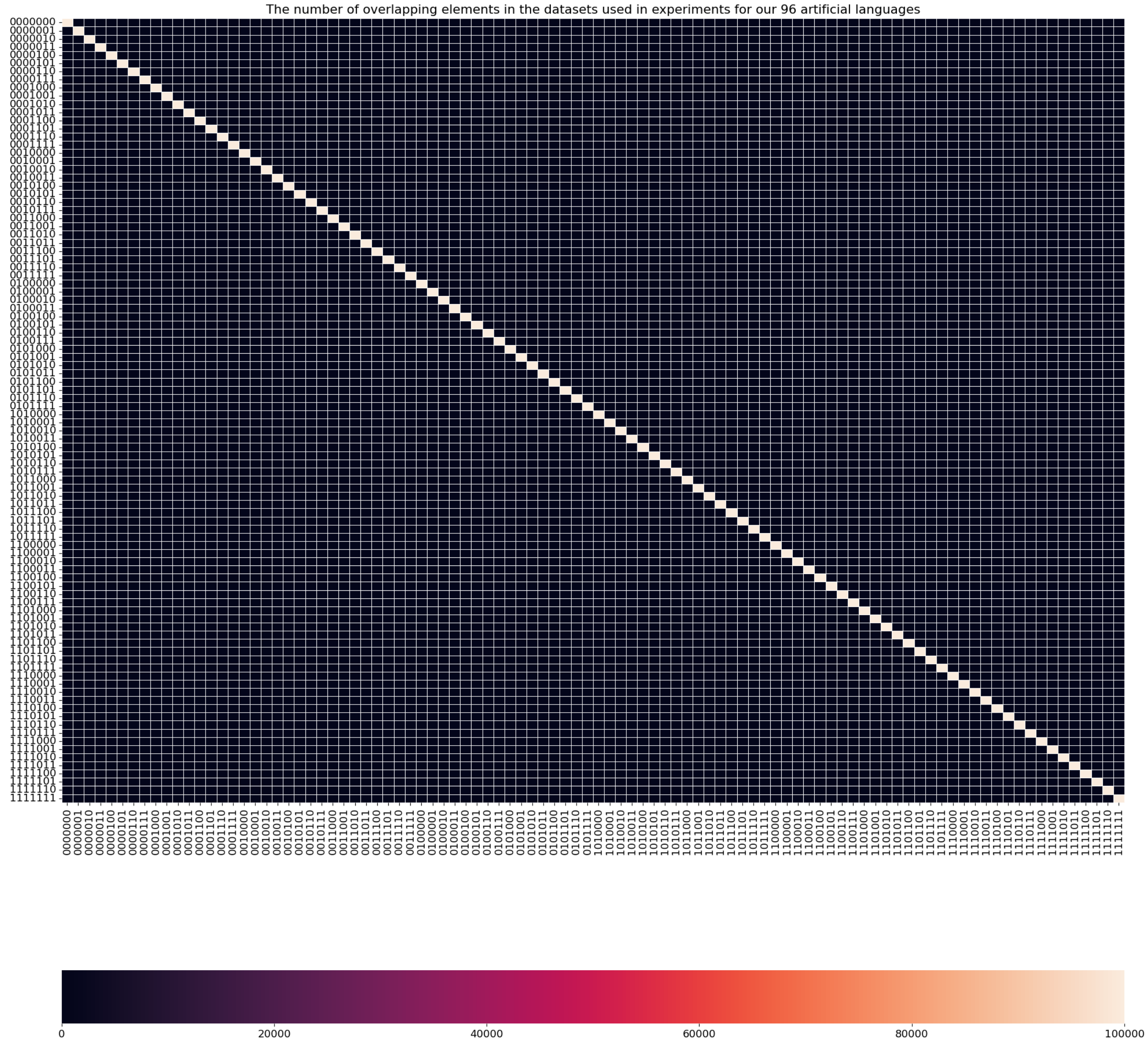


Figure 7: Heatmap showing the number of overlapping elements in the datasets for the 96 artificial languages we use in experiments.

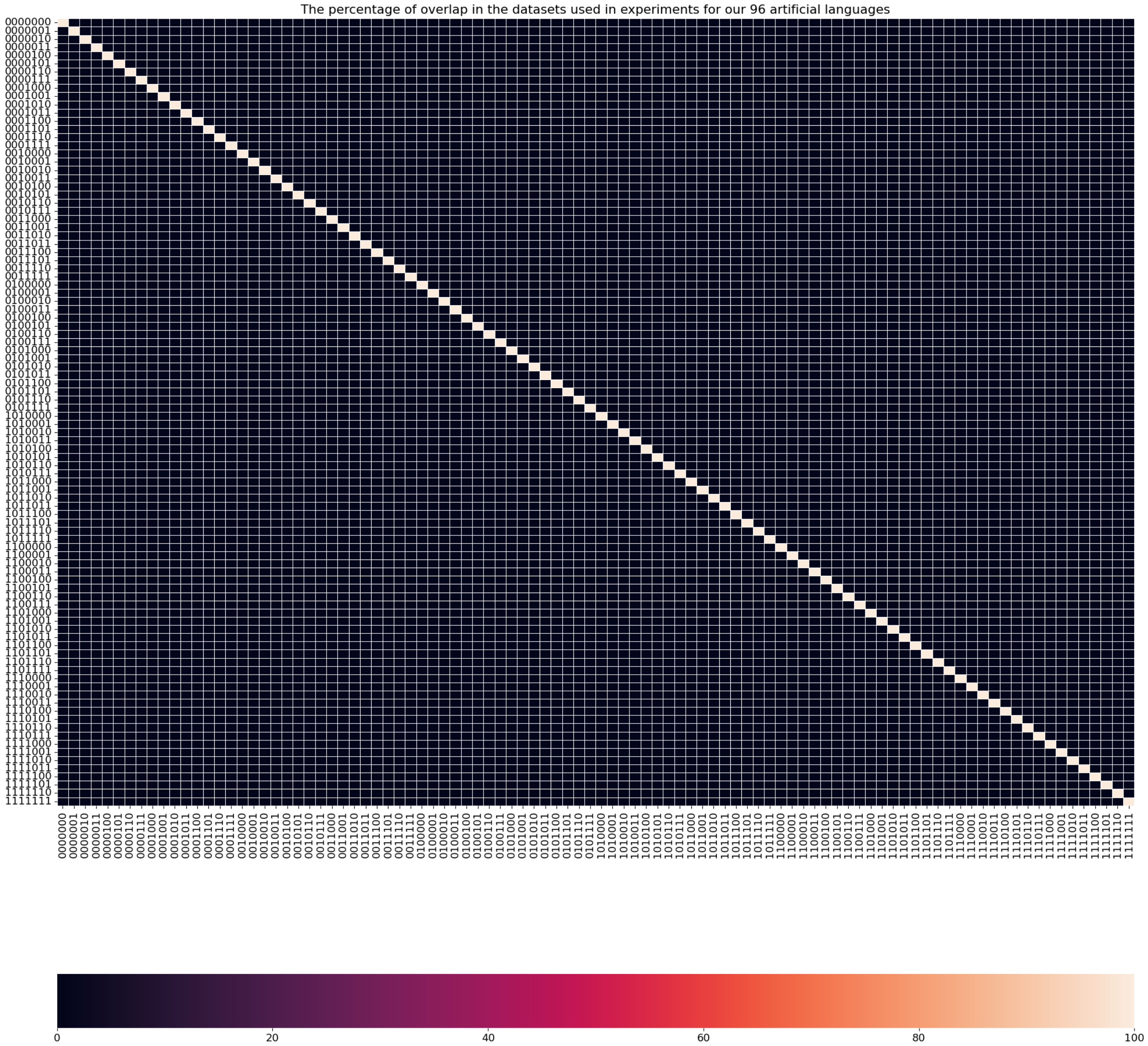


Figure 8: Heatmap showing the percentage of overlapping elements in the datasets for the 96 artificial languages we use in experiments.

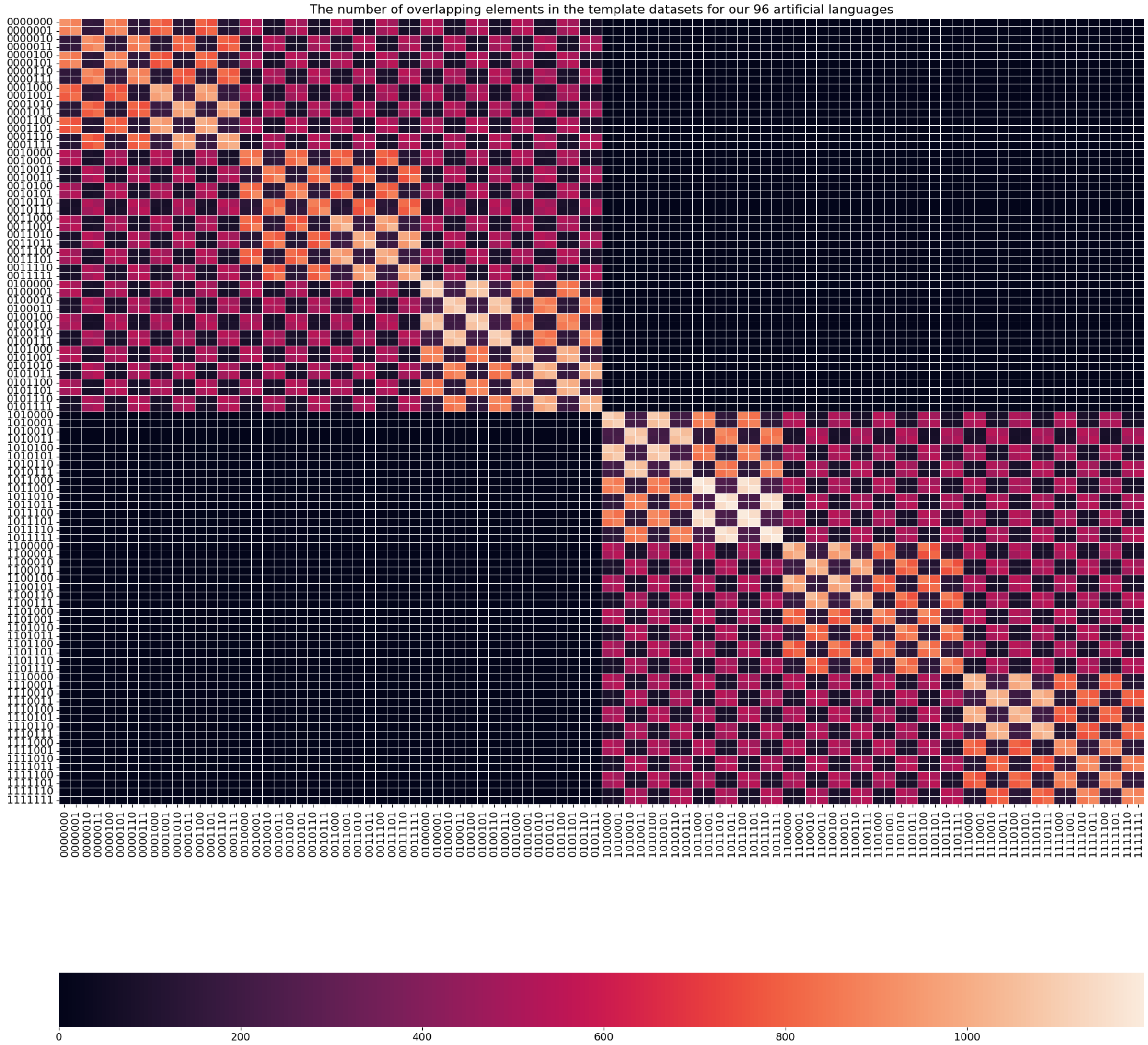


Figure 9: Heatmap showing the number of overlapping elements in the template datasets for the 96 artificial languages.

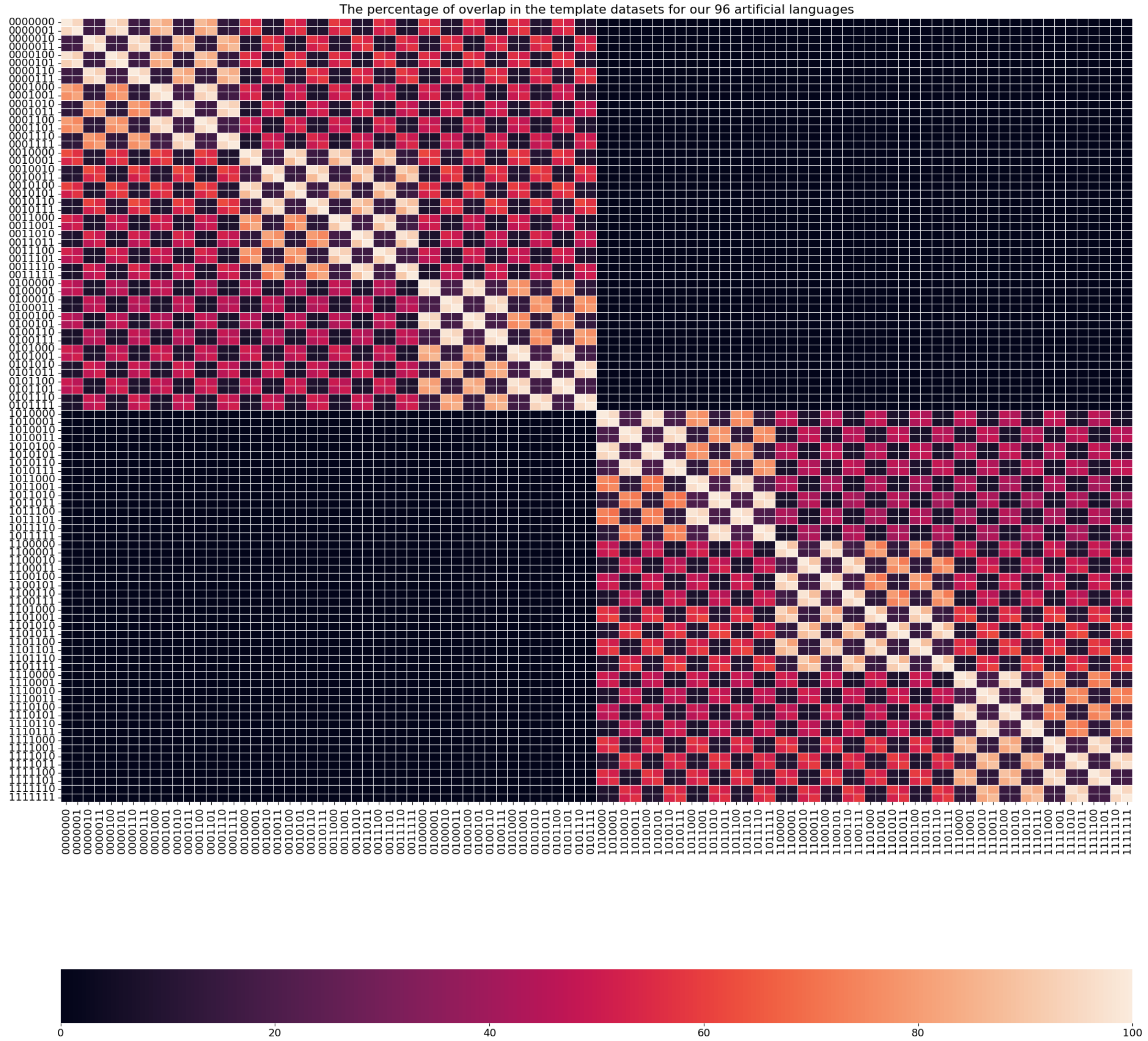


Figure 10: Heatmap showing the percentage of overlapping elements in the template datasets for the 96 artificial languages.

Beyond Accuracy: Revisiting Out-of-Distribution Generalization in NLI Models

Zahra Delbari
Tehran Institute for Advanced Studies
Khatam University, Iran
z.delbari@khatam.ac.ir

Mohammad Taher Pilehvar
Cardiff University
United Kingdom
pilehvarmt@cardiff.ac.uk

Abstract

This study investigates how well discriminative transformers generalize in Natural Language Inference (NLI) tasks. We specifically focus on a well-studied bias in this task: the tendency of models to rely on superficial features and dataset biases rather than a true understanding of language. We argue that the performance differences observed between training and analysis datasets do not necessarily indicate a lack of knowledge within the model. Instead, the gap often points to a misalignment between the decision boundaries of the classifier head and the representations learned by the encoder for the analysis samples. By investigating the representation space of NLI models across different analysis datasets, we demonstrate that even when the accuracy is nearly random in some settings, still samples from opposing classes remain almost perfectly linearly separable in the encoder's representation space. This suggests that, although the classifier head may fail on analysis data, the encoder still generalizes and encodes representations that allow for effective discrimination between NLI classes.

1 Introduction

With the rise of pre-trained language models (PLMs), NLI models have surpassed human performance on several benchmarks. However, this raises questions about whether these models truly understand the NLI task or merely exploit shortcuts and superficial patterns to achieve high accuracy without genuine linguistic comprehension. To address these concerns, researchers have developed analysis and controlled datasets to expose the limitations of NLI models, revealing their reliance on spurious correlations rather than deep linguistic understanding (McCoy et al., 2019; Ravichander et al., 2019; Naik et al., 2018a). For example, models often struggle with numerical reasoning or generalize poorly to adversarial datasets like HANS (McCoy et al., 2019). However, does evaluating a model solely based on its predicted labels provide a complete picture of what it has learned? If a model performs poorly on an out-of-distribution (OOD) dataset, can we conclusively argue that it lacks the essential knowledge for the task? Prior work challenges these assumptions. Studies show that classifier accuracy can be highly sensitive to decision thresholds (Yaghoobzadeh et al., 2021; Zhao et al., 2021a), and representation-space analyses reveal rich task-relevant structures even when classifier predictions fail (Marks and Tegmark, 2024; Amini and Ciaramita, 2023). This highlights that the representation space contains a meaningful structure beyond what accuracy captures. Similar findings exist in computer vision, where models trained on digit recognition datasets—even with some labels withheld—still cluster unseen categories meaningfully (Dyballa et al., 2024).

This paper revisits the generalization of NLI models on OOD datasets[1]: Does poor performance on OOD datasets truly indicate a lack of knowledge, or is it a symptom of misalignment between the encoder's representations and the classifier's decision boundaries? Our findings reveal that the latter could also be true in some settings. We analyze the representation space of NLI models (Section 3), focusing on linear separability (LS) across OOD datasets. The encoder representations exhibit strong LS for all datasets—even those where classifier accuracy is poor. For instance, on the Stress Test Numerical subset, the encoder representations show near-perfect LS (>96%), despite the classifier head achieving only 42% accuracy. This stark contrast suggests that the encoder captures task-relevant knowledge that the classifier fails to exploit. In Section 4, we further examine whether LS can serve as a reliable indicator of a model's knowledge in NLI, exploring encoder's behavior

---

[1] In this paper, we use *OOD* and *analysis* datasets interchangeably.

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 557–570
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

across different scenarios.

2 NLI Task and Analysis Datasets

NLI task requires determining the logical relationship between two input sentences: the premise and the hypothesis. The goal is to classify whether the hypothesis entails the premise, contradicts it, or is neutral (neither entailing nor contradicting). The Stanford Natural Language Inference (Bowman et al., 2015, SNLI) and Multi-Genre Natural Language Inference (Williams et al., 2018, MNLI) dataset are among the most widely used benchmarks for this task. Although fine-tuned PLMs achieve high performance on these benchmarks, their performance on analysis datasets suggests that these high results do not necessarily indicate a deep understanding of the task. In this section, we introduce the analysis datasets we selected for this study. These datasets are among the most popular and relatively large evaluation benchmarks for NLI, each designed to target different aspects of linguistic knowledge.

2.1 SICK

Sentences Involving Compositional Knowledge (Marelli et al., 2014, SICK) is a benchmark dataset designed for evaluating compositional distributional semantics models. Comprising over 10,000 pairs of sentences labeled as entailment, contradiction, or neutral, SICK serves as a benchmark for evaluating models' ability to handle compositional meaning and inference (see examples in Appendix Table 6).

2.2 HANS

The Heuristic Analysis for NLI Systems (McCoy et al., 2019, HANS) is a synthetic dataset created to expose the reliance of NLI models on the overlap heuristic. It features premise-hypothesis pairs where all words in the hypothesis appear within the premise. The dataset is divided into three heuristic categories based on word overlap patterns: lexical overlap, subsequence, and constituent. For each category, half of the examples align with the heuristic and are labeled as "Entailment," while the other half contradict the heuristic and are labeled as "Non-Entailment." Some examples from this dataset are provided in Appendix Table 5. NLI models often incorrectly classify samples that contradict the heuristic as "Entailment," demonstrating their reliance on superficial cues rather than true sentence understanding.

2.3 Stress Test

The Stress Test (**ST**) (Naik et al., 2018b) was designed to uncover weaknesses in models fine-tuned on the MNLI dataset by analyzing their performance on challenging validation samples. It identifies key linguistic phenomena, such as *word overlap*, *negation*, *length mismatch*, *antonyms*, *spelling errors*, and *numerical reasoning*, that frequently caused models to make errors.

To create subsets targeting these phenomena, specific strategies were applied: for *word overlap* (ST-WO) and *negation* (ST-N), phrases like "and true is true" and "and false is not true" were appended to the hypotheses. For *length mismatch* (ST-LM), the phrase "and true is true" was repeated five times at the end of the premises. *Numerical reasoning* (ST-NU) was crafted using premises extracted from the AQuA-RAT dataset, paired with generated hypotheses (see examples Appendix Table 7).[2] Except for ST-LM, model performance was significantly lower on these subsets compared to the standard validation set, with particularly poor accuracy on ST-N, where results approached random chance.

3 Representation Space and Linear Separability

Discriminative transformers are composed of two key components: the encoder, which typically uses a pre-trained language model, and the classifier head, which is usually a shallow multi-layer perceptron (MLP). In classification tasks, the `[CLS]` token, representing the entire input sequence, is passed to the classifier head to generate the final prediction. Since the `[CLS]` token encodes all the input information and serves as the primary feature for classification, our investigation centers on understanding its representation within the model.

3.1 Experimental Setup

Baseline models. We explore the representation space produced by the `[CLS]` token across three models: RoBERTa (Liu et al., 2019b), BERT (Devlin et al., 2019), and DistilBERT (Sanh et al., 2020). For consistency, we employ the base versions of all models. While BERT has been the focal point in most analytical works, our study extends this analysis to RoBERTa, known for its robust-

---

[2] We dismiss anatomy subset because it contains samples of only one class.

Dataset	DistilBERT	BERT	RoBERTa
MNLI-m	82.1±0.2	84.3±0.4	87.5±0.1
MNLI-mm	82.2±0.2	84.4±0.5	87.4±0.2
SICK	54.4±0.6	56.4±0.8	57.5±0.5
HANS+	97.3±0.8	97.7±1.2	98.7±0.1
HANS−	9.6±2.7	32.4±5.5	50.1±2.0
ST-NU	35.1±1.5	42.6±1.7	59.5±2.9
ST-LM	80.1±0.2	82.3±0.3	85.2±0.2
ST-N	54.6±1.0	56.0±0.3	57.1±0.7
ST-WO	60.1±1.3	59.0±1.3	63.0±2.7

Table 1: Accuracy of the three baseline models on NLI analysis datasets SICK, HANS, and Stress Test (ST-X), as well as the standard validation sets MNLI matched (-m) and MNLI mismatched (-mm), reported for five runs.

ness, and DistilBERT, a more lightweight alternative with less capacity to gain knowledge.[3]

Datasets. We fine-tune the baseline models on the MNLI and SNLI datasets. Then, we examine the `[CLS]` token generated by these models for analysis datasets mentioned in Section 2. Since the training datasets have three labels (*entailment*, *contradiction*, and *neutral*), while HANS only has two (*entailment* and *non-entailment*), we map both *contradiction* and *neutral* predictions to *non-entailment* and leave *entailment* unchanged.

Fine-tuning. Each fine-tuning run consists of training the models for 5 epochs with a learning rate of 2×10^{-5}, a batch size of 32, the AdamW optimizer, and a learning rate decay of 0.02.

Dimension reduction. To gain a deeper understanding of the representation space in classification, we visualized it by plotting the representations. Since the embedding space is high-dimensional (768 for base models), we applied Principal Component Analysis (PCA) to reduce the dimensionality to three, allowing for a clearer visualization. The reduced space captures approximately 77% of the total variance, with each remaining component contributing less than 2%, as shown in Figure 8 in the Appendix. Therefore, this three-dimensional representation provides a reasonable approximation of the original high-dimensional space.

3.2 Representation Space Visualization

The average performance of all baselines models are reported in Table 1 for MNLI and in Table 8 (in the Appendix) for SNLI. Consistent with the purpose of the HANS dataset, the table confirms that all models tend to classify HANS samples as *entailment*, achieving near-perfect results on HANS+ but very poor performance on HANS−, which indicates a strong reliance on overlap heuristics. For the Stress Test dataset, the results for the ST-NU subset are particularly poor, with performance close to random chance for DistilBERT, suggesting that these models struggle to infer anything meaningful from mathematical or equation-based samples.

Figure 1 illustrates the representation space of one trial from each model. Given that HANS has the largest sample size (30000) compared to the other datasets, we find it clearer to visualize its representation in relation to the other analysis datasets. As a result, all visualizations are for HANS unless otherwise specified. To match the number of HANS samples, we selected 30K MNLI (train) samples and plotted the 3D space for all 60K data points.[4]

All the models show distinct regions within the representational space, with each region corresponding to one class of MNLI. This structure enables the classifier head to achieve linear separation. The representational space can be visualized as a three-petaled flower, with each petal representing one of the three classes.

For the HANS dataset, however, the data is positioned beneath these petals. If the model's accuracy (trained on MNLI) on HANS matched its performance on MNLI, we would expect the data points to be similarly organized into distinct petals. Instead, the majority of the HANS data is concentrated in the (blue) petal corresponding to the *entailment* label, which cause the poor accuracy presented in Table 1.

But the interesting point is that despite the clustering of HANS data in the entailment region, the orange and yellow points—representing entailment and non-entailment labels, respectively—are still clearly separated. This suggests that although the HANS data is incorrectly categorized according to the standard regions determined by the classifier head, the opposite labels remain well-separated in

---

[3] We also checked BERT-large, and the LS remains strong despite poor accuracy.

[4] Given the challenges of displaying 3D images, we provide 2D views from different angles to offer a clearer understanding of the 3D representation space.

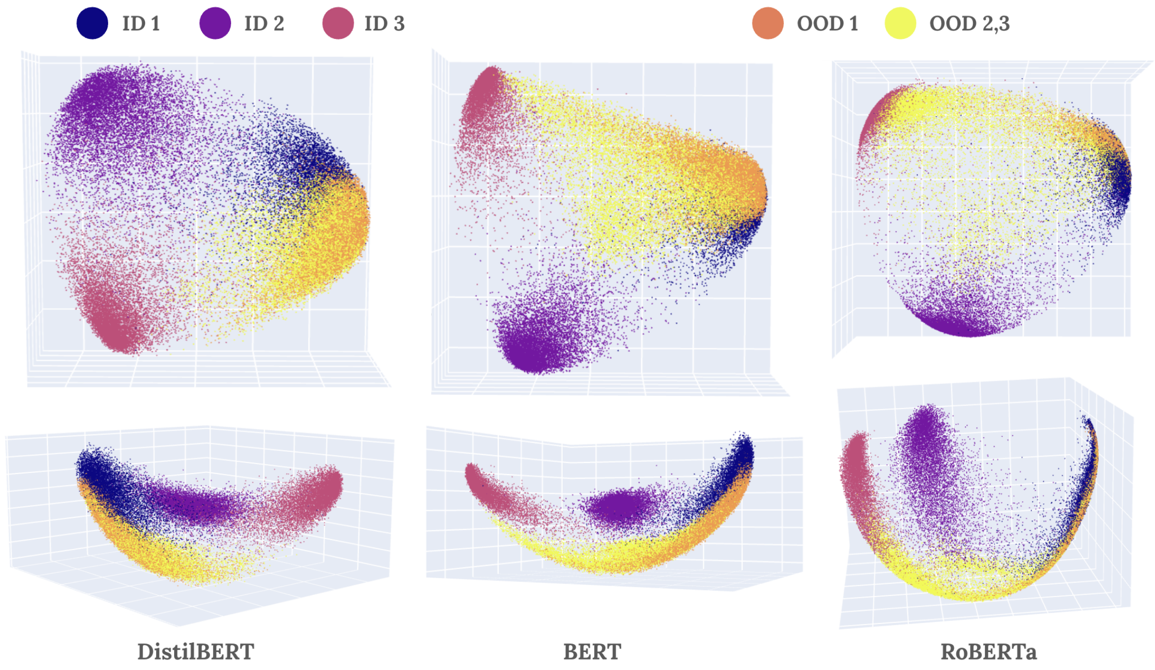


Figure 1: 3D visualization of the [CLS] token representation space for the MNLI (in-distribution, ID) and HANS (OOD) datasets, generated by the three baseline models. Colors indicate the gold labels. In all baseline models, the orange and yellow points (representing the two classes of HANS) are clearly distinguishable. The 3D spaces are visualized from two different perspectives (class 1: Entailment, class 2: Neutral, and class 3: Contradiction). Despite encoder's positioning of the OOD samples towards ID1, they are internally separated for their two classes (OOD1, and OOD2,3), as particularly visible from the BERT visualization (middle, top).

the representational space. For additional clarity, see Figure 2, which compares model outputs (labels given by the classifier head) (2b) with the gold labels (2a) of HANS.

3.3 Linear Separability (LS)

To evaluate whether the encoder's `[CLS]` embeddings admit **linear separability** between classes, we formalize the problem as follows. Let $\mathbf{h}_i \in \mathbb{R}^d$ denote the last layer hidden state of the `[CLS]` token for the i-th input sample, and $y_i \in \{1, \ldots, K\}$ its corresponding class label. We assess whether there exists a linear decision boundary that separates classes in the embedding space. This reduces to solving for parameters $\mathbf{W} \in \mathbb{R}^{K \times d}$ and $\mathbf{b} \in \mathbb{R}^K$ such that $\hat{y}_i = \arg\max_k (\mathbf{W}\mathbf{h}_i + \mathbf{b})$ achieves minimal cross-entropy loss over N samples:

$$\mathcal{L}(\mathbf{W}, \mathbf{b}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{I}[y_i = k] \log \left(\frac{\exp(\mathbf{W}_k \mathbf{h}_i + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{W}_j \mathbf{h}_i + b_j)} \right), \quad (1)$$

where $\mathbb{I}[\cdot]$ is the indicator function. High accuracy on held-out data implies the existence of a hyperplane $\mathbf{W}_k \mathbf{h} + b_k = \mathbf{W}_{k'} \mathbf{h} + b_{k'}$ separating classes k and k'.

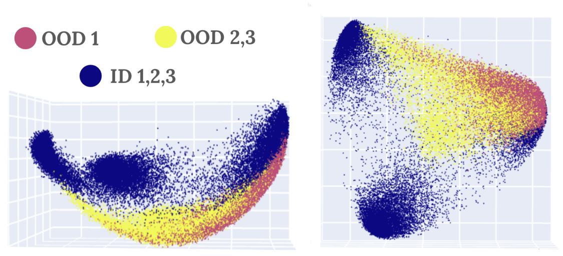
3.4 LS Results

Table 2 quantifies the degree of LS for all analysis datasets across the baseline models fine-tuned on MNLI and SNLI. For comparison, we also present the results of a random experiment, where the labels of the `[CLS]` token are shuffled randomly, and decision boundaries are then computed (in parentheses). It is important to note that, in higher-dimensional spaces and when the dataset size is small, the accuracy of purely random data can exceed the expected accuracy (50% for two classes and 33% for three classes).

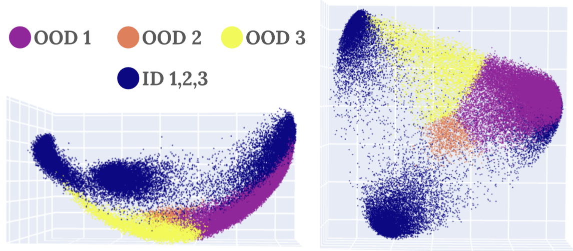
Universal Linear Separability (LS) All models achieve high LS scores (77–98% for models fine-tuned on MNLI), confirming that learned representations inherently encode task-relevant features rather than relying on superficial patterns. This is particularly evident in HANS, where LS exceeds 90% (e.g., BERT: 95.6%). Despite the high degree of overlap between entailment and non-

	MNLI			SNLI		
Dataset	**DistilBERT**	**BERT**	**RoBERTa**	**DistilBERT**	**BERT**	**RoBERTa**
MNLI	80.1±0.2 (48.2)	82.4±0.1 (48.0)	86.4±0.3 (47.7)	70.3±0.2 (48.5)	74.6±0.3 (48.4)	79.4±0.3 (48.5)
SNLI	77.8±1.2 (49.0)	82.0±1.2 (48.8)	85.6±0.2 (48.1)	85.6±0.5 (48.7)	88.2±0.6 (48.9)	89.3±0.8 (49.0)
SICK	84.4±0.8 (59.7)	87.4±0.8 (60.0)	89.4±0.6 (59.5)	86.6±1.5 (59.9)	87.8±0.8 (59.9)	89.7±0.5 (59.8)
HANS	91.1±0.5 (56.1)	95.6±0.7 (56.2)	95.4±0.7 (55.9)	88.4±0.8 (56.0)	93.9±0.6 (56.3)	95.6±0.4 (56.1)
ST-LM	77.7±0.2 (48.0)	80.3±0.2 (47.7)	84.3±0.3 (47.5)	69.0±0.3 (48.6)	72.7±0.2 (48.6)	77.2±0.3 (48.1)
ST-N	77.6±0.5 (48.3)	80.1±0.3 (47.9)	84.0±0.3 (47.3)	67.5±0.3 (48.0)	71.6±0.7 (48.3)	76.1±0.5 (47.7)
ST-WO	78.4±0.1 (48.0)	80.9±0.3 (48.0)	84.8±0.3 (47.5)	69.0±0.3 (48.3)	72.8±0.5 (48.6)	77.5±0.5 (48.5)
ST-NU	96.3±0.4 (51.5)	97.4±0.6 (51.5)	98.4±0.9 (50.3)	94.2±1.3 (51.5)	96.3±0.6 (51.6)	98.4±0.2 (51.7)

Table 2: Results of linear separability for analysis datasets, based on models fine-tuned on MNLI and SNLI. The linear separability is the accuracy of linear boundaries reported for the evaluation set of HANS, the mismatched subsets of the ST datasets (ST-X), and MNLI, as well as the validation sets of SICK and SNLI. The numbers in parentheses represent results from random experiments.



(a) Gold Labels



(b) Predicted Labels

Figure 2: A comparison of (a) the scatter of [CLS] tokens for two HANS classes in space and (b) how a fine-tuned BERT model classifies them into three classes, with MNLI data included for reference.

entailment data points in HANS, the models do not treat them as identical—contrary to what accuracy in Table 1 suggests. For ST-LM, ST-N, ST-WO, and SICK, LS is slightly less pronounced compared to HANS. The most striking result comes from ST-NU (numerical reasoning), where all baseline models achieve over 95% LS. Although the classifier head's poor accuracy suggests that models struggle with numerical reasoning, the high LS indicates that they effectively capture the necessary information for this task.

Accuracy Paradox While classifier head accuracy suggests that ST-NU and HANS are difficult, and ST-WO and ST-N are easier, the representation space reveals the opposite. ST-WO and ST-N are as challenging as the MNLI validation set, while ST-NU and HANS are much easier. Notebly, ST-WO, ST-LN, and ST-LM, which are derived from the MNLI validation set with some modifications, exhibit LS values similar to MNLI itself. This is an interesting finding, as it suggests that these subsets, being structurally similar to MNLI, pose comparable challenges for the model. Since they are as difficult as MNLI, their LS does not exceed MNLI accuracy or reach the high LS values observed in easier datasets like HANS, SICK, and ST-NU.

4 LS as Evidence of Encoder Knowledge

In the previous section, we observed that despite the NLI model's poor accuracy on the analytical dataset, their encoder's outputs remain nearly linearly separable. In this section we argue that low accuracy does not necessarily indicate a lack of NLI or linguistic knowledge. Instead, our results highlight a misalignment between the encoder's learned representations and the classifier head's decision boundaries.

4.1 LS and Training Dynamics

In traditional machine learning, feature engineering was guided by domain experts who carefully crafted features based on their deep understanding of the task. These features were designed to effectively differentiate between classes, making them easy to separate with a simple MLP. In contrast, transformer models delegate this responsibility to

Model	MNLI	H+	H-	HANS
BERT_{Full}	84.7±0.2	97.7±1.2	32.4±5.5	65.0±2.6
$\text{BERT}_{Balanced}$	81.6±0.4	79.1±3.5	48.8±3.4	63.9±1.2

Table 3: Comparison of BERT model accuracy when fine-tuned on the full MNLI dataset (with $392K$ samples) and the balanced dataset (with $235K$ samples). The mean accuracy is reported over 5 different seeds.

the encoder, which is tasked with generating meaningful representations from raw input data. The classifier head, on the other hand, merely maps these representations to labels without any inherent understanding of the task itself. If a model truly grasps the underlying task, this understanding should be reflected in the features produced by the encoder. The fact that the encoder can generate linearly separable features, even for datasets that differ significantly from the training data, suggests that it has captured genuine, task-relevant knowledge. Moreover, we demonstrate that this LS is not just an artifact of the model's representation but also correlates with its process of acquiring knowledge during training. By varying the amount of training data and limiting the number of update steps, we explored the relationship between task understanding (as reflected by standard validation set accuracy) and LS of analysis dataset, with the following findings:

- **Effect of Training Data Size:** Fine-tuning BERT on varying proportions of the MNLI dataset (from 5% to 100%) revealed a clear trend, as the amount of training data increased, LS improved for both the MNLI validation set and the HANS dataset (Figure 3).
- **Effect of Training Iterations:** Similarly, tracking the model's performance on the full dataset at 500-step intervals (Figure 9 in the Appendix) showed that as validation accuracy increased, the LS of analysis datasets also improved.

These findings suggest that as the model refines its understanding of the NLI task, it simultaneously enhances its ability to produce clearer and more distinguishable representations, reinforcing the connection between knowledge acquisition and LS.

4.2 Re-evaluating the Lexical Overlap Bias in NLI Models

One common argument against NLI models achieving true linguistic mastery is their poor perfor-

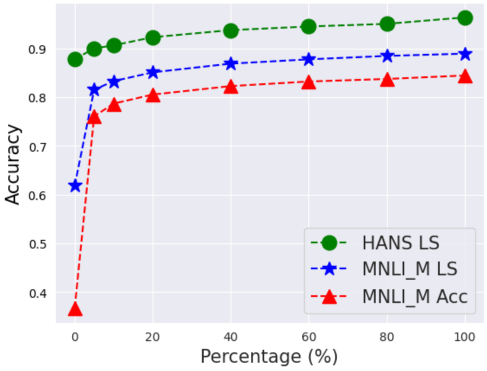


Figure 3: LS of the HANS and MNLI (matched) datasets for BERT fine-tuned on different percentages of the MNLI dataset, along with model accuracy. The consistent rise in LS alongside accuracy shows that improved LS is not incidental but emerges as the model learns the task more deeply with more data; reflecting the accumulation of generalized, task-relevant knowledge in the encoder.

mance on heuristic-based datasets. This is often cited as evidence that these models rely on shortcuts in the training data rather than acquiring genuine linguistic knowledge. HANS, as a prominent example of such datasets, is frequently used to support this claim due to its design, which specifically targets lexical overlap heuristics. Since we argue that the model does acquire sufficient linguistic knowledge, we challenge this assumption by conducting an experiment to remove the potential influence of lexical overlap bias and examine whether the model's performance improves.

To explore this, we calculated the overlap percentage for all training examples and grouped them into 100 bins, each representing a 1% range (e.g., [88, 89) overlap). Within each bin, we ensured an equal distribution of examples across all three labels by selecting a balanced number of samples from the least frequent label. This process eliminated label imbalance across different levels of lexical overlap, as shown in Figure 4. Using this balanced dataset, we fine-tuned a BERT model for five epochs, with results reported in Table 3. While accuracy on HANS$-$ improved, this came at the cost of decreased accuracy on HANS$+$, leading to an overall drop in HANS performance compared to the model trained on the full MNLI dataset. Figure 5 visualizes the representation space of the `[CLS]` tokens from the model trained on the balanced dataset. The HANS representations remain largely clustered together within the entailment region, rather than forming distinct groups. If the

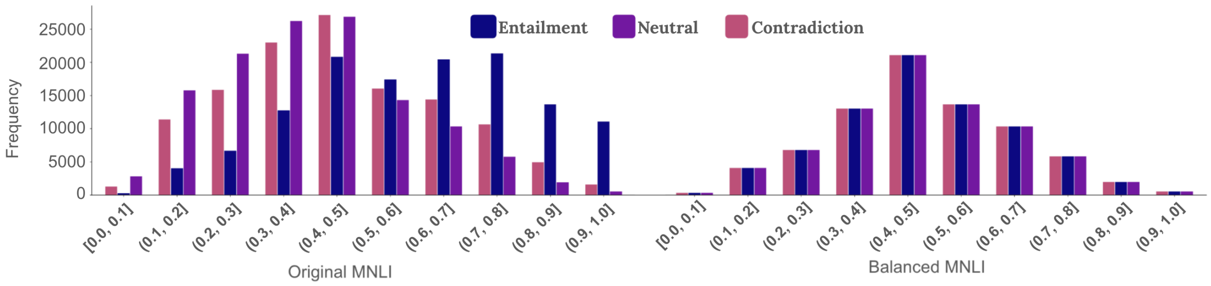


Figure 4: Histograms of label frequency across different overlap percentages, before and after balancing the dataset. The original experiment used 100 bins, but for the sake of space, we present both histograms with 10 bins.

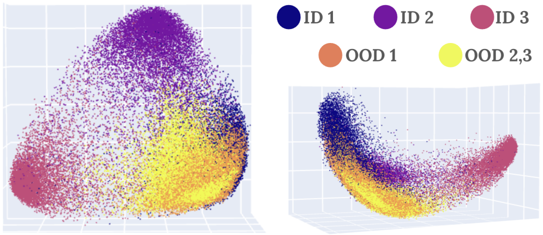


Figure 5: Visualization of the `[CLS]` representation space for BERT fine-tuned on the blended MNLI dataset. Colors indicate the gold labels.

overlap heuristic was the primary cause of the bias, balancing the dataset should have improved the results.

4.3 Effect of Random Seed on Performance

Prior works (McCoy et al., 2020; Zhou et al., 2020) have reported that models trained on standard NLI datasets exhibit consistent in-domain (ID) validation performance across different random seeds, yet their performance on challenge datasets (OOD cases) such as HANS fluctuates significantly. In some subsets of HANS, accuracy varies between 0% and 66% depending on the seed. As shown in Table 1, accuracy variance is large for HANS and ST-NU, whereas the MNLI validation set shows almost no variance. Notably, these results are based on only five random seeds; increasing the number of trials would likely reveal even greater variance.

Based on these results, prior work suggests that while the model consistently learns patterns that perform well on the validation set, its generalization to OOD or adversarial cases is unstable. However, a closer analysis points to an alternative explanation. The encoder, which encodes linguistic knowledge, exhibits a high degree of consistency across random seeds. Its representations maintain LS even for adversarial inputs, regardless of initialization. In Table 4 we compare the accuracy and LS of two BERT models with two different initial seeds, one with very poor HANS performance and one with very strong performance, yet their encoder representations remain distinguishable in the same way. This suggests that the encoder reliably captures task-relevant linguistic features which are preserved across seeds.

Trial	HANS Accuracy	Linear Separability
High-performing	67.6	95.5
Low-performing	52.8	95.0

Table 4: Comparison of HANS accuracy and LS for a high-performing and a low-performing trials.

Instead, the classifier head—a shallow, randomly initialized MLP—is highly sensitive to weight initialization. Different random seeds result in divergent decision boundaries within the encoder's representation space. While these boundaries work well for ID validation data (MNLI), they fail to generalize to OOD datasets like HANS. This is because the classifier is primarily optimized for MNLI's feature distribution, which does not necessarily align with the structure of adversarial or OOD samples.

Thus, rather than instability arising from differences in learned knowledge, it stems from the classifier's inconsistent mapping of the encoder's representations, leading to poor generalization beyond the training domain.

5 Discussion

We have shown that despite the poor and unstable performance of NLI models on OOD datasets, the encoder representations of these datasets remain consistently and highly discriminative with respect to class labels. This suggests that the model acquires core linguistic knowledge relevant to the

NLI task that generalizes beyond the training distribution. If this were not the case, it would be unclear why the encoder organizes unseen data in a way that permits linear separation. Notably, this behavior does not universally occur for all types of OOD data; it stands in contrast to tasks such as paraphrase detection, where the encoder often fails to produce similarly structured representations.

For example, QQP is a standard benchmark for paraphrase detection, while PAWS (Zhang et al., 2019) was introduced to challenge models that rely on shallow heuristics such as word overlap. A BERT model fine-tuned on QQP performs poorly on PAWS, misclassifying most examples as paraphrases, despite nearly half being non-paraphrases. In this case, LS is close to random—61.4% compared to 57.2%—and the PAWS examples appear scattered within the QQP duplicate region in the representation space, as shown in Figure 6.

As discussed in Section 4.3, one potential explanation for why encoder representations can be discriminative for OOD datasets despite low accuracy is that, from the perspective of the PLM, the MNLI dataset occupies a distinct and well-defined region in the representation space, whereas analytical datasets reside elsewhere (Figure 7). During fine-tuning, the encoder and classifier head are updated jointly to establish decision boundaries. However, this optimization process focuses only on MNLI training examples, which are explicitly supervised. As a result, the encoder is shaped to structure MNLI data effectively while ignoring how these changes affect other parts of the space. Since OOD data are not included during training, misalignments in those regions incur no penalty, leading to reduced generalization performance.

It is important to emphasize that the LS values we report are not the result of any additional training. Rather, they reflect the decision boundaries already present in the representation space after fine-tuning. This distinction is critical, as it rules out multitask learning as a source of the observed patterns. In multitask learning, the encoder is jointly trained on multiple objectives, encouraging knowledge sharing across tasks. In our setting, however, the encoder is fine-tuned solely on MNLI, and the analysis datasets are never seen during training. We simply train a linear classifier on frozen representations using cross-entropy loss, thereby probing the task-relevant structure already encoded by the model.

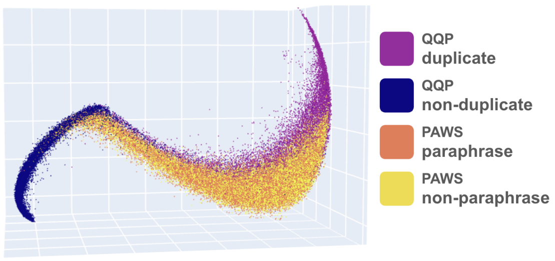


Figure 6: Representational space of the [CLS] token generated by the BERT model fine-tuned on the QQP dataset.

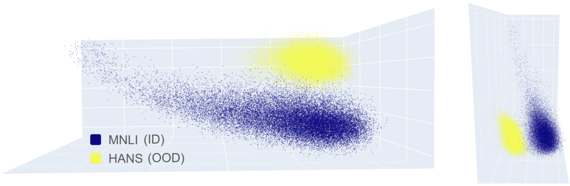


Figure 7: Visualization of the [CLS] representation of the MNLI training set and HANS from the perspective of pre-trained BERT.

6 Related Work

6.1 Probing Knowledge

Probing the representation space of PLMs has been central to understanding the knowledge they encode. Early studies analyzed layer-wise representations to identify where syntactic and semantic information is captured, revealing a hierarchical organization of linguistic features (Liu et al., 2019a; Jawahar et al., 2019; Tenney et al., 2019). Follow-up work examined attention mechanisms, showing that specific attention heads specialize in tasks such as coreference and syntax (Clark et al., 2019; Voita et al., 2019). Other approaches explored the geometry of the representation space, finding that upper layers tend to produce more context-specific, anisotropic embeddings (Ethayarajh, 2019). While initial work focused on static PLMs, later studies investigated how fine-tuning alters representations, showing that core structural properties often remain stable despite task-specific adaptations (Merchant et al., 2020; Zhou and Srikumar, 2022).

6.2 Discrepancies Between Final Predictions and Model Representations

Yaghoobzadeh et al. (2021) showed that adjusting the classification threshold for HANS data can significantly impact BERT's accuracy. This phenomenon is not exclusive to encoder models, generative models also exhibit discrepancies between what they learn and what their final outputs imply. Zhao et al. (2021b) highlighted a similar issue in generative models, showing that the structure of a prompt can influence the threshold required for classification tasks such as sentiment analysis. By calibrating models with a null input, they achieved more reliable results. Amini and Ciaramita (2023) argue that the sensitivity of encoder-decoder model to instruction phrasing stems from the constraint that models must verbalize their predictions. By bypassing the decoding step and directly probing the encoder representations, they achieved more stable and improved results. Marks and Tegmark (2024). Furthermore, Marks and Tegmark (2024) found that LLMs encode the truth or falsehood of factual statements in a linear manner, despite their tendency to generate incorrect information.

6.3 Instability in OOD Generalization

Models that appear stable and performant on standard ID test sets often exhibit significant variability when evaluated on OOD datasets (McCoy et al., 2020; Zhou et al., 2020), raising concerns about their generalization capabilities. Similarly, Zhao et al. (2021b) demonstrated that even powerful generative models like GPT-3 suffer from notable instability in few-shot learning scenarios. This instability has been attributed to several factors, including catastrophic forgetting during fine-tuning (Lee et al., 2020), limited size and diversity of available datasets (Dodge et al., 2020), and optimization difficulties such as vanishing gradients in deeper architectures (Mosbach et al., 2021). In addition to these architectural and data-related challenges, the structure of prompts and the order in which training examples are presented have also been shown to significantly influence performance in few-shot settings (Zhao et al., 2021b), highlighting the sensitivity of model behavior to seemingly minor variations in input.

7 Conclusion

In this paper, we revisited the performance of fine-tuned PLMs on challenging NLI datasets. Our experiments revealed that, despite poor classifier accuracy, the encoder's representation space often demonstrates clear linear separability between classes. This suggests that the models possess relevant task-specific knowledge, but there is a misalignment between the classifier's decision boundaries and the knowledge embedded in the encoder's representations. While we proposed some hypotheses for this misalignment, further in-depth investigation is required, which we leave for future work.

8 Limitations

One limitation of this study is that the analysis was limited to three pretrained language models—DistilBERT, BERT, and RoBERTa. While these models are widely used, they do not represent the full spectrum of transformer-based models, and therefore, the findings may not be fully generalizable to newer or more specialized models. Additionally, this study does not provide a direct solution for improving classification accuracy. Although we demonstrate the existence of linear boundaries, determining the optimal decision boundaries for each dataset still requires access to the full dataset, which may not be efficient or feasible for OOD datasets. Furthermore, relying on linear separability as a proxy for model knowledge may oversimplify the complexity of how models truly understand the nuances of inference. There is room for further exploration using alternative probing techniques to assess and deepen our understanding of model comprehension.

References

Afra Amini and Massimiliano Ciaramita. 2023. In-context probing: Toward building robust classifiers via probing large language models. *Preprint*, arXiv:2305.14171.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *Preprint*, arXiv:2002.06305.

Luciano Dyballa, Evan Gerritz, and Steven W. Zucker. 2024. A separability-based approach to quantifying generalization: which layer is best? *Preprint*, arXiv:2405.01524.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. 2020. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *Preprint*, arXiv:2310.06824.

R. Thomas McCoy, Junghyun Min, and Tal Linzen. 2020. BERTs of a feather do not generalize together: Large variability in generalization across models with similar test set performance. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 217–227, Online. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018a. Stress test evaluation for natural language inference. In *The 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, New Mexico, USA.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018b. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

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, Florence, Italy. Association for Computational Linguistics.

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Yadollah Yaghoobzadeh, Soroush Mehri, Remi Tachet des Combes, T. J. Hazen, and Alessandro Sordoni. 2021. Increasing robustness to spurious correlations using forgettable examples. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3319–3332, Online. Association for Computational Linguistics.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021a. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021b. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Xiang Zhou, Yixin Nie, Hao Tan, and Mohit Bansal. 2020. The curse of performance instability in analysis datasets: Consequences, source, and suggestions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8215–8228, Online. Association for Computational Linguistics.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

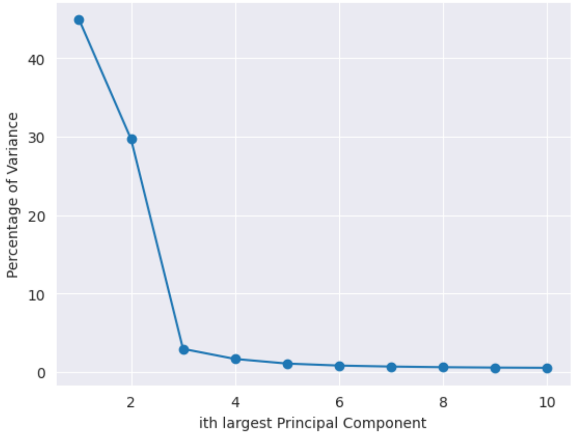


Figure 8: The percentage of variance explained by each of the 10 greatest principal components of PCA applied to the `[CLS]` tokens from the HANS and MNLI training sets.

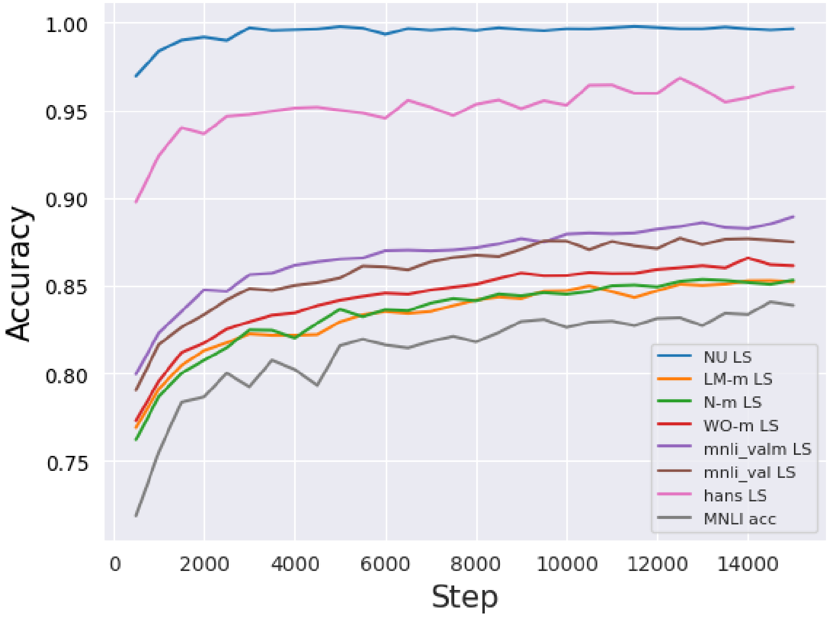


Figure 9: Linear separability (LS) of the analysis datasets along with the accuracy of BERT reported at every 500 optimization steps.

Heuristic	**Hypothesis**	**Premise**	**Label**
Lexical Overlap	The banker near the judge saw the actor.	The banker saw the actor.	E
	The lawyer was advised by the actor.	The actor advised the lawyer.	E
Subsequence	The doctors visited the lawyer.	The lawyer visited the doctors.	N
	The judge by the actor stopped the banker.	The banker stopped the actor.	N
	The artist and the student called the judge.	The student called the judge.	E
	Angry tourists helped the lawyer.	Tourists helped the lawyer.	E
	The judges heard the actors resigned.	The judges heard the actors.	N
	The senator near the lawyer danced.	The lawyer danced.	N
Constituent	Before the actor slept, the senator ran.	The actor slept.	E
	The lawyer knew that the judges shouted.	The judges shouted.	E
	If the actor slept, the judge saw the artist.	The actor slept.	N
	The lawyers resigned, or the artist slept.	The artist slept.	N

Table 5: Examples from the HANS dataset, where E represents the entailment label and N represents the non-entailment label.

Premise	**Hypothesis**	**Label**
A woman is painting a man's face	Makeup is being put on the man by the woman	Entailment
The women are singing and dancing	The persons are singing and dancing	Entailment
A group of kids is playing in a yard and an old man is standing in the background	A group of boys in a yard is playing and a man is standing in the background	Neutral
A black, red, white and pink dress is being worn by a woman	A woman is taking off a cloak, which is very large, and revealing extravagant apparel	Neutral
Two large persons are sitting on a park bench and they have a bottle of soda between them	Two large persons are standing near a park bench and they have nothing between them	Contradiction
A man is playing a violin	There is no man playing a violin	Contradiction

Table 6: Examples from the SICK dataset.

Subset	**Premise**	**Hypothesis**	**Label**
LM	The chain wielder smiled at her and true is true and true is true and true is true and true is true and true is true.	The chain wielder was friends with the woman.	Neutral
	The two programs are currently housed in buildings about a block apart and true is true and true is true and true is true and true is true and true is true	The two buildings are on opposite sides of the city.	Contradiction
	oh i did and i laughed real hard when i took it in for the two thousand mile checkup and uh and true is true and true is true and true is true and true is true and true is true	I had a good laugh when I took it for the two thousand mile checkup.	Entailment
WO	I can't do any jumping up and down because it makes it hurt.	The pain is too much after jumping, it needs surgery and true is true.	Neutral
	Both professors soon realized that creating a new language was not an easy task.	Professors realized it was easy to make a new language and true is true.	Contradiction
	Well that's good that's great.	That is not bad, it is actually pretty good and true is true.	Entailment
N	They did this to us.	The practical joke was played on them and false is not true.	Neutral
	No, Dave Hanson, you were too important to us for that.	Yes, Dave Man, you are not important to us and false is not true.	Contradiction
	Then he sobered.	He had sobered up and false is not true.	Entailment
NU	Jose joined him less than 6 months later, investing Rs.	Jose joined him 2 months later, investing Rs.	Neutral
	On every Adidas shoe the restaurant makes a profit of $ 10 while on every Puma shoe it makes a profit of $ 8	On every Adidas shoe the restaurant makes a profit of $ more than 10 while on every Puma shoe it makes a profit of $ 8	Contradiction
	A train leaves Delhi at 9 a.m.	A train leaves Delhi at more than 3 a.m.	Entailment

Table 7: Examples from different subsets of the stress test dataset.

Dataset	**DistilBERT**	**BERT**	**RoBERTa**
SNLI	89.3±0.1	90.9±0.3	91.8±0.1
SICK	53.5±1.0	56.6±0.3	57.1±0.4
HAN	52.9±0.6	58.9±1.1	66.6±1.0
ST-NU	35.3±0.7	37.8±4.6	38.1±2.1
ST-LM	65.1±0.7	70.6±0.6	76.6±0.2
ST-N	45.8±2.3	51.4±2.5	63.4±1.6
ST-WO	56.7±3.6	59.2±2.6	69.8±1.8

Table 8: Accuracy of SNLI fine-tuned models on NLI analysis datasets, SICK, HANS, and Stress Test (ST) alongside the standard validation sets of SNLI.

Spatial relation marking across languages: extraction, evaluation, analysis

Barend Beekhuizen
Department of Linguistics, University of Toronto
Department of Language Studies, University of Toronto, Mississauga
`barend.beekhuizen@utoronto.ca`

Abstract

This paper presents a novel task, detecting Spatial Relation Markers (SRMs, like English ***in** the bag*), across languages, alongside a model for this task, RUIMTE . Using a massively parallel corpus of Bible translations, the model is evaluated against existing and baseline models on the basis of a novel evaluation set. The model presents high quality SRM extraction, and an accurate identification of situations where language have zero-marked SRMs.

1 Introduction

Massively parallel corpora, where the same source text has been translated into many different languages, form a unique opportunity to compare how the languages of the world express the same message, allowing for both fine-grained (utterance level) and large-scale comparisons (Mayer and Cysouw, 2012; Wälchli, 2014; Levshina, 2016; Asgari and Schütze, 2017; Liu et al., 2023). While earlier studies (e.g. Wälchli, 2014) applied manual extraction procedures, automated methods for retrieving the translation-equivalent markers across languages, at the level of parallel utterance tokens, are necessary to study crosslinguistic variation at scale. Substantial progress has been made in developing such methods (e.g. Wälchli, 2014; Asgari and Schütze, 2017; Liu et al., 2023; Beekhuizen et al., 2024). However, to date, none of these methods has undergone substantial intrinsic evaluation.

Here, I consider the typologically interesting and well-studied domain of spatial relation marking (i.e. prepositions like *in the cup* and *onto the mountain* in English and their translation equivalents; Levinson et al., 2003; Feist, 2008; Viechnicki et al., 2024). This domain can be expected to be challenging for automated extraction procedures due to the great degree of crosslinguistic variation in how languages divide up spatial relational meanings (Levinson et al., 2003; Feist, 2008; Viechnicki et al., 2024), the diversity of means of expression (as affixes or adpositions), and the extensive presence of zero marking of the spatial relation – marginally present in English expressions like *I'm _ home* and *I'm going _ home*, but prevalent in many other languages (Stolz et al., 2014; Haspelmath, 2019).

doculect	example	SRM(s)
English	**On** their heads were something …	on
Indonesian	… dan **di atas** kepala mereka …	di, atas
Quechua	… kansapa umankuna**pi** kurimanta	-pi
German	… und **auf** ihren köpfen wie …	auf
Kilivila	… leikatububulaisi paila kabilia **o**pwanetasi eisikamsi …	o-
Mixtec	… saá ná'a ña kánóo sini ñii ñii …	∅
Nigerian Pidgin	… war wetin dey dia head bi like gold	∅

Table 1: Equivalent Spatial Relation Markers in Rev. 7:9

This paper presents five contributions, with Materials at `https://github.com/dnrb/ruimte`.

- a dataset for evaluating the extraction of spatial relation markers (SRMs) in 18 languages;
- the formulation of two novel tasks for assessing the extraction quality (1) of the SRMs themselves, and (2) of the identification of the *absence* of any SRM;
- a novel model, RUIMTE , for extracting SRMs from a massively parallel corpus;
- an evaluation of it against comparable models;
- a brief demonstration of downstream insight of these results for the typology of spatial relation marking.

2 Data and seed set

For typological coverage, I used a corpus of Bible translations. As the Parallel Bible Corpus (Mayer and Cysouw, 2014) is not publicly available, Bible translations were downloaded through the API of `faithcomesbyhearing.com` (N=1, 367), and from `aboriginalbibles.org.au` (N=14). Given translation availability, only New Testament data were used. Each unique translation (identified with

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 571–585
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

the language's 3-character ISO-939 code followed by a 3-character identifier) was considered one 'doculect'.[1] Under copyright agreements, translations cannot be reproduced but the materials contain a list of doculects.

To the best of my knowledge, there are no languages with spatial relation markers (SRMs) that are not polysemous with non-spatial (temporal and metaphorical/abstract) meanings, and as such no surface form of a (set of) markers can be used as a 'clean' seed to the extraction procedure. Instead, I manually created a seed set of exclusively spatial usages of English prepositions, as follows:

All prepositional phrases with nominal complements and one of the prepositions *in, on, at, to, onto, into, from, out, off* were identified in the World English Bible (ENGWEB) translation using the SpaCy dependency parser (Honnibal and Montani, 2017), and subsequently manually annotated for whether (1) whether they involved a spatial relation, (2) the **dynamicity** of the relation: static ('Locative'), dynamic towards a Ground ('Allative') or dynamic away from a ground ('Ablative'; using the terminology of Haspelmath, 2019), (3) the spatial **relation**, using the finest-grained categories of Levinson et al. (2003), and (4) the **ground type**, or the semantic class of the ground, using distinctions made in Haspelmath (2019) alongside several bottom-up identified categories. Examples are given in Appendix A.

3 Extraction methods

The task at hand is to extract, given a spatial relation marker (SRM) token in one doculect, all and only the translation equivalent SRMs in other doculects. Table 1 exemplifies the challenge: some languages have adpositions (English, German), others affixal SRMs (Kilivila, Quechua). Many doculects use one SRM, but Indonesian uses two, and Mixtec and Nigerian Pidgin use no explicit SRM in this example. In this section, I define several components of a model addressing this extraction problem. The critical components of the pipeline are: (1) morphological segmentation, (2) an alignment/extraction heuristic, and (3) the post-processing of allomorphy and complex SRMs.

3.1 Morphological segmentation

Given that SRMs can be affixal, the extraction procedure should be able to consider affixes as candidate extractions. Some extraction procedures (e.g. Liu et al., 2023) already consider sub-word strings, but others might require the space-bound words to be further segmented into the stems and bound morphemes prior to extraction. Given that for the majority of the 1,381 doculects in the corpus no off-the-shelf morphological segmentation procedure is available, we will have to rely on unsupervised segmentation procedures that can be trained on the parallel Bible corpus itself. Here, I consider three unsupervised models.

MORFESSOR (Virpioja et al., 2013), first, can be trained on wordlists derived from the Bible corpus. Word frequencies affect the likelihood of segmentation in the model – I used the three suggested transformations of word frequency: the **type** model assigns a frequency of 1 to each type, **log-token** the log-transformed token frequency and **token** the token frequency. As MORFESSOR does not provide a distinction between stems and affixes, FLATCAT (Grönroos et al., 2014) was applied to determine the morphological status of the segments.

Second, MORSEL (Lignos, 2010) is a precision-oriented unsupervised procedure based on a best-first heuristic processing the space of possible morphological transforms. Like MORFESSOR, it relies on word lists and frequencies. I used the two predefined parameter settings **Aggressive** and **Conservative**, differing only in their approach to the detection of stem compounds.

Finally, VORM (Beekhuizen, 2025), is an unsupervised model that leverages translations to constrain the search space of morphological transforms and follows the intuition of MORSEL in making a best-first pass through the hypothesized morphological transforms. The minimal number of instances of a morphological transform was set to $N = 10$.

3.2 Extraction procedures

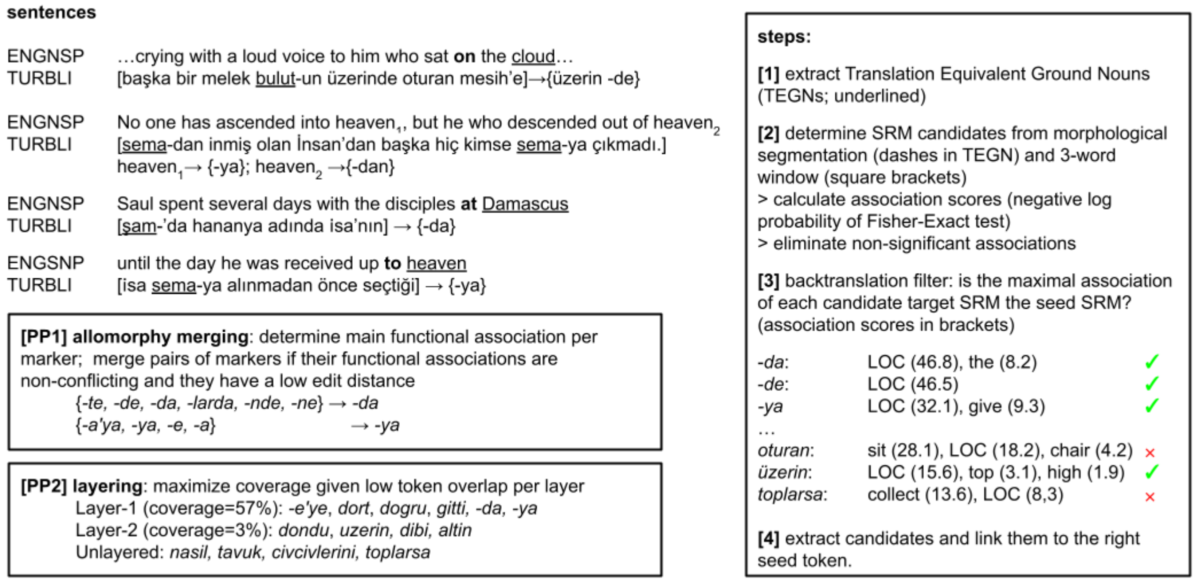
The second component of the pipeline is the extraction procedure itself. I will introduce the novel model, after which I present comparison models.

3.2.1 The RUIMTE extraction model

The Ground nouns and their translations form relatively easily identified anchors of spatial relation marking, with SRMs expected to occur close by. The RUIMTE ('**R**etrieval of **U**nique **I**nstances of **M**arkers of **T**opological **E**ssence') model leverages

---

[1] I adapt the term 'doculect' (Cysouw and Good, 2013) in order to reflect the fact that one 'language' (i.e., an ISO-939 code) may have multiple translations, each representing one 'documented variety', and to stress the somewhat tenuous relation between the documentation of a language through a text with potentially culturally foreign content that is often created (to a large extent) by a non-native speaker.



sentences

ENGNSP …crying with a loud voice to him who sat **on** the cloud…
TURBLI [başka bir melek bulut-un üzerinde oturan mesih'e]→{üzerin -de}

ENGNSP No one has ascended into heaven$_1$, but he who descended out of heaven$_2$
TURBLI [sema-dan inmiş olan İnsan'dan başka hiç kimse sema-ya çıkmadı.]
heaven$_1$→ {-ya}; heaven$_2$ →{-dan}

ENGNSP Saul spent several days with the disciples **at** Damascus
TURBLI [şam-'da hananya adında isa'nın] → {-da}

ENGSNP until the day he was received up **to** heaven
TURBLI [isa sema-ya alınmadan önce seçtiği] → {-ya}

[PP1] allomorphy merging: determine main functional association per marker; merge pairs of markers if their functional associations are non-conflicting and they have a low edit distance
{*-te, -de, -da, -larda, -nde, -ne*} → *-da*
{*-a'ya, -ya, -e, -a*} → *-ya*

[PP2] layering: maximize coverage given low token overlap per layer
Layer-1 (coverage=57%): *-e'ye, dort, dogru, gitti, -da, -ya*
Layer-2 (coverage=3%): *dondu, uzerin, dibi, altin*
Unlayered: *nasil, tavuk, civcivlerini, toplarsa*

steps:

[1] extract Translation Equivalent Ground Nouns (TEGNs; underlined)

[2] determine SRM candidates from morphological segmentation (dashes in TEGN) and 3-word window (square brackets)
> calculate association scores (negative log probability of Fisher-Exact test)
> eliminate non-significant associations

[3] backtranslation filter: is the maximal association of each candidate target SRM the seed SRM? (association scores in brackets)

-da:	LOC (46.8), the (8.2)	✓
-de:	LOC (46.5)	✓
-ya	LOC (32.1), give (9.3)	✓
…		
oturan:	sit (28.1), LOC (18.2), chair (4.2)	×
üzerin:	LOC (15.6), top (3.1), high (1.9)	✓
toplarsa:	collect (13.6), LOC (8,3)	×

[4] extract candidates and link them to the right seed token.

Figure 1: Exemplification of the RUIMTE model: main steps and postprocessing

this fact to identify SRMs, using intuitions similar to the noun-case extraction model of Weissweiler et al. (2022), namely that the overrepresentation inside the window of a target noun is a cue for extraction. Figure 1 provides an example, to follow along with the description of the steps below.

In **Step 1** of the procedure, translations of all Ground nouns are extracted with an adapted forward pass of the Liu et al. (2023) method. This procedure determines, for a seed noun type n, the character string s_{max} in the target language with the strongest statistical association to n, based on their co-occurrence frequency across Bible verses. More precisely, let U be the set of verses containing n, V the set of verses whose translation contains a character string s, and A the set of all utterances for which translations are available. The association between n and s is then defined as the negative log probability of a one-tailed Fisher Exact test over the following 2×2 table:

$$\begin{array}{cc} |U \wedge V| & |U \setminus V| \\ |V \setminus U| & |A \setminus (U \vee V)| \end{array}$$

Two constraints on s_{max} eliminate spurious associations, namely that $|U \wedge V| > 0.10 \times |U|$ and $|U \wedge V| > 0.10 \times |V|$. Next, any space-bound strings in the translations of utterances in U containing s_{max} are extracted as translation-equivalent ground noun tokens (TEGNs) of n, and the utterances in $|U \wedge V|$ are removed from U, A, and V. The procedure is repeated until no more valid candidates can be extracted.

Step 2, next, determines SRM candidates. For each seed item, all target language words in a 3-word window around each of the item's TEGN tokens (including the TEGN token) are retrieved and morphologically segmented using a morphological model. Each segment, combined with its position (whether it occurs before the TEGN, after it, or is part of it), forms a candidate SRM. Keeping track of the position is informative for the model, as adpositions typically occur on one side of the head noun. Given the extracted candidate SRMs, we then calculate the association of each candidate SRM type to the full set or a subset of the seed items. The association score of a candidate SRM is defined as the maximal negative log probability of a one-tailed Fisher Exact test applied to the following 2×2 table, maximizing U_{x} from the set of U and any U_{f} used:

$$\begin{array}{cc} |U_{\text{x}} \wedge C| & |N \wedge C| \\ |U_{\text{x}} \wedge D| & |N \wedge D| \end{array}$$

where:

- U is the set of verses containing a seed item,
- U_{f} is the set of verses containing seed items with a specific annotated feature value or combination of feature-values,
- N is the set of verses containing any seed noun in a non-prepositional (and therefore almost certainly non-locative) context,
- C is the set of verses whose translation contains the candidate SRM,
- D is the set of verses whose translation does not contain the candidate SRM.

We can define U_{f} variably. First, the spatial relation itself ('rel') can be used, defining three nodes in the taxonomy of Levinson et al. (2003): $U_{\text{containment}}$ (annotated instances of 'IN-2D' and 'IN-3D'), U_{support} (instances of 'ON', 'ON-TOP', and 'ATTACHMENT'), and $U_{\text{colocation}}$ (instances of 'COLOCATION'). This allows us to find associations with SRMs that are exclusively used for one but not the other relation, effectively introducing a prior from the typological literature on what languages frequently do. Another such prior comes from the dynamicity ('dtc') of the spatial relation, defining three seed sets U_{static}, U_{goal}, and U_{source} (cf. Haspelmath, 2019). A third option is to combine them ('dtc&rel'). Fourth, we can use no U_{f}s ('all'), and finally, we can use the English prepositions ('prp') as an easily accessible seed type. We call these five settings the **seed types**.

SRMs with scores $< -\log 1^{e-6}$ are omitted. The resulting set still contains spurious markers. To remove these, **Step 3** implements a backtranslation filter that eliminates candidate target SRMs that are more strongly associated with frequently co-occurring context words of the seed SRMs (e.g. *go* in the context of *go into their house*). To do so, the forward-pass of the Liu et al. model is applied to each seed language word occurring with a frequency ≥ 10 in a 3-word window around any seed item noun (excluding the prepositions). This procedure gives us a statistical association score between the seed language word and the candidate SRMs that is comparable to the association score retrieved in the previous step. If a context word has a stronger association with a candidate SRM than the seed, that candidate SRM is deleted, as it is more likely a translation of a frequent context word of a spatial relation marker in the seed language.

Finally, **Step 4** takes one seed item token at a time, and finds the TEGN token whose candidate SRMs are most strongly associated with the U_{f} of the seed item. This ensures the correct extraction for sentences with multiple seed SRMs (e.g. the second sentence in Figure 1. A further constraint imposed is that only the highest-ranked affix is extracted, as spatial relations are not expected to be expressed through multiple affixes the same noun.

3.2.2 CONCEPTUALIZER

A first comparable model is the procedure of Liu et al. (2023), CONCEPTUALIZER, which, given a set of seed utterances in which a particular marker occurs, iteratively finds the substrings in a target language that are statistically most strongly associated with that set of seed utterances. As with the RUIMTE model, we can use various U_{f} independently as seeds and concatenate the results. In particular, I define the same five seed types as for RUIMTE. I further used the parameter settings cited in the paper ('original') as well as a loosening of some of the stricter settings ('bare': allowing up to 30 iterations, and only considering target-language substrings occurring in $\geq 0.1\%$ of U_{f}).

3.2.3 Alignment-based baseline models

Both the CONCEPTUALIZER and RUIMTE models are designed for the task of marker extraction in massively parallel corpora. As informed baseline models, I consider models based on unsupervised word alignment. Word alignment models allow us to create a bigraph between the seed language utterances and the (morphologically segmented) target language utterances. Extracting all aligned segments (words and affixes) to the seed SRMs can be expected to perform reasonably well as an extraction procedure. I apply two alignment procedures, EFLOMAL (Östling and Tiedemann, 2016) and FASTALIGN (Dyer et al., 2013), to the bitext between the seed language and each target language, retrieving any alignment to seed item prepositions.

As in previous models, we can vary the seed types, replacing the tokens of seed item prepositions in the bitext by a string identifying their feature representation. Similarly, we vary the morphological segmentation procedure used to preprocess the target language. Two final model parameters for aligner-based extraction procedures are symmetrization heuristics and a frequency filter (as proposed by Liu et al., 2023). Alongside the seed-to-target alignments ('fwd') and target-to-seed alignments ('rev'), we can consider their union and intersection, as well as three symmetrization heuristics that add and remove further alignments, namely 'diag-grow', 'diag-grow-final' and 'diag-grow-final-and' (implemented in `atools` Dyer et al., 2013). For the frequency filter, I consider no filter ('> 0'), an expected frequency of an extracted SRM given a seed type of more than 1 ('> 1'), or of more than 1% of the size of the set of seed types it was aligned to ('$> 1\%$').

3.3 Postprocessing

Two postprocessing steps were found to improve extraction quality on the development doculects (see Section 4.2). They are modular steps that can be applied to the outputs of any of the extraction procedures defined above. First, many languages display allopmorphy, either in their adpositions (German *in*, *im*, *ins*, 'in') or affixes (Turkish *-da/de/ta/te* 'locative case'). I define a simple heuristic to automatically **merge** these:

An agenda is initialized with all extracted SRMs per doculect ranked by frequency. Starting with the most frequent SRM, all remaining SRMs in the agenda are considered in turn, merging them with the current target SRM and its already-merged other SRMs if either they are formally near-identical (i.e., string identity after stripping diacritics and ignoring whether it is an affix vs. an adposition and preposition vs. postposition) or if they are formally possible allomorphs (i.e., having a low string edit distance) *and* have functionally similar patterns. Such allomorphs are then removed from the ranked list and the next marker is considered. At the end, all instances of allomorphs are replaced by the most frequent allomorph.

For the 'functional similarity' constraint, we consider per SRM (or cluster of already merged SRMs) which feature-value combination (from among whichever features are used in the seed type for that language; defaulting to both dynamicity and relation if the seed type was 'all' or 'prp') leads to the greatest Information Gain in classifying whether a seed item is translated with that SRM (of: one SRM from that SRM cluster) or not. If the values overlap for at least one feature (i.e., one SRM has 'containment' and 'colocation' for 'relation' as its feature-values optimally discriminating it, while the other has just 'containment') and do not contrast (i.e., the same two SRMs do not have non-overlapping values for the other feature, 'dynamicity' – e.g., 'static' for the first SRM and 'goal' for the latter), the two SRMs are considered functionally similar.

Second, complex adpositions were identified through **layering**: first, the set of SRMs that (1) minimally overlap with each other w.r.t. the TEGNs they occur with and (2) jointly cover the largest set of tokens is extracted as Layer-1, after which the procedure is repeated on the remaining markers to find a possible Layer-2. Affixal markers are eliminated from the second layer, as (obligatory) locative case marking should take place on the layer with the greatest coverage, and any affixes found on Layer 2 in the development doculects were false positives. Any unlayered markers are eliminated. Note that this step may aid in quality but was initially conceived for analytic purposes. Many languages have complex adpositions (e.g., Indonesian in Table 1) and being able to determine which adpositions form a paradigm is an important step in characterizing SRM systems.

4 Experimental set-up

4.1 Preprocessing

As not all Bibles come in Roman script, and as several morphological models depend on ASCII encoding, I transduced the text with an isomorphic mapping into ASCII for each doculect that at least partially used Roman characters (e.g. Vietnamese), and applied `unidecode` to transliterate the unicode characters into ASCII in other cases (e.g. Persian).

4.2 Annotation

Comparably little structured evaluation on the extraction of translation equivalent linguistic elements from massively parallel corpora has been carried out. Here, I introduce a dataset of 180 seed items for which I manually extracted (using grammars and dictionaries, alongside Google Translate) the SRMs in each of a typologically diverse set of 18 doculects. The 180 items were randomly sampled by selecting 60 seed items of each dynamicity value (locative, allative, ablative).

Zero coding was decided as follows: if no translation of the ground noun was found, the category 'noTEGN' was assigned. If the verbal predicate entailed the relation between the subject and the ground noun (like *enter* or *ascend*), or if some other non-spatial relation (commitatives like *with* and partitives like *of* are common) was marked, 'nonSpatialRelation' was assigned. Finally, if there was a translation-equivalent ground noun and a spatial relation to some verbal or nominal head that did not entail the relation, but no overt marking, 'trueZero' was assigned.

Table 5 in Appendix C presents the doculects, along with their top-3 markers and their proportion of zeros (both nonSpatialRelation and trueZero). I split the data into a development set of the top 9 doculects and a test set of the last 9, further only considering the even items of the development set when developing and tuning the various models.

4.3 Evaluation procedure

With these data, we can define an evaluation procedure. I formulate the extraction of SRMs and the correct identification of zero marking as separate tasks, given the interest in the typological literature in zero marking.

SRM extraction is a multi-class classification problem (multiple SRNs may simultaneously apply). Moreover, the strings in the extracted data may not match the annotated data exactly, due to variation in extraction and allomorphy. The proposed metric of evaluation is able to work with these constraints. First, given a set of extracted SRMs $E = \{e_1, e_2, \ldots, e_n\}$ and a set of annotated SRMs $A = \{a_1, a_2, \ldots, a_n\}$, each defining a set of seed tokens $U(x)$, where x is an SRM from A or E, we find the injective mapping M between A and E that maximizes the model's extraction accuracy, by maximizing the sum of the cardinalities of the intersections of $U(a_i)$ and $U(M(a_i))$, or: the seed items in which a_i occurs resp. the seed items in which some e_j in E, mapped to by $M(a_i)$ occurs.

With this mapping, we can determine, for each a_i mapping to some $e_j = M(a_i)$, how many True Positives ($|U(a_i) \cap U(e_j)|$), False Positives ($|U(e_j) \setminus U(a_i)|$) and False Negatives ($|U(a_i) \setminus U(e_j)|$) it has, and sum those across all $a_i \rightarrow M(a_i)$ mappings. The token count of any unmapped annotated items is added to the False Negatives, while unmapped extracted items are added to the False Positives. This allows us to define, for each doculect, the Precision, Recall, and F_1-score.

For the **evaluation of zero extraction** Precision, Recall, and F_1-score were defined as usual for a binary categorization problem. Predictions of zeros were compared against annotated cases, and counted as correct if 'trueZero' or 'nonSpatialRelation' was annotated; instances of 'noTEGN' were left out of consideration.

5 Results

5.1 Basic pipelines

I first consider the basic pipelines without postprocessing. The two alignment procedures have further hyperparameters like alignment symmetrization and frequency filtering that multiply out to a large number (1730) of unique models. The Materials present a full spreadsheet with performance per doculect for each unique model.

To narrow down the scope I consider only model components that perform substantially better than others; Figure 4 in Appendix D presents these comparisons. I only keep models that use (if applicable) the 'forward' symmetrization heuristic (which provides a good balance on the performance on both SRM extraction and zero extraction), and a frequency filter of $\geq 1\%$, which performs better than the other two filters on both tasks. The MORFESSOR 'token' and 'logtoken' models never performed as well as the 'type' model and were eliminated from consideration; similarly, using no morphological segmentation performed consistently worse. Between the alignment models, FASTALIGN had consistently lower scores than EFLOMAL and was not considered further.

Table 2 presents the performance of the remaining models on both SRM extraction (left columns) and zero extraction (right columns), reporting F_1-scores averaged over 18 doculects or over the 5 'zero doculects', i.e., doculects with $\geq 10\%$ zeros. The best-performing model on both tasks ($F_1 = 66.4$ for SRM extraction and $F_1 = 77.4$ for zero extraction) is the RUIMTE model using MORSEL-Aggressive. Notably, the optimal seed differs between the tasks: English prepositions ('prp') are optimal for SRM extraction while seeding on any spatial relation ('all') works better for zero extraction, due to the lower numbers of spurious markers extracted in the latter case. The VORM morphological model performs almost as well on both tasks at $F_1 = 66.1$ resp. $F_1 = 77.2$.

5.2 Effects of post-processing

Next, I assess the effect of the **postprocessing steps**, merging and layering, on the extraction of SRMs and zeros, considering the same models as in the last section. Table 3 presents the results, narrowing the seed types down to only 'all' and 'prp' as no optimal performances were found among the semantic-feature based seed types. We find that in particular the merging step has a positive influence on extraction. The best SRM-extraction models, i.e., VORM with a 'dtc' or 'prp' seed and using the merging step ($F_1 = 70.6$ resp. $F_1 = 69.6$), outperform the counterpart without merging (the 'basic' model) ($F_1 = 66.1$ resp. $F_1 = 64.2$) by > 4 points. Layering, however, does not appear to have the anticipated impact, with scores similar or somewhat lower than the basic model.

5.3 Performance analysis

What models by and large get right, are the most frequent SRMs per doculect, for which the statistical association is unequivocally strong. The remainder of this section considers where variation between the models was found and what the varying performance could be attributed to.

Looking at **individual doculects** might lead to insight in the variation. Figure 2 presents the Preci-

aligner; morphological model	SRM extraction					Zero extraction				
	all	dtc	dtc&rel	prp	rel	all	dtc	dtc&rel	prp	rel
Conceptualizer-bare	17.4	17.2	16.5	16.7	17.3	33.3	36.7	33.7	30.7	32.1
Conceptualizer-original	10.2	14.9	17.3	17.6	12.8	44.2	43.2	44.2	41.3	44.7
eflomal; Morfessor-type	48.7	53.6	53.5	49.7	49.3	48.9	48.0	48.9	56.4	48.2
eflomal; Morsel-agg.	50.9	55.5	55.3	52.2	51.6	45.5	49.3	46.3	51.8	49.7
eflomal; Morsel-con.	50.1	54.3	54.4	50.4	50.6	45.1	47.6	49.7	53.6	47.4
eflomal; Vorm	49.7	55.2	55.4	50.5	51.5	42.3	44.4	45.2	50.7	45.8
Ruimte; Morfessor-type	57.8	60.2	60.1	60.6	59.1	**76.9**	69.4	67.4	70.4	74.4
Ruimte; Morsel-agg.	62.5	65.0	65.2	**66.4**	62.6	**77.4**	65.0	64.1	66.8	72.4
Ruimte; Morsel-con.	61.6	64.1	64.4	**65.8**	63.0	76.3	65.6	64.9	67.6	72.5
Ruimte; Vorm	63.6	64.2	64.3	**66.1**	63.2	**77.2**	67.2	65.2	70.4	73.0

Table 2: **Basic models.** Mean F_1-scores for SRM extraction (left) and zero extraction (right) per combination of aligner and morphological model (rows) and seed type (columns) for the best-performing model components.

morphological model; seed	SRM extraction				Zero extraction			
	basic	merge	layer	both	basic	merge	layer	both
Morfessor-type; seed = all	57.8	58.4	55.1	55.7	**76.9**	**76.9**	75.8	75.8
Morfessor-type; seed = prp	60.6	61.5	59.8	59.3	70.4	70.4	71.1	70.1
Morsel-aggressive; seed = all	62.5	64.7	58.4	60.6	**77.4**	**77.4**	75.9	75.9
Morsel-aggressive; seed = prp	66.4	69.4	65.2	65.5	66.8	66.8	70.3	66.6
Morsel-conservative; seed = all	61.6	63.9	58.1	60.4	76.3	76.3	75.7	75.7
Morsel-conservative; seed = prp	65.8	68.1	64.9	64.9	67.6	67.6	71.0	67.4
Vorm; seed = all	63.6	65.0	61.8	63.3	**77.2**	**77.2**	**76.6**	**76.6**
Vorm; seed = prp	66.1	**70.6**	65.7	69.0	70.4	70.4	71.0	71.6

Table 3: **Postprocessing steps.** Mean F_1-scores for SRM extraction (left) and zero extraction (right) per combination of aligner and morphological model (rows) and combination of postprocessing steps (columns).

sion and Recall, per doculect, for a select number of models: the best-performing conceptualizer, fastalign, and eflomal models (prior to postprocessing) alongside the top-7 best performing Ruimte models. For SRM extraction, the results for 5 illustrative doculects are shown in Figure 2 with the full set in Figure 5 in Appendix E. There are two types of doculects: those where the two alignment-based models (eflomal and fastalign) perform more like Ruimte and those where they perform more like conceptualizer.

This distinction seems to line up with the morphological expression of the SRMs: affixal SRMs, like in Cree and Kilivila, are not as well extracted with alignment-based extraction as with Ruimte, while for doculects with primarily adpositional SRMs, like German and Vietnamese, the differences between alignment-based models and the Ruimte are smaller. It is possible that he increased space of possible alignments for morphologically complex doculects decreases the alignment quality. This explanation is supported by the finding that the most-frequent SRM in Cree, *-ihk*, is in alignment-based models aligned to the seed item in only a subset of the cases for which it is annotated, suppressing the Recall. Moreover, 20+ unique non-spatial markers are (spuriously) aligned to the seed items, suppressing the Precision.

For zero marking (See Figure 6 in App. E), we notice, first, that the Ruimte-based models achieve near-perfect Recall across the 5 zero-marking doculects, meaning that most annotated zeros are indeed extracted as zeros. The challenge, however, is Precision, i.e. : the model detecting a zero SRM where there is a non-zero SRM present. This effect is particularly strong for Bambara and Somali. For both, Precision is poor due to the high degree of polysemy of the SRMs (as noun classifiers in the former and highly general verbal particles in the latter) which leads to their spurious

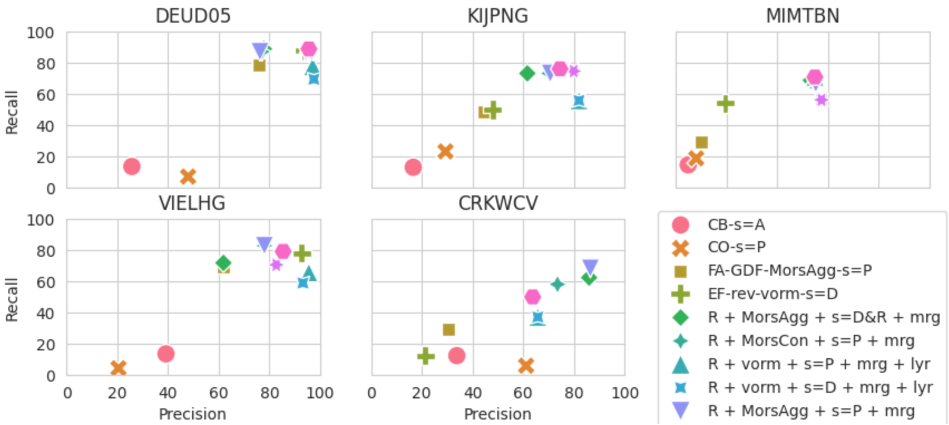


Figure 2: Precision and Recall on SRM extraction in 9 select models (see text) for 5 doculects.

presence in windows where the true marker is zero.

Considering doculects that show low performance, we find Mixtec, Finnish, and Malayalam, all with $F_1 < 70$ in any model. Weak performance for Mixtec can be attributed to the doculect's frequent use of zero marking. Lower-frequency prepositions (*sata, nandoso, ndaa*) did not reach the significance threshold of Step 2, and several spurious markers were instead extracted where zeros should be predicted. For Finnish and Malayalam, the morphological complexity might lead to an increased number of missegmented cases. Further, allomorphy (for Finnish) presents challenges. While allomorphy is generally resolved correctly, some non-allomorphs that are formally and functionally similar, such as *-lla/-llä* 'adessive' and *-lle* 'allative', tend to be merged erroneously. For Malayalam, a final challenge consists of its stacked locative cases, which the model cannot extract given the 'one-affix' constraint.

When considering the doculects for which the model performs well ($F_1 > 80$ on the top model, i.e., German, Indonesian, Dutch, Vietnamese, and Bambara), we find that the most common SRMs are correctly identified in a vast majority of cases (often well over 90%). Three sources of remaining errors can be identified. First, there are instances where the target-language SRM falls outside of the 3-word window around the Translation-Equivalent Seed Noun and is thus not extracted. Second, the 3-word window may contain spurious, but more strongly associated markers (e.g., when two adpositional phrases occur closeby to each other). Third, we find cases of failure to extract SRMs when they are either of low frequency or have a more frequent homonymous meaning (e.g. Dutch *te* 'to, at' is also the infinitive marker, like English *to*). In both scenarios, the association score with the seed SRMs in Step 2 is suppressed, leading to non-extraction.

5.4 Discussion

Reasonably good performance was achieved on the tasks of extracting SRMs and zeros across 18 doculects. Components of the best performing models included the novel Ruimte extraction procedure and Precision-oriented morphological segmentation (Morsel and the novel Vorm model), as well as using the English prepositions as seed items. The latter was particularly surprising, given that most doculects do not encode spatial relations exactly along the lines of English SRMs.

Among the extraction procedures, CONCEPTUALIZER performed remarkably poorly, in contrast with its compelling performance as reported by Liu et al. (2023), as well as its reliability as a component in the Ruimte model for extracting TEGNs. This suggests that CONCEPTUALIZER works well for lexical, open-class items, but not so much for more closed-class ones. Nonetheless, its components and general intuitions (regarding the use of co-occurrence statistics) translate well to this domain and form the engine of the Ruimte model. Furthermore, Precision-oriented segmentation procedures such as Morsel and Vorm outperform the Morfessor baseline substantially for SRM extraction, suggesting that oversegmentation is harmful to the extraction, likely because it introduces noisy candidate SRMs.

6 Applications

To study potential use for typology, I briefly explore the best-performing model, Ruimte +Vorm,

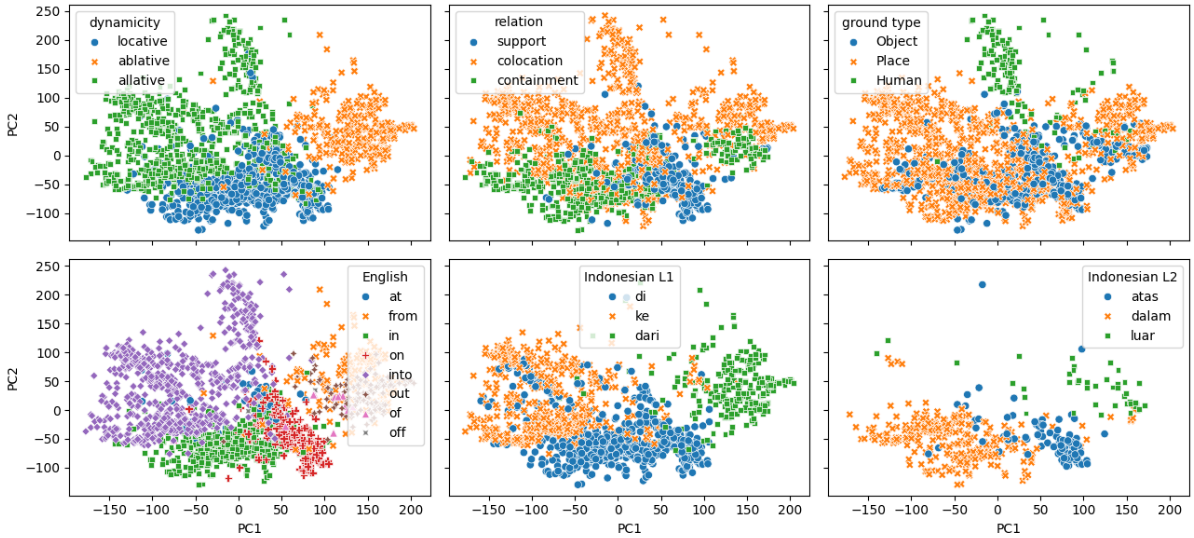


Figure 3: Exponential PCA space

with 'prp' seed type and merging. Markers in 1161 doculects were extracted. All seed items for which $> 66.7\%$ of doculects had no TEGN and subsequently all doculects for which $> 66.7\%$ of the seed items had no TEGN were eliminated, leaving a 1966×912 (tokens by doculects) table.

These extractions allow us to compare doculects on a token level. The main axes of crosslinguistic variation can be visualized with dimensionality reduction, here using Exponential PCA (Collins et al., 2001). Figure 3 shows 6 different colour codings of the same space, with each marker representing one seed item: the top three panels display the three annotated features – dynamicity, relation, and ground type. Notably, the values of all three features are reasonably separable in the 2-dimensional PCA space, meaning that there are doculects drawing SRM contrasts on the basis of each of these features. Indeed, we see some of those patterns play out in the two doculects in the right panels: while the English prepositions align with all three features, Indonesian neatly splits out dynamicity and relation between two sets (layers) of prepositions.

Second, these data allow us to assess typological claims concerning zero marking. Based on a survey of grammars, Stolz et al. (2014) conclude that zero marking is crosslinguistically more common in locatives and allatives than in ablatives. My data supports this finding. Per item, the proportion of zero-extractions across doculects was calculated. Aggregating those proportions, we find that the median locative item has zero marking in 34% (IQR: 27-41%) of doculects, the median allative in 31% (IQR: 25-38%), and the median ablative in 25% (IQR: 20-34%), with the differences between each pair significant (independent t-test; $P < .001$). It further suggests that locatives may be more commonly zero-marked than the allatives. While a fuller considerations is beyond the scope of this paper, these initial explorations demonstrate the potential for the typology of SRMs.

7 Conclusion

This paper introduces a novel procedure for extracting Spatial Relation Markers (SRMs) across languages, and evaluates it on a novel dataset. The method is shown to have good potential for answering typological questions about SRMs.

Interestingly, the components of my pipeline, as well as others working on similar tasks (Wälchli, 2014; Weissweiler et al., 2022; Liu et al., 2023), are mostly using simple statistics and best-first extraction procedures, rather than more complex Machine Learning techniques (e.g., seq2seq models) using global optimization. This is not a coincidence: the size of the data, combined with the nature of the hypothesis space (the number of SRMs – 0, 1, or 2; affixes vs. adpositions), makes extraction procedures that explicitly constrain the search space on priorly motivated grounds more successful.

This paper intends to contribute to the growing body of work on computational semantic typology with this paper, by introducing more rigorous evaluation techniques, providing annotated seed and evaluation data, and suggesting novel ways that spatial relation markers can be extracted.

Limitations

The work presented here was run on a corpus of Bible translations. The question whether the same methodology works well on other parallel corpora in different genres and dealing with different topics has not been positively answered, thus potentially constituting a limitation of the method that future work would have to settle.

Acknowledgments

This work has been over a decade in the making since its first conception, with the publication of Liu et al. (2023) and the subsequent invitation to talk about it at the CogSci 2023 symposium *Space in Context* (Grigoroglou et al., 2023) forming a major impetus to its completion. I am indebted to Suzanne Stevenson and Bernhard Wälchli for intermittent discussion over the years, and to Kit Donohue for contributing to the annotations of the seed set. Any inaccuracies and imperfections are, however, solely mine. I would further like to express gratitude to the three anonymous CoNLL reviewers who each provided helpful feedback.

References

Ehsaneddin Asgari and Hinrich Schütze. 2017. Past, present, future: A computational investigation of the typology of tense in 1000 languages. *arXiv preprint arXiv:1704.08914*.

Barend Beekhuizen. 2025. VORM: Translations and a constrained hypothesis space support unsupervised morphological segmentation across languages. In *29th Conference on Computational Natural Language Learning (CoNLL 2025)*.

Barend Beekhuizen, Maya Blumenthal, Lee Jiang, Anna Pyrtchenkov, and Jana Savevska. 2024. Truth be told: a corpus-based study of the cross-linguistic colexification of representational and (inter) subjective meanings. *Corpus Linguistics and Linguistic Theory*, 20(2):433–459.

Michael Collins, Sanjoy Dasgupta, and Robert E Schapire. 2001. A generalization of principal components analysis to the exponential family. In *Advances in Neural Information Processing Systems*, volume 14.

Michael Cysouw and Jeff Good. 2013. Languoid, doculect and glossonym: Formalizing the notion 'language'. *Language Documentation & Conservation*, 7.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Ramy Eskander, Smaranda Muresan, and Michael Collins. 2020. Unsupervised cross-lingual part-of-speech tagging for truly low-resource scenarios. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4820–4831, Online. Association for Computational Linguistics.

Michele I Feist. 2008. Space between languages. *Cognitive science*, 32(7):1177–1199.

Myrto Grigoroglou, Barbara Landau, Anna Papafragou, Ercenur Ünal, Kevser Kırbaşoğlu, Dilay Karadoller, Beyza Sumer, Asli Ozyurek, Barend Beekhuizen, Kenny R Coventry, Piotr J. Barc, Lucy-Amber Roberts, and Harmen Gudde. 2023. Space in context: Communicative factors shape spatial language. In *Proceedings of the Annual Meeting of the Cognitive Science Society*.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185.

Martin Haspelmath. 2019. Differential place marking and differential object marking. *STUF-Language Typology and Universals*, 72(3):313–334.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Jane Klavan, Maarja-Liisa Pilvik, and Kristel Uiboaed. 2015. The use of multivariate statistical classification models for predicting constructional choice in spoken, non-standard varieties of estonian. *SKY Journal of Linguistics*, 28.

Stephen Levinson, Sérgio Meira, The Language, and Cognition Group. 2003. 'Natural concepts' in the spatial topological domain-adpositional meanings in crosslinguistic perspective: An exercise in semantic typology. *Language*, pages 485–516.

Natalia Levshina. 2016. Why we need a token-based typology: A case study of analytic and lexical causatives in fifteen European languages. *Folia Linguistica*, 50(2):507–542.

Constantine Lignos. 2010. Learning from unseen data. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 35–38, Helsinki, Finland. Aalto University School of Science and Technology.

Yihong Liu, Haotian Ye, Leonie Weissweiler, Philipp Wicke, Renhao Pei, Robert Zangenfeind, and Hinrich Schütze. 2023. A crosslingual investigation of conceptualization in 1335 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12969–13000, Toronto, Canada. Association for Computational Linguistics.

Thomas Mayer and Michael Cysouw. 2012. Language comparison through sparse multilingual word alignment. In *Proceedings of the EACL 2012 Joint Workshop of LINGVIS & UNCLH*, pages 54–62.

Thomas Mayer and Michael Cysouw. 2014. Creating a massively parallel Bible corpus. *Oceania*, 135(273):40.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Robert Östling and Jörg Tiedemann. 2016. Efficient word alignment with Markov Chain Monte Carlo. *Prague Bulletin of Mathematical Linguistics*, 106:125–146.

Patrick Schone and Dan Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Thomas Stolz, Sander Lestrade, and Christel Stolz. 2014. *The crosslinguistics of zero-marking of spatial relations*, volume 15. Walter de Gruyter GmbH & Co KG.

Thomas Stolz, Nataliya Levkovych, and Aina Urdze. 2017. When zero is just enough… in support of a special toponymic grammar in Maltese. *Folia Linguistica*, 51:453–482.

Willy Van Langendonck. 2007. *Theory and typology of proper names*. Mouton de Gruyter.

Peter Viechnicki, Kevin Duh, Anthony Kostacos, and Barbara Landau. 2024. Large-scale bitext corpora provide new evidence for cognitive representations of spatial terms. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1089–1099.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Bernhard Wälchli. 2014. Algorithmic typology and going from known to similar unknown categories within and across languages. In Benedikt Szmrecsanyi and Bernhard Wälchli, editors, *Aggregating Dialectology, Typology, and Register Analysis: Linguistic Variation in Text and Speech*, pages 355–393. Walter de Gruyter.

Leonie Weissweiler, Valentin Hofmann, Masoud Jalili Sabet, and Hinrich Schuetze. 2022. CaMEL: Case Marker Extraction without Labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5506–5516, Dublin, Ireland. Association for Computational Linguistics.

A Seed annotation methods

The basis of the features used throughout this paper involves a token-level manual annotation of several dimensions of the spatial relation. While manual (semantic) annotation is admittedly prone to challengeable decisions and reductively frames the spatial relations as mutually exclusive, the paper hopefully demonstrates that it leads to useful and interpretable results.

The decision procedure for annotating the 'Relation' feature was based on the finest-grained foci (relations) of Levinson et al. (2003):

- 'Containment' was assigned if the boundaries of the Ground could be conceptualized as restricting the movement of the Figure in two ('In-2D'; in a boat, into his ear) or three ('In-3D'; in his house, into the prison) dimensions;
- 'Support' was assigned if either the Figure was higher than the Ground and not touching it ('Over'; [hang s/t] above his head) or if the surface contact with the Ground could be conceptualized as restricting the movement of the Figure – specifically, the relation annotated was:
  - 'On-Top' if contact was with the highest vertical region of the Ground (on the housetop, [put s/t] on his head);
  - 'Attachment' if contact was with a non-vertical region of the Ground and was maintained mechanically through an external source of Force (pinning, adhesion, clamping; on the stake; [sew s/t] on a garment),
  - 'On' otherwise (on the throne, [pour s/t] on his face)
- 'Under' was annotated if English uses under (under his feet, [take s/o] under her wing). Being a very small category, Under was excluded from the analysis.
- the residual category 'At3' breaks down into 'Near' relations, assigned whenever English uses near, at or by and no Figure-Ground contact is entailed (e.g., at the fire, [let s/o down] by the wall), and 'Colocation' otherwise. Notably, Colocation contains all cases where the Ground denotes a 'place', a (named) symbolically defined region (at Jerusalem, in Judea, to the country, in heaven), following Van Langendonck (2007) in considering their dimensionalities as irrelevant.

For the Dynamicity feature, 'Allative' was assigned whenever a (caused) motion predicate was present with a preposition marking a Ground that is a Goal of the motion predicate (*to, in(to), on(to)*), 'Ablative' if there was a (caused) motion predicate with a Source-marking preposition (*from*, *out of*, *off of*), and 'Static' otherwise.

For types of Grounds ('Ground Type'), I developed a categorization schema based on observations that Places, Named Places, and Human Grounds are occasionally marked differently from regular object-denoting Grounds (Stolz et al., 2017; Haspelmath, 2019), the existence of aquatic adpositions (Levinson et al., 2003), and that the mobility of the Ground affects the lexical choice (Klavan et al., 2015), as well as a bottom-up categorization of prevalent ground types in the corpus, distinguishing:

- Places: a region that is not easily conceptualized as a 'thing' but rather as inherently a 'location' of something else (to the place, in heaven), including Toponyms (in Asia, to Mount Sinai) and Buildings – an Object with unique relevance to humans as shelter, dwelling (in the temple, into the house);
- Object: a bound, countable physical, natural or artificial, entity (to the ship, in his hand);
- Human (bring him to the high priest)

B The intuition of the VORM segmentation model

The VORM model ('**V**ertaling **O**ndersteunt **R**edelijke **M**orfologie'; Dutch for 'Translations support reasonable morphology') is an unsupervised morphological segmentation procedure. Here, I present the intuition briefly; for a complete treatment see Beekhuizen (2025). Like MORSEL, VORM first finds recurrent character string transformations between pairs of words and makes a best-first pass through the word list to obtain derivations based on such transformations. However, only those pairs of words are inspected for the presence of potential transformations that are translation equivalents of the same word in a reference language (here: the seed translation ENGWEB). Distributional semantic information has long been used to bootstrap

doculect	Turkish	Finnish
word	*sofradakiler*	*polveutuu*
meaning	'them at the table'	'descends'
gold	sofra -da -ki -ler	polvi -ua -tuu
MORFESSOR-type	sofrada -kiler	polveutu -u
MORFESSOR-logtok.	sofrada -kiler	polve -utuu
MORFESSOR-token	sofrada -kiler	polveutuu
MORSEL-aggressive	sofradakiler	polveutu -u
MORSEL-cons.	sofradakiler	polveutu -u
VORM	sofra -da -ki -ler	polvi -i/ea -a/utuu

Table 4: Examples of the morphological models

morphological segmentation (Schone and Jurafsky, 2000; Narasimhan et al., 2015), but the proposal here is that translation is similarly a strong signal to constrain the unsupervised learning of morphological segmentation, as has been argued for other tasks, like PoS tagging (Eskander et al., 2020).

While this initial step provides a high-precision inventory of (sequences of) morphological transformations, many morphologically related words do not map onto the same translation equivalent in other languages and are as such not yet linked to each other. In a second step, all possible derivations of all words are generated, on the basis of the set of transformation sequences found in the first step. An agenda with all words is initialized, after which a best-first procedure finds the stem that has the largest morphological family size (i.e., occurs in the candidate derivations of the most words). The modeled words are removed from the agenda and the procedure is repeated until the agenda is empty.

Table 4 presents examples for the morphological extraction procedure applied to two words from Turkish and Finnish and compares it to the other models.

C Information on the annotated doculects

See Table 5.

D Model component comparison

See Figure 4 for an aggregated comparison of the model components.

E Doculect-level performance analysis

Figures 5 and 6 present the Precision and Recall per doculect, for all doculects on both tasks.

name (iso); affiliation	macroarea	most common three markers	% zeros
Plains Cree (CRKWCV)	Algic, North-America	*-ihk* (141) *oci* (39) *isi* (10)	7.3
German (DEUD05)	Indo-European, Eurasia	*in* (51) *aus* (34) *auf* (32)	1.2
Finnish (FINELC)	Finno-Ugric, Eurasia	*-an* (51) *-sta* (46) *-ssa* (24)	6.1
Indonesian (INDNTV)	Austronesian, Oceania	*di* (61) *dari* (52) *ke* (40)	3.9
Kilivila (KIJPNG)	Austronesian, Oceania	*o-* (67) *wa* (33) *metoya* (29)	28.5
Mixtec (MIMTBN)	Mixe-Zoque, North-Am.	*noo* (33) *ini* (7) *ndaa* (2)	62.0
Nigerian Pidgin (PCMTSC)	Creole, Africa	*for* (47), *from* (47) *inside* (7)	26.2
Somali (SOMSIM)	Afro-Asiatic, Africa	*ku* (59) *ka* (52) *soo* (35)	17.9
Turkish (TURBLI)	Turkic, Eurasia	*-ya* (57) *-dan* (53) *-da* (51)	3.4
Bambara (BAMLSB)	Mande, Africa	*la* (63) *bo* (41) *kono* (26)	23.2
Basque (EUSNLT)	isolate, Eurasia	*-an* (61) *-tik* (58) *-ra* (49)	1.7
Malayalam (MALNIB)	Dravidian, Eurasia	*-il* (101) *-kku* (34) *ninnu* (32)	10.1
Dutch (NLDDSV)	Indo-European, Eurasia	*in* (43) *uit* (34) *op* (31)	0.0
Persian (PESTPV)	Indo-European, Eurasia	*az* (49) *bah* (40) *dar* (38)	9.0
San Martín Quechua (QVSTBL)	Quechuan, South-Am.	*-pi* (77), *-manta* (41), *-man* (29)	0.0
Rundi (RUNBSB)	Niger-Congo, Africa	*mu* (83) *i* (35) *ku* (25)	9.5
Spanish (SPABDA)	Indo-European, Eurasia	*en* (59) *de* (44) *a* (34)	6.7
Vietnamese (VIELHG)	Austroasiatic, Eurasia	*tu* (35) *trên* (19) *o* (18)	9.0

Table 5: The 18 annotated doculects. The top nine are development doculects; bottom nine test doculects

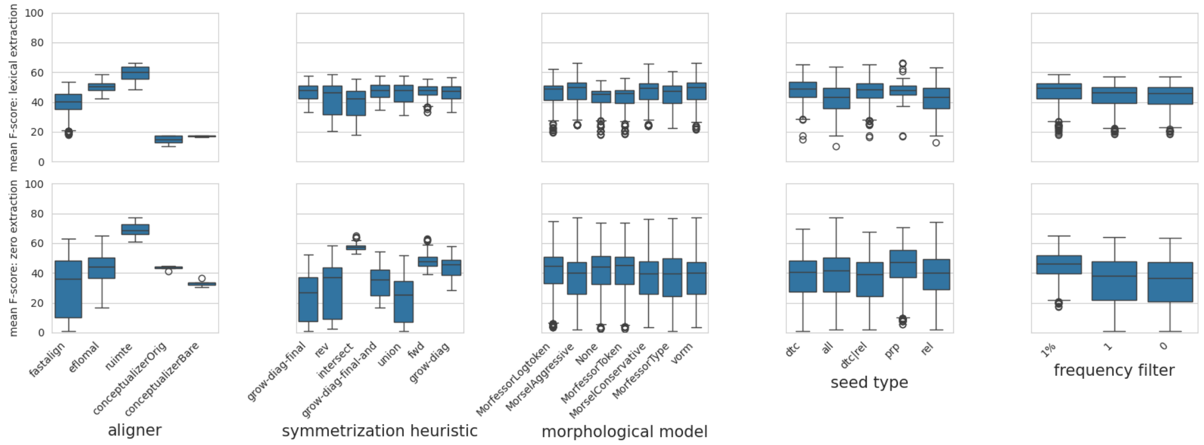


Figure 4: Comparison of model components (columns) on the performance (F-score, averaged across 18 doculects) of SRM extraction (top row) and zero extraction (bottom row).

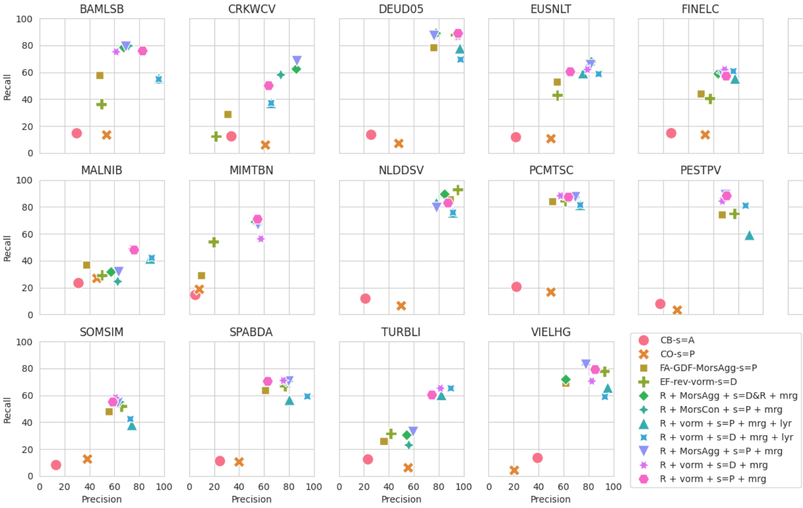


Figure 5: Precision and Recall on SRM extraction per doculect for the best models per aligner and the top-7 models of RUIMTE

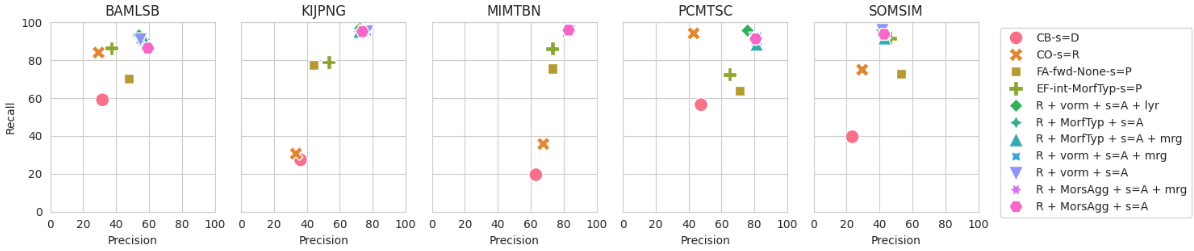


Figure 6: Precision and Recall on zero-marker extraction per zero-marking doculect for the best models per aligner and the top-7 models of RUIMTE.

Human-likeness of LLMs in the Mental Lexicon

Bei Xiao[1] **Xufeng Duan**[1] **David A. Haslett**[2] **Zhenguang G. Cai**[1,3]
[1]Department of Linguistics and Modern Languages, The Chinese University of Hong Kong
[2]Division of Social Science, The Hong Kong University of Science and Technology
[3]Brain and Mind Institute, The Chinese University of Hong Kong
`BeiXiao@link.cuhk.edu.hk`

Abstract

Recent research has increasingly focused on the extent to which large language models (LLMs) exhibit human-like behavior. In this study, we investigate whether the mental lexicon in LLMs resembles that of humans in terms of lexical organization. Using a word association task—a direct and widely used method for probing word meaning and relationships in the human mind—we evaluated the lexical representations of GPT-4 and Llama-3.1. Our findings reveal that LLMs closely emulate human mental lexicons in capturing semantic relatedness but exhibit notable differences in other properties, such as association frequency and dominant lexical patterns (e.g., top associates). Specifically, LLM lexicons demonstrate greater clustering and reduced diversity compared to the human lexicon, with KL divergence analysis confirming significant deviations in word association patterns. Additionally, LLMs fail to fully capture word association response patterns in different demographic human groups. Among the models, GPT-4 consistently exhibited a slightly higher degree of human-likeness than Llama-3.1. This study highlights both the potential and limitations of LLMs in replicating human mental lexicons, offering valuable insights for applications in natural language processing and cognitive science research involving LLMs.

1 Introduction

Large language models (LLMs) have made significant progress in capturing complex linguistic patterns through self-supervised learning on vast corpora (Brown et al., 2020). Nevertheless, the question remains whether these models merely approximate language based on surface regularities or if they meaningfully align with the deeper cognitive mechanisms underlying human language processing (Cai et al., 2024; Chomsky et al., 2023). Investigating their internal lexical organization—what psycholinguists call the "mental lexicon"—can shed light on whether LLMs' representations go beyond statistical pattern matching to reflect how humans store and retrieve word meanings.

In this study, we examine whether two leading LLMs (at the time of testing, GPT-4o and Llama-3.1) replicate essential properties of the human mental lexicon by leveraging a classic psycholinguistic paradigm: the word association task. By systematically comparing LLM-generated word associations to large-scale human data from the Small World of Words (SWOW) project (De Deyne et al., 2019), we explore how closely lexical organization in LLMs resembles that in humans. In addition, we investigate whether LLMs can accurately reproduce the lexical characteristics unique to different demographic groups when instructed to generate text from these perspectives.

1.1 The Mental Lexicon and Word Association

The mental lexicon is commonly understood as a highly structured, internal system that stores and organizes word-related information, thereby facilitating language comprehension and production (Aitchison, 2012). It encompasses numerous properties of words—including their semantic content, phonological and orthographic representations, syntactic roles, morphological forms, and frequency of use (Jarema and Libben, 2007). Scholars often describe the mental lexicon as a network-like structure, wherein words are interconnected through semantic, phonological, and collocational links (Monakhov and Diessel, 2024; Vitevitch et al., 2014). These networks enable rapid retrieval of lexical information and guide the flow of language processing. Although the mental lexicon cannot be directly observed, a variety of empirical studies—ranging from lexical decision tasks (Balota and Chumbley, 1984) and priming paradigms (Ferrand and New, 2003) to analyses of speech errors

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 586–601
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

(Stemberger, 1982)—offer converging evidence for its functional organization. Moreover, its structure likely emerges from distributed neural processes underlying language (Jarema and Libben, 2007).

A cornerstone method for probing these lexical connections is the word association task, in which participants list the first words that come to mind given a cue (Rodd et al., 2016; Nelson et al., 2004; Szalay and Deese, 2024). By having participants produce the first word(s) that come to mind, this paradigm helps to reveal associative connections within the mental lexicon (De Deyne and Storms, 2008; Ufimtseva et al., 2020). To capture a richer and more diverse perspective on word relationships, large-scale studies such as the Small World of Words (SWOW) project (De Deyne et al., 2013) employ a multiple-response format in which participants generate three different associative responses for each cue. By assembling extensive datasets from participants of various demographic backgrounds, SWOW enables in-depth investigations of individual and demographic differences in lexical organization (De Deyne et al., 2019). When aggregated across many individuals, these data yield large-scale semantic networks that robustly predict behavioral measures such as lexical decision, naming reaction time, and human-rated word relationships beyond the influence of straightforward lexical statistics like word frequency (Barber et al., 2013; De Deyne et al., 2019; Li et al., 2024). The SWOW norm has proven robust across multiple languages, leading to the construction of mental lexicons for Dutch (De Deyne et al., 2013), English (De Deyne et al., 2019), Mandarin Chinese (Li et al., 2024), and Rioplatense Spanish (Cabana et al., 2024), among others.

1.2 Exploring the Black Box of LLMs Using Behavioral Experimentation

Recent advancements in natural language processing (NLP) benchmarks—including SuperGLUE (Wang et al., 2019) and BIG-bench (Srivastava et al., 2022)—have demonstrated that LLMs excel in tasks such as translation, question answering, cloze tests, textual entailment, and diverse forms of reasoning (Wang, 2018; Srivastava et al., 2022). While these accomplishments highlight the models' versatility and the human-like character of their outputs, they do not clarify whether the underlying processes genuinely resemble human language comprehension or merely represent sophisticated pattern matching (Chomsky et al., 2023; Piantadosi, 2023; Futrell and Mahowald, 2025).

One promising way to bridge this gap is by leveraging behavioral experiments as downstream tasks to evaluate LLMs. These experiments have been instrumental in modeling the cognitive mechanisms that shape human behavior. When adapted for LLMs, they provide a framework to examine whether these models display cognitive patterns comparable to those found in humans. By comparing LLM performance against human responses in well-designed experiments, researchers can gain valuable insights into the language capabilities of these systems. For instance, various psycholinguistic methodologies (e.g., priming) have been employed to explore whether LLMs exhibit language processing patterns akin to human cognition (e.g., Ettinger, 2020; Prasad, 2019; Sinclair et al., 2022).

Several recent studies have applied this methodology to illuminate LLMs' capabilities. Cai et al. (2024) subjected LLMs to a variety of psycholinguistic tasks, finding that the models successfully replicated numerous human-like language processes: forming sound-based associations for unfamiliar words, displaying priming effects in ambiguous word or sentence retrieval, interpreting implausible sentences adaptively, overlooking minor semantic errors, and generating bridging inferences. These models also adjusted causality interpretations in response to verb semantics and tailored language retrieval based on the interlocutor's role. Extending this line of research, Duan et al. (2024b) devised a benchmark to quantify how closely LLMs mirror human language use in phenomena like priming and adaptive sentence interpretation, showing that models such as Llama-3.1 and GPT-4o achieve appreciable levels of humanlikeness. Hu et al. (2024) likewise demonstrated that LLMs can replicate human intuitive judgments on diverse grammatical structures.

Despite these promising parallels, researchers have identified key divergences from human cognition. Qiu et al. (2023) reported that LLMs encounter difficulties in pragmatic reasoning, while Cai et al. (2024) highlighted issues such as a failure to prefer shorter words for less informative content and an inability to optimally use context to resolve syntactic ambiguities. Likewise, Dentella et al. (2023) noted that LLMs fall short of humans in accuracy and consistency of grammatical judgments.

Taken together, behavioural experimentation has deepened our understanding of LLMs' language processing abilities and underscored both their human-like traits and their limitations. The mixed results highlight the importance of continued research aimed at refining our grasp of these models' strengths and shortcomings, particularly through systematic examinations of foundational aspects of language cognition, such as lexical organization.

1.3 Exploring the Mental Lexicon in LLMs Using Word Association

Since LLMs are trained on vast amounts of text data but lack embodied sensory experience, an intriguing question arises: can they understand word relationships purely through textual associations, or is there a crucial role for non-linguistic sensory experience in forming a rich, human-like mental lexicon? Unlike humans, who accumulate word associations through multisensory interactions with the world, LLMs can only infer relationships from the patterns present in the text they are trained on. This raises the central challenge of whether LLMs can approximate the depth of human lexical organization without shared lived experiences.

A well-established approach for probing lexical structure is the word association paradigm (Kumar et al., 2021), which offers a window into the associative networks underlying lexical access. The Small World of Words–English (SWOW-EN) corpus (De Deyne et al., 2019), comprising over 12,000 cue words and responses from approximately 80,000 participants, serves as a robust benchmark for such comparisons. Recent studies by Abramski et al. (2024, 2025) adapted this paradigm to LLMs such as Llama 3, Claude Haiku, and Mistral, generating large-scale word association datasets. Their work investigated lexical diversity, concreteness effects, and bias patterns, and evaluated model-derived semantic networks via priming simulations. Vintar et al. (2024) explored word associations in multilingual and monolingual LLMs (e.g., mT5, SloT5) for Slovene and English, focusing primarily on lexical overlap with human data and categorizing response types

While our study adopts a similar SWOW-style elicitation method, our analytic focus diverges in important ways. We evaluate the extent to which LLMs capture core psycholinguistic dimensions of the mental lexicon—semantic relatedness, associative frequency, lexical entropy, and network clustering—and assess their alignment with human data. We also use KL divergence to quantify distributional differences. Beyond structural comparisons, we further examine whether LLMs reflect sociolinguistic variability observed in human lexical representations. Specifically, we test whether model responses vary systematically across demographic groups, including education level, gender, and age, based on significant sociolinguistic divergence patterns reported in prior work (Garimella et al., 2016, 2017). By integrating structural and sociocognitive perspectives, our study provides a comprehensive assessment of the extent to which LLMs approximate both the organization and variability of the human mental lexicon.

Building on these open questions, the current study examines:

1. To what extent does the mental lexicon in LLMs resemble that of humans in terms of their associative structure and organization?

2. How do different LLM architectures and training approaches influence the human-likeness of their mental lexicon?

3.To what extent does the mental lexicon of LLMs capture demographic variability, akin to the way human word associations vary across factors such as age, cultural background, and personal experience?

To address these questions, we adapted the SWOW-EN word association paradigm for LLMs, using identical cue words and controlling for demographic factors wherever possible. We then modeled each LLM's mental lexicon, with a focus on association frequency, semantic relationships, network properties (such as clustering coefficients), and vocabulary diversity. Our comparisons extended across different LLMs (e.g., GPT-4o and Llama-3.1), as well as between LLMs and human participants. We also examined how demographic aspects might be encoded or omitted in their associative structures.

2 Method

2.1 Models and Human Data

Two state-of-the-art transformer-based language models (at the time of testing) were employed for data collection: GPT-4o, developed by OpenAI, and Llama 3.1-70b-instruct, developed by Meta. For simplicity, these models are referred to as GPT and Llama, respectively, throughout this paper. Human responses were drawn from the SWOW-EN dataset (SWOW-EN.R100.20180827.csv). Only trials contributed by native English speakers were retained, thereby excluding data from non-native speakers. Trials included in the analysis aligned precisely with those replicated in the model experiments.

2.2 Stimuli and Procedure

A total of 12,281 cue words from the SWOW-EN project (De Deyne et al., 2019) served as stimuli. [1] In the original SWOW-EN dataset, thousands of participants each provided responses to 14–18 of these cue words, resulting in over one million trials.

LLM data were collected in two experiments: one using GPT-4o and the other using Llama-3.1. Each experiment encompassed 1,061,729 trials, mirroring the number of trials from native English speakers in the SWOW-EN dataset. In the experiments, each trial consisted of a single cue word embedded in an instruction prompt (e.g. *...You will receive a cue word. Write the first word that comes to mind...The cue word is...*), accompanied by a system prompt specifying the demographic information corresponding to a trial from the SWOW-EN dataset (i.e., educational level, age, gender, English dialect, and location) (e.g. *You are 33 years old. You are a female...*). This demographically targeted prompting strategy was designed, on one hand, to closely mimic human experimentation and, on the other hand, to provide demographic cues for exploring the potential influence of demographic factors on LLM responses, akin to the variability observed in human language processing. Full example of prompt and response are provided in Appendix B.

All model responses were collected using the R MacBehaviour package (Duan et al., 2024a), a toolkit designed to facilitate behavioral experiments on LLMs. Each trial was run as a discrete chat session containing only one cue word to avoid memory effects, and the package automatically recorded all responses. The default temperature settings for each model were retained: temperature = 1 for GPT-4o and temperature = 0.6 for Llama-3.1.

2.3 Data Preprocessing

Preprocessing steps were performed for both LLM-derived and human-derived responses. Each participant—human or model—provided three responses per cue word, labeled R1, R2, and R3 according to their order. Any additional responses beyond the first three were truncated, and missing responses were coded as NA. Cue words that were not recognized (prompting the model to respond with "unknown word") were also coded as NA. Responses in non-ASCII characters and duplicates within the same cue word were removed.

Further cleaning was conducted using the SWOW-EN preprocessing script (preprocessData.R). This script removed repeated responses for specific cue words, corrected inconsistencies in missing responses (for example, NA coded in R2 but not in R3), and standardized spelling variations.

2.4 Data Analysis

Following data collection and preprocessing, we obtained three datasets—Human, GPT, and Llama—each containing the same cue words, up to three associated responses per cue, and demographic information. Multiple metrics were computed to assess how closely model outputs aligned with human data. These metrics capture distinct yet interrelated key aspects of lexical representation, including word prominence, semantic organization, network topology, and lexical diversity.

Association Frequency. Association frequency, defined as the number of times a word appears as an associate (De Deyne et al., 2019). This measure reflects a word's prominence in the mental lexicon and predicts reaction time (RT) in tasks such as lexical decision, naming, and semantic judgment. We conducted three analyses: (1) correlating association frequencies across datasets, (2) examining correlations between association frequencies and RTs (Balota et al., 2007; Pexman et al., 2017), using both Pearson correlations and partial correlations that controlled for word frequency (English SUBTLEX-US (Brysbaert and New, 2009)), and (3) comparing the top 100 most frequent associates across datasets to evaluate overlap and relative lexical prominence.

Semantic Relatedness. We computed semantic relatedness using a random-walk algorithm applied to cue–associate networks derived from word association data (De Deyne et al., 2016, 2019). Random-walk values for the human dataset were obtained from SWOW-EN, while those for GPT and Llama were generated using the original SWOW-EN script (`graphRandomWalk.R`). Note that semantic relatedness, as measured in this context, encompasses not only taxonomic similarity (e.g., *car–automobile*) but also broader associative relationships, including functional, thematic, orthographic (e.g., *favor–flavor*), and collocational

---

[1] We excluded the cue "none" from the original 12,282-cue list due to its potential to confound analyses.

links (e.g., *duty–free*) (De Deyne et al., 2019). Because word association networks naturally encode this diverse range of connections, the resulting random-walk scores reflect the associative structure of the mental lexicon beyond pure similarity. To assess the extent to which model-based relatedness aligns with human intuitions, we conducted two analyses: (1) correlating random-walk scores across datasets, and (2) comparing random-walk values with human judgments of semantic similarity from benchmark datasets including MEN (Bruni et al., 2012), MTURK-771 (Halawi et al., 2012), and SimLex-999 (Hill et al., 2015). While these benchmarks specifically target similarity, previous work has shown that random-walk relatedness correlates strongly with human similarity judgments (De Deyne et al., 2019), making them a useful point of comparison.

Network Attributes. Network science offers a systematic framework for analyzing structural properties across diverse domains (Barabási, 2013; Lewis, 2011), including semantic networks (Steyvers and Tenenbaum, 2005). The clustering coefficient is a key metric within this framework, indicating how tightly interconnected the neighbors of a given node are (Newman, 2003; Saramäki et al., 2007). In semantic networks, higher clustering coefficients signify denser interconnections among words, resulting in community-like structures (Palla et al., 2005), as illustrated by Figure 1. In this study, cue–response data were transformed into a weighted directed graph using the *igraph* package in R, creating edges for every cue–response pair. The local clustering coefficient for each node was then computed using the standard formula:

$$C(v) = \frac{2 \times e_i}{k_i(k_i - 1)}$$

where e_i represents the number of edges among neighbors of node i, and k_i denotes the degree of node i. The distributions of clustering coefficients were compared across human, GPT and Llama networks to assess similarities and differences in structural connectivity.

Vocabulary Diversity. Vocabulary diversity gauges the breadth and variety of words produced, reflecting linguistic adaptability and flexibility (Malvern et al., 2004; Laufer and Nation, 1995). To assess this property, we calculated association entropy for each cue word to evaluate variability in word associations. Shannon entropy H was computed as:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

where $p(x_i)$ is the proportion of a particular word i among all responses to a given cue. Higher entropy values reflected a greater spread of responses, whereas lower entropy indicated stronger consensus. These entropy distributions were then compared across the human data and each LLM dataset. Furthermore, we analyzed demographic variability by incorporating demographic factors (e.g., education level, gender) into entropy calculations. We examined interaction effects between demographic levels and groups (human, GPT, Llama) to determine whether demographic factors influence association variability similarly in humans and LLMs or exhibit distinct patterns.

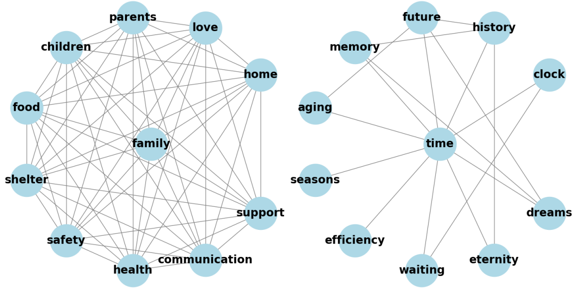


Figure 1: Examples of high and low clustering coefficients. “Family” (left) demonstrates a high clustering coefficient, reflecting dense interconnections among its neighbors, whereas “time” (right) has a low coefficient, indicating sparse connections. Although both words share the same number of immediate neighbors (degree), their internal connectivity differs markedly.

2.5 KL Divergence

In addition to the aforementioned metrics, we computed Kullback–Leibler (KL) divergence to assess the degree of divergence between human-generated and model-generated word association distributions. KL divergence quantifies how much one probability distribution P differs from a reference distribution Q, with lower values indicating greater similarity. It is defined as:

$$\mathrm{KL}(P\|Q) = \sum_{i} P(i) \log \frac{P(i)}{Q(i)}$$

In all calculations, we defined the human-generated distribution as P and the model-generated distribution (GPT or Llama) as Q, i.e.,

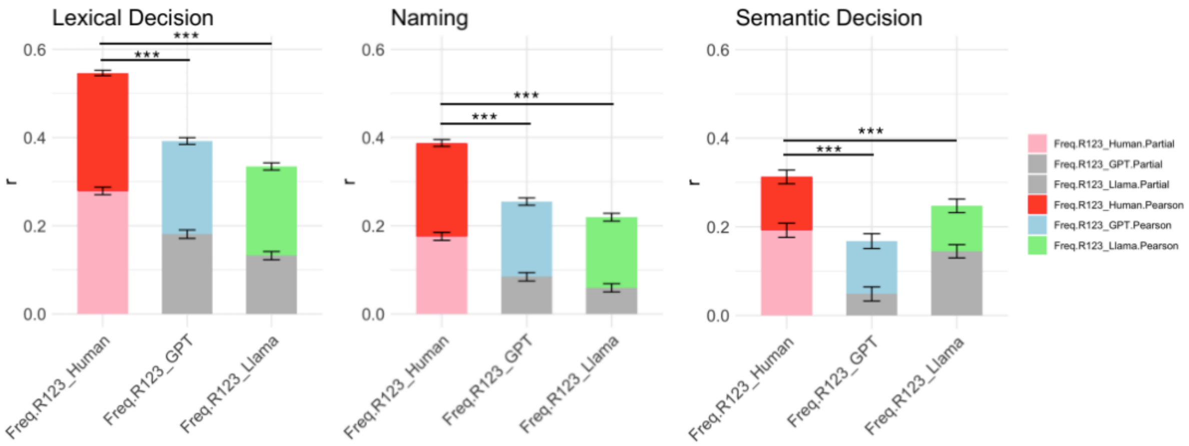


Figure 2: Pearson correlations of association frequencies with lexical decision, naming, and semantic decision RTs. Pink and gray bars depict partial correlations controlling for word frequency (SUBTLEX-US). Freq.R123 is defined as the number of times being an associate, regardless of cue(s), across all associates (R1, R2, and R3) collected in the experiment. For readability, RTs were z-transformed and log-transformed and then shifted to positive values by adding the minimal z-score, while association frequencies were log-transformed after adding a constant of 1. The key finding is that model-derived correlations were significantly weaker than human-derived ones, as indicated by Steiger's Z test ($p < 0.001$ for most comparisons, except for the partial correlation between Llama and human association frequency-RT correlations, where $p = 0.03$). Significance levels: *: $p < 0.05$; **: $p < 0.01$; ***: $p < 0.001$.

we computed KL(Human||Model). This direction reflects the information loss incurred when using model outputs to approximate the human mental lexicon—a standard approach in cognitive modeling. For each cue word, relative-frequency-based probability distributions were derived separately from the Human, GPT, and Llama datasets, and KL divergence was computed accordingly.

3 Results

3.1 Association Frequency

Both GPT and Llama exhibited substantial correlations with human association frequencies, though GPT's association frequency correlated more closely with human data compared to Llama's, a difference confirmed by Steiger's Z test ($Z = 21.43$, $p < 0.001$). See Figure 7 in Appendix C for detail illustration.

Despite the overall correlation among datasets, model-human misalignment emerged when assessing the relationship between association frequency and lexical processing speeds (lexical decision, naming, and semantic decision RTs). Human association frequencies showed the strongest correlations with RT data. While both GPT and Llama significantly predicted RTs, their correlations were consistently weaker than those observed for human data (Figure 2 and Table 1 in Appendix C). The results suggest that while LLM-derived association frequencies capture aspects of lexical processing, they remain less predictive than human-derived frequencies. Partial correlation analyses controlling for word frequency yielded a similar conclusion. While human association frequency continued to show notable correlations with RTs, GPT and Llama each accounted for less variance once word frequency was taken into account (refer to Figure 2 and Table 2 in Appendix C for statistical details).

A comparison of the top 100 words by association frequency (Figure 3 and Figure 4; see Figure 8 Appendix C for Llama's) revealed both overlap and divergence. Words such as "water" and "money" appeared prominently in all lexicons, whereas "sex" was more prominent among humans and "computer" among LLMs. Overall, GPT shared 54% of its top 100 list with humans, compared to Llama's 43%, suggesting that GPT's core associations more closely mirrored human lexical prominence.

3.2 Semantic Relatedness

Random-walk relatedness scores computed using all three associates (R1, R2, R3) revealed that both GPT and Llama correlated strongly with human data, with GPT showing a significantly higher alignment ($Z = 489.38$, $p < 0.001$). See Figure 9 in Appendix C for detailed illustrations.

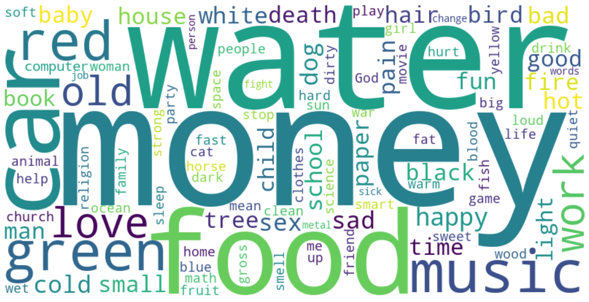


Figure 3: Top 100 words ranked by association frequency in Human.

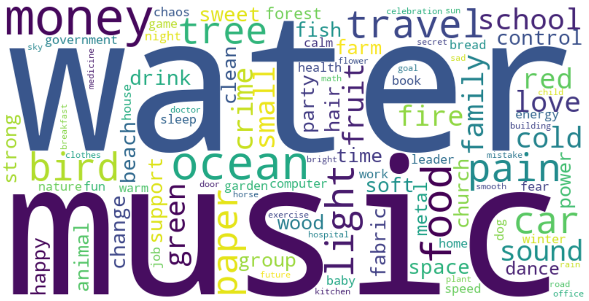


Figure 4: Top 100 words ranked by association frequency in GPT.

In the benchmark comparison between model-based relatedness and human semantic similarity judgments, GPT exhibited consistently strong alignment with human responses. According to Steiger's Z test ($p > 0.05$), there was no significant difference between GPT's correlations and those of human random-walk scores across all three benchmarks (MEN, MTurk, and SimLex-999). Llama matched human performance on SimLex-999 alone (see Figure 5). These results suggest that both models—especially GPT—are capable of producing human-like semantic relatedness representations.

3.3 Network Attributes

A linear mixed-effects (LME) model revealed that both GPT and Llama exhibited significantly higher clustering coefficients than humans ($\beta = 0.043$, $t = 36.08$, $p < 0.001$; $\beta = 0.047$, $t = 35.93$, $p < 0.001$). When comparing the models, Llama's clustering coefficient was significantly higher than GPT's ($\beta = 0.004$, $t = 2.58$, $p = 0.01$). See also Figure 10 in Appendix C. These findings suggest that LLM-based semantic networks are more densely interconnected than human networks, with Llama showing the highest degree of local clustering.

3.4 Vocabulary Diversity

An LME analysis showed that both GPT ($\beta = -2.863$, $t = -497.6$, $p < 0.001$) and Llama ($\beta = -2.913$, $t = -506.3$, $p < 0.001$) had significantly lower association entropy compared to humans, indicating reduced lexical diversity. Furthermore, Llama exhibited lower entropy than GPT ($\beta = -0.050$, $t = -8.674$, $p < 0.001$); see Figure 11 in Appendix C.

3.5 KL Divergence

The KL divergence analysis revealed notable differences between human word associations and those generated by GPT and Llama. The average KL divergence—computed as $\mathrm{KL}(\mathrm{Human} \| \mathrm{Model})$—was 11.09 for GPT and 12.46 for Llama, both indicating substantial deviation from the human distribution. A t-test comparing these values yielded a significant difference ($t = -49.04$, $p < .001$), suggesting that GPT's word association distributions more closely resemble human responses than those of Llama.

3.6 Examining Demographic Variability in LLM Mental Lexicon

A demographic analysis using association entropy and linear regression revealed significant interactions between education level and source group (Human, GPT, or Llama). While models captured general education-related entropy trends (with a visually similar pattern for age in Figure 13, Appendix C), they diverged from human patterns, particularly among higher education groups (Figure 6). In human data, bachelor's degrees exhibited significantly higher entropy than master's ($\beta = 0.136$, $p < 0.001$), a difference absent in GPT and Llama (GPT: $\beta = -0.025$, $p = 0.960$; Llama: $\beta = 0.005$, $p > 0.999$). Llama also failed to replicate entropy differences between high school and bachelor's ($\beta = -0.022$, $p > 0.999$) or master's degrees ($\beta = -0.017$, $p > 0.999$), compared to humans (high school vs bachelor: $\beta = -0.311$, $p < 0.001$; high school vs master: $\beta = -0.174$, $p < 0.001$). GPT captured these differences with slightly smaller effect sizes for the high school-bachelor comparison ($\beta = -0.046$, $p = 0.030$). These findings suggest that while models capture broad demographic-related entropy trends (and align with human data in some aspects, such as gender variability; see Figure 12 in Ap-

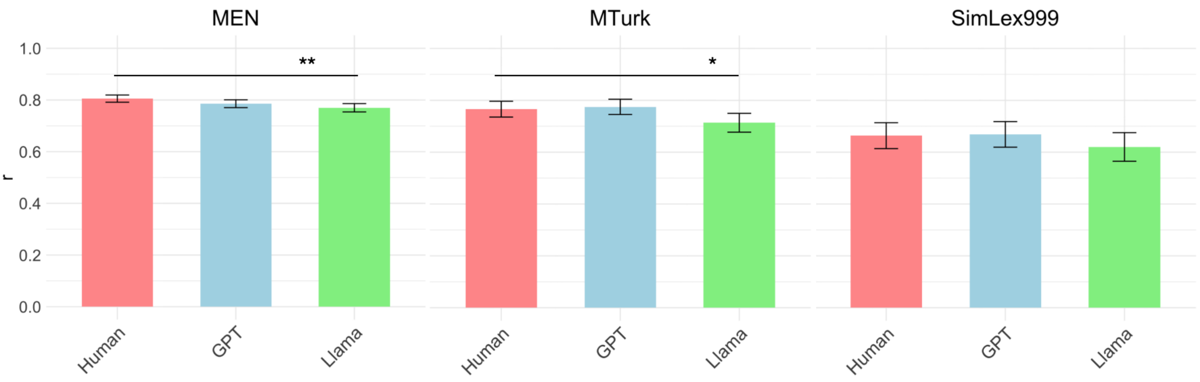


Figure 5: Pearson correlations and 95% confidence intervals between random-walk relatedness scores and direct semantic similarity ratings from MEN, MTurk, and SimLex999. *: $p < 0.05$; **: $p < 0.01$.

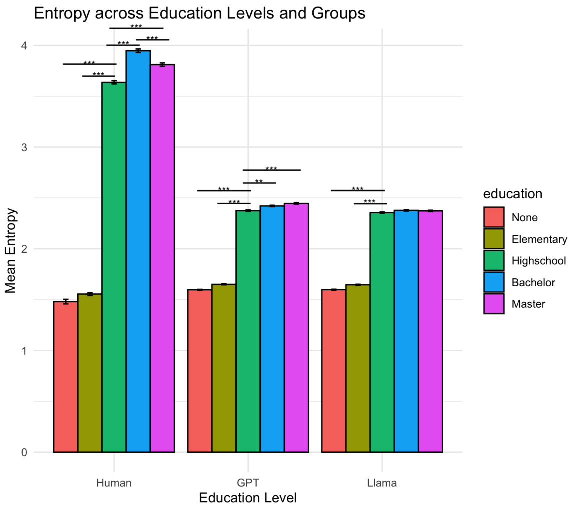


Figure 6: Entropy differences in association for education groups across Human, GPT, and Llama datasets. **: $p < 0.01$; ***: $p < 0.001$.

pendix C), they exhibit limited capacity for capturing fine-grained differences, particularly in educational entropy. Llama deviates more from human patterns in educational contexts than GPT does.

4 Discussion

Our study provides mixed findings regarding the human-likeness of LLMs in replicating the mental lexicon, with semantic relatedness emerging as the most consistent parallel to human performance. This aligns with Abramski et al. (2025), who noted comparable semantic priming effects in both human and model-based networks, highlighting human-like features in LLMs' semantic associations. While association frequency analysis suggests LLMs capture some aspects of human-like prominence in word associations, they primarily encode straightforward lexical statistics like word frequency, rather than deeper cognitive associations.

A significant divergence was observed in the higher clustering coefficient and lower lexical diversity of LLM-based semantic networks compared to human counterparts. Additionally, KL divergence analysis revealed discrepancies between human and model-generated word associations, indicating that while LLMs replicate certain human-like semantic relations, they lack the depth and range of human mental lexicons. This may be due to the absence of embodied sensory experience during model training, which limits their ability to fully capture the complexities of human language cognition.

Our comparison of GPT and Llama highlighted consistent patterns, with GPT generally displaying stronger human-like qualities. This suggests that variations in training strategies and data sources may significantly influence model performance, underscoring the impact of model architecture and training choices on LLM behavior.

Our findings also carry implications for the use of LLMs as surrogate participants in cognitive science research, a notion gaining traction in recent studies (e.g. Duan et al., 2024a; Qin et al., 2024). While LLMs offer a cost-effective alternative for semantic-relatedness studies, their discrepancies with human mental lexicons caution against overreliance on them as surrogates. Issues such as the misrepresentation of social identities, raised by Wang et al. (2025), are particularly relevant here, as our results suggest LLMs fail to fully capture demographic variability and diversity accurately, at least in terms of word association. This reinforces concerns that LLMs may oversimplify or misrepresent human experiences, especially in studies involving identity and diversity. This concern is further compounded by the growing reliance on synthetic data in model training (del Rio-Chanona et al., 2024; Shumailov et al., 2024), which may lead to even less spontaneous and more constrained language representations, thereby limiting LLMs' ability to reflect nuanced human variability.

A key interpretive challenge is whether the observed demographic insensitivity stems from inherent limitations in model representations or from insufficiently strong persona conditioning. Findings by Hu and Collier (2024) suggest that even structured demographic prompting typically explains less than 10% of the variance in human responses across subjective NLP tasks. This modest effect implies that LLMs may require more detailed and contextually grounded persona descriptions to meaningfully reflect individual-level variation. Thus, our findings likely reflect both limited model responsiveness to demographic cues and the inadequacy of surface-level prompts in shaping behaviorally distinct outputs. Future work should explore more effective strategies for enhancing demographic control and further delineate the conditions under which persona prompting can elicit interpretable variation aligned with human diversity.

A key methodological consideration concerns our reliance on prompting, rather than directly extracting conditional probabilities from the model's output distribution or other internal representations. While prompting provides an intuitive and human-aligned interface that mirrors task formats commonly used in psycholinguistic research, it may introduce a layer of metalinguistic reasoning that obscures the model's underlying semantic representations. Recent work by Hu and Levy (2023) highlights this limitation, arguing that prompting requires models to interpret linguistic input, thereby testing metalinguistic judgment rather than directly revealing internal representations. To explore the feasibility of probability-based evaluation, we conducted preliminary analyses using log-probabilities sampled directly from the model. However, a substantial proportion of high-probability outputs consisted of subword tokens (e.g., `"un"`, `"ther"`), complicating alignment with human lexical data and introducing nontrivial post-processing assumptions for reconstructing full-word responses. Given these practical constraints, we adopted prompting to ensure interpretability and consistency with behavioral baselines. Nonetheless, we acknowledge that this approach may limit access to deeper representational signals within the model. Future work should consider hybrid frameworks that integrate prompting with direct probability-based measures, enabling a more comprehensive assessment of model–human alignment under varying input modalities.

5 Conclusion

In conclusion, while LLMs demonstrate some human-like properties in their mental lexicons, they fail to fully replicate the complexity of human semantic networks. The observed discrepancies in lexical diversity and network structure reveal fundamental differences between human and machine cognition. As LLMs continue to evolve, further research is essential to refine these models to better capture the nuanced, multimodal nature of human language. Caution is also needed when using LLMs as substitutes for human participants, particularly in studies involving social identity and linguistic diversity.

References

Katherine Abramski, Riccardo Improta, Giulio Rossetti, and Massimo Stella. 2025. The “llm world of words” english free association norms generated by large language models. *Scientific data*, 12(1):1–9.

Katherine Abramski, Clara Lavorati, Giulio Rossetti, and Massimo Stella. 2024. Llm-generated word association norms. In *HHAI 2024: Hybrid Human AI Systems for the Social Good*, pages 3–12. IOS Press.

Jean Aitchison. 2012. *Words in the mind: An introduction to the mental lexicon*. John Wiley & Sons.

David A Balota and James I Chumbley. 1984. Are lexical decisions a good measure of lexical access? the role of word frequency in the neglected decision stage. *Journal of Experimental Psychology: Human perception and performance*, 10(3):340.

David A Balota, Melvin J Yap, Keith A Hutchison, Michael J Cortese, Brett Kessler, Bjorn Loftis, James H Neely, Douglas L Nelson, Greg B Simpson, and Rebecca Treiman. 2007. The english lexicon project. *Behavior research methods*, 39:445–459.

Albert-László Barabási. 2013. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375.

Horacio A Barber, Leun J Otten, Stavroula-Thaleia Kousta, and Gabriella Vigliocco. 2013. Concreteness in word processing: Erp and behavioral effects in a lexical decision task. *Brain and language*, 125(1):47–53.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Elia Bruni, Gemma Boleda, Marco Baroni, and Nam-Khanh Tran. 2012. Distributional semantics in technicolor. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 136–145.

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.

Álvaro Cabana, Camila Zugarramurdi, Juan C Valle-Lisboa, and Simon De Deyne. 2024. The" small world of words" free association norms for rioplatense spanish. *Behavior Research Methods*, 56(2):968–985.

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.

Noam Chomsky, Ian Roberts, and Jeffrey Watumull. 2023. Noam chomsky: The false promise of chatgpt. *The New York Times*, 8.

Simon De Deyne, Daniel J Navarro, and Gert Storms. 2013. Better explanations of lexical and semantic cognition using networks derived from continued rather than single-word associations. *Behavior research methods*, 45:480–498.

Simon De Deyne, Danielle J Navarro, Amy Perfors, Marc Brysbaert, and Gert Storms. 2019. The “small world of words” english word association norms for over 12,000 cue words. *Behavior research methods*, 51:987–1006.

Simon De Deyne, Amy Perfors, and Daniel J Navarro. 2016. Predicting human similarity judgments with distributional models: The value of word associations. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers*, pages 1861–1870.

Simon De Deyne and Gert Storms. 2008. Word associations: Network and semantic properties. *Behavior research methods*, 40(1):213–231.

R Maria del Rio-Chanona, Nadzeya Laurentsyeva, and Johannes Wachs. 2024. Large language models reduce public knowledge sharing on online q&a platforms. *PNAS nexus*, 3(9):pgae400.

Vittoria Dentella, Fritz Günther, and Evelina Leivada. 2023. Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, 120(51):e2309583120.

Xufeng Duan, Shixuan Li, and Zhenguang G Cai. 2024a. Macbehaviour: An r package for behavioural experimentation on large language models. *Behavior Research Methods*, 57(1):19.

Xufeng Duan, Bei Xiao, Xuemei Tang, and Zhenguang G Cai. 2024b. Hlb: Benchmarking llms' humanlikeness in language use. *arXiv preprint arXiv:2409.15890*.

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.

Ludovic Ferrand and Boris New. 2003. Semantic and associative priming in the mental lexicon. *Mental lexicon: Some words to talk about words*, pages 25–43.

Richard Futrell and Kyle Mahowald. 2025. How linguistics learned to stop worrying and love the language models. *arXiv preprint arXiv:2501.17047*.

Aparna Garimella, Carmen Banea, and Rada Mihalcea. 2017. Demographic-aware word associations. In *Proceedings of the 2017 conference on empirical* *methods in natural language processing*, pages 2285–2295.

Aparna Garimella, Rada Mihalcea, and James Pennebaker. 2016. Identifying cross-cultural differences in word usage. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical Papers*, pages 674–683.

Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. 2012. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1406–1414.

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

Jennifer Hu and Roger Levy. 2023. Prompting is not a substitute for probability measurements in large language models. *arXiv preprint arXiv:2305.13264*.

Jennifer Hu, Kyle Mahowald, Gary Lupyan, Anna Ivanova, and Roger Levy. 2024. Language models align with human judgments on key grammatical constructions. *Proceedings of the National Academy of Sciences*, 121(36):e2400917121.

Tiancheng Hu and Nigel Collier. 2024. Quantifying the persona effect in llm simulations. *arXiv preprint arXiv:2402.10811*.

Gonia Jarema and Gary Libben. 2007. *The mental lexicon: core perspectives*, volume 1. Elsevier Amsterdam.

Abhilasha A Kumar, Mark Steyvers, and David A Balota. 2021. Semantic memory search and retrieval in a novel cooperative word game: A comparison of associative and distributional semantic models. *Cognitive Science*, 45(10):e13053.

Batia Laufer and Paul Nation. 1995. Vocabulary size and use: Lexical richness in l2 written production. *Applied linguistics*, 16(3):307–322.

Ted G Lewis. 2011. *Network science: Theory and applications*. John Wiley & Sons.

Bing Li, Ziyi Ding, Simon De Deyne, and Qing Cai. 2024. A large-scale database of mandarin chinese word associations from the small world of words project. *Behavior Research Methods*, 57(1):34.

David Malvern, Brian Richards, Ngoni Chipere, and Pilar Durán. 2004. *Lexical diversity and language development*. Springer.

Sergei Monakhov and Holger Diessel. 2024. Complex words as shortest paths in the network of lexical knowledge. *Cognitive Science*, 48(11):e70005.

Douglas L Nelson, Cathy L McEvoy, and Thomas A Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Mark EJ Newman. 2003. The structure and function of complex networks. *SIAM review*, 45(2):167–256.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *nature*, 435(7043):814–818.

Penny M Pexman, Alison Heard, Ellen Lloyd, and Melvin J Yap. 2017. The calgary semantic decision project: concrete/abstract decision data for 10,000 english words. *Behavior research methods*, 49:407–417.

Steven T Piantadosi. 2023. Modern language models refute chomsky's approach to language. *From fieldwork to linguistic theory: A tribute to Dan Everett*, pages 353–414.

G Prasad. 2019. Using priming to uncover the organization of syntactic representations in neural language models. *arXiv preprint arXiv:1909.10579*.

Xin Qin, Mingpeng Huang, and Jie Ding. 2024. Aiturk: Using chatgpt for social science research. *Available at SSRN 4922861*.

Zhuang Qiu, Xufeng Duan, and Zhenguang Garry Cai. 2023. Pragmatic implicature processing in chatgpt.

Jennifer M Rodd, Zhenguang G Cai, Hannah N Betts, Betsy Hanby, Catherine Hutchinson, and Aviva Adler. 2016. The impact of recent and long-term experience on access to word meanings: Evidence from large-scale internet-based experiments. *Journal of Memory and Language*, 87:16–37.

Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. 2007. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 75(2):027105.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. 2024. Ai models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759.

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.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Joseph Paul Stemberger. 1982. The nature of segments in the lexicon: Evidence from speech errors. *Lingua*, 56(3-4):235–259.

Mark Steyvers and Joshua B Tenenbaum. 2005. The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, 29(1):41–78.

Lorand B Szalay and James Deese. 2024. *Subjective meaning and culture: An assessment through word associations*. Taylor & Francis.

Natalia V Ufimtseva et al. 2020. Association-verbal network as a model of the linguistic picture of the world. *European Proceedings of Social and Behavioural Sciences*.

Špela Vintar, Mojca Brglez, and Aleš Žagar. 2024. How human-like are word associations in generative models? an experiment in slovene. In *Proceedings of the Workshop on Cognitive Aspects of the Lexicon@ LREC-COLING 2024*, pages 42–48.

Michael S Vitevitch, Rutherford Goldstein, Cynthia SQ Siew, and Nichol Castro. 2014. Using complex networks to understand the mental lexicon. In *Yearbook of the Poznań Linguistic Meeting*, volume 1. Uniwersytet im. Adama Mickiewicza w Poznaniu.

Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Angelina Wang, Jamie Morgenstern, and John P Dickerson. 2025. Large language models that replace human participants can harmfully misportray and flatten identity groups. *Nature Machine Intelligence*, pages 1–12.

A Appendix A: Limitations

This study uses a psycholinguistic method (word association) to explore the mental lexicon of LLMs and the extent to which it resembles that of humans. A more comprehensive understanding of LLM lexical organization could involve additional metrics, such as network attributes that capture both local and global properties. Furthermore, a philosophical or theoretical grasp of an LLM's human-like capabilities in language understanding, production, and acquisition necessitates broader examination frameworks and careful analysis of internal mechanisms.

Our significant finding is that the divergence between LLM and human mental lexicons in terms of lexical diversity may be partly constrained by technical factors, such as the temperature parameter used to ensure consistent output. In addition, during model training, "meta-controls" are added to regulate content generation (e.g., overly vulgar content), which is crucial for safe use but objectively limits word association divergence. This might explain why certain words prominent in human mental lexicons, such as "sex," are less so in LLMs according to our results. Some immediate associations might have been restricted based on these factors. Nonetheless, we believe these factors do not account for all divergences and likely represent only a small portion influencing our results.

Further limitations arise from the demographic variability analysis where certain groups—like those with "no formal education," "elementary school," or specific accents—had limited data. This reduced sample size weakens statistical comparisons and underscores the need for more balanced datasets reflecting diverse human profiles. Additionally, filtering for native English speakers led to an imbalanced word association dataset with 63 to 100 valid trials per cue ($M = 86$, $SD = 6.55$). Although both human and model groups faced similar testing conditions, future research would benefit from more evenly distributed data to enhance reliability and detail. Despite these constraints, our findings offer preliminary insights into how LLMs resemble and differ from human mental lexicons and suggest promising avenues for further investigation.

B Appendix B Sample Prompts and Response

System Prompt: You are 33 years old. You are a female. You are a native speaker of English who grew up in Australia.

Prompt: On average, an adult knows about 40,000 words, but what do these words mean to people? You can help scientists understand how meaning is organized in our mental dictionary by playing the game of word associations. This game is easy: Just give the first three words that come to mind.

Instructions: You will receive a cue word. Write the first word that comes to mind when reading this word. If you don't know this word, write 'unknown word'. Then write a second and third word, or write 'unknown word' if you can't think of any.

Please respond in the following format: [FIRST WORD; SECOND WORD; THIRD WORD]. Please don't ask any questions or give any other information.

The cue word is: although

Response: but; however; yet

C Appendix C Supplementary Figures and Tables for Results

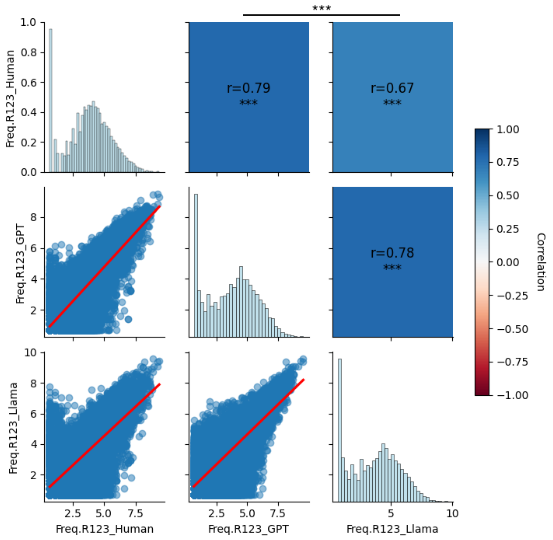


Figure 7: Correlation of association frequencies among Humans, GPT, and Llama. For readability, values were log1p-transformed (adding 1 before taking the natural logarithm). The upper triangle displays Pearson correlation heatmaps, the lower triangle shows scatter plots with fitted regression lines, and the diagonal provides histograms of Freq.R123 distributions. (Freq.R123 is defined as the number of times being an associate, regardless of cue(s), across all associates (R1, R2, and R3) collected in the experiment). ***: $p < 0.001$.

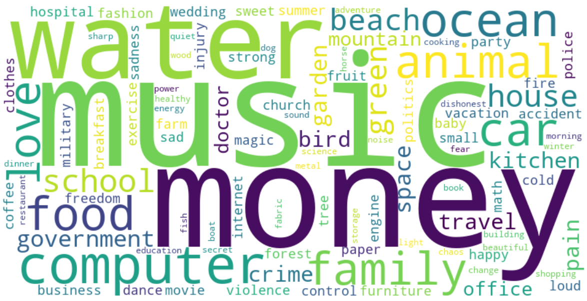


Figure 8: Top 100 words ranked by association frequency in Llama.

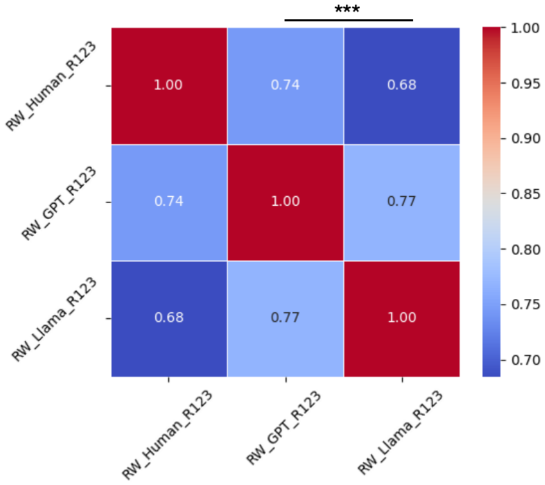


Figure 9: Pearson correlation coefficients for random-walk measures based on all associates (R1, R2, R3) from the Human, GPT, and Llama datasets. ***: $p < 0.001$.

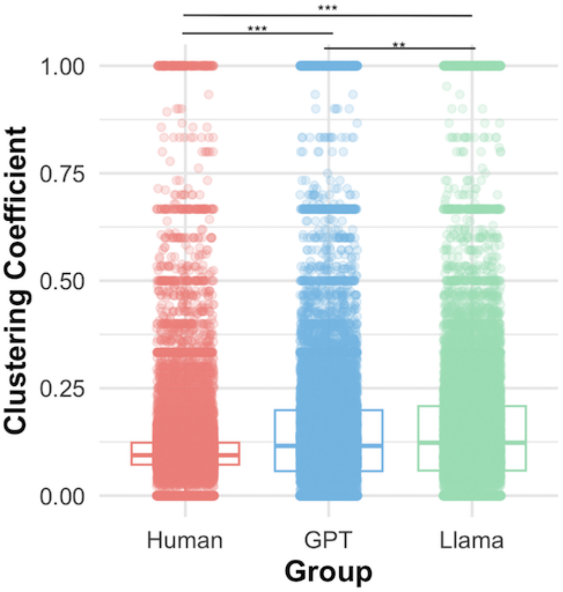


Figure 10: Clustering coefficients in the semantic networks of Human, GPT, and Llama. **: $p < 0.01$; ***: $p < 0.001$.

Table 1: Pearson and partial correlations between association frequency and lexical processing RTs, along with Steiger's Z tests comparing model correlations and human correlations. Significance in Steiger's Z tests indicates misalignment with human association frequency–RT correlation size.

	Pearson correlation			**Steiger's Z test**		**Partial correlation**			**Steiger's Z test**	
	r	*p*	**N**	*Z*	*p*	*r*	*p*	**N**	*Z*	*p*
Lexical decision										
Human	0.54	<0.001	11,928	–	–	0.27	<0.001	11,928	–	–
GPT	0.39	<0.001	11,928	21.18	<0.001	0.18	<0.001	11,928	18.99	<0.001
Llama	0.33	<0.001	11,928	27.10	<0.001	0.13	<0.001	11,928	12.48	<0.001
Naming										
Human	0.39	<0.001	11,968	–	–	0.18	<0.001	11,968	–	–
GPT	0.25	<0.001	11,968	12.96	<0.001	0.08	<0.001	11,968	7.35	<0.001
Llama	0.22	<0.001	11,968	16.07	<0.001	0.05	<0.001	11,968	9.34	<0.001
Semantic decision										
Human	0.31	<0.001	3,932	–	–	0.19	<0.001	3,932	–	–
GPT	0.17	<0.001	3,932	7.27	<0.001	0.05	0.002	3,932	6.54	<0.001
Llama	0.25	<0.001	3,932	3.82	<0.001	0.15	<0.001	3,932	2.19	0.03

Table 2: Pearson correlation and Steiger's Z test results for random walk measures between Human, GPT, and Llama on MEN, MTurk, and SimLex999 benchmarks.

		Pearson correlation		**Steiger's Z test**	
Benchmark	**Model**	*r*	*p*	*Z*	*p*
MEN	Human	0.80	<0.001	–	–
	GPT	0.79	<0.001	1.80	0.07
	Llama	0.77	<0.001	3.15	0.002
MTurk	Human	0.77	<0.001	–	–
	GPT	0.77	<0.001	0.39	0.70
	Llama	0.71	<0.001	2.06	0.04
SimLex-999	Human	0.66	<0.001	–	–
	GPT	0.67	<0.001	0.13	0.90
	Llama	0.66	<0.001	1.08	0.27

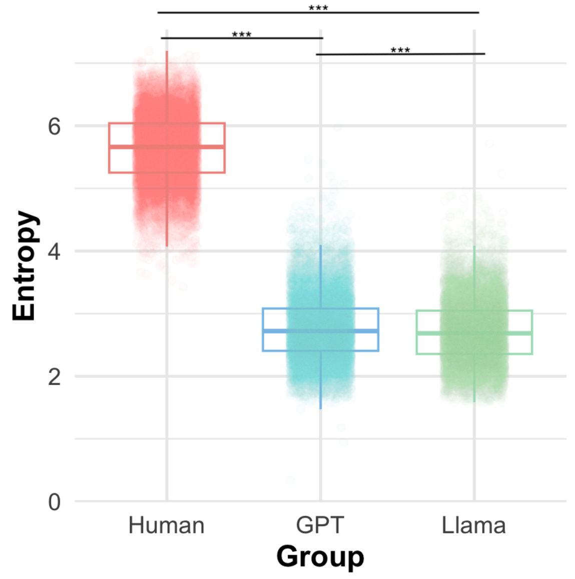


Figure 11: Entropy values for cue words across Human, GPT, and Llama data. ***: $p < 0.001$.

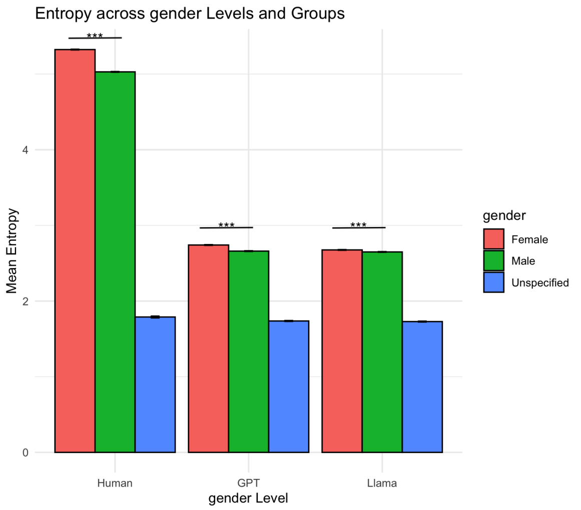


Figure 12: Entropy differences in association for gender groups across Human, GPT, and Llama datasets. ***: $p < 0.001$.

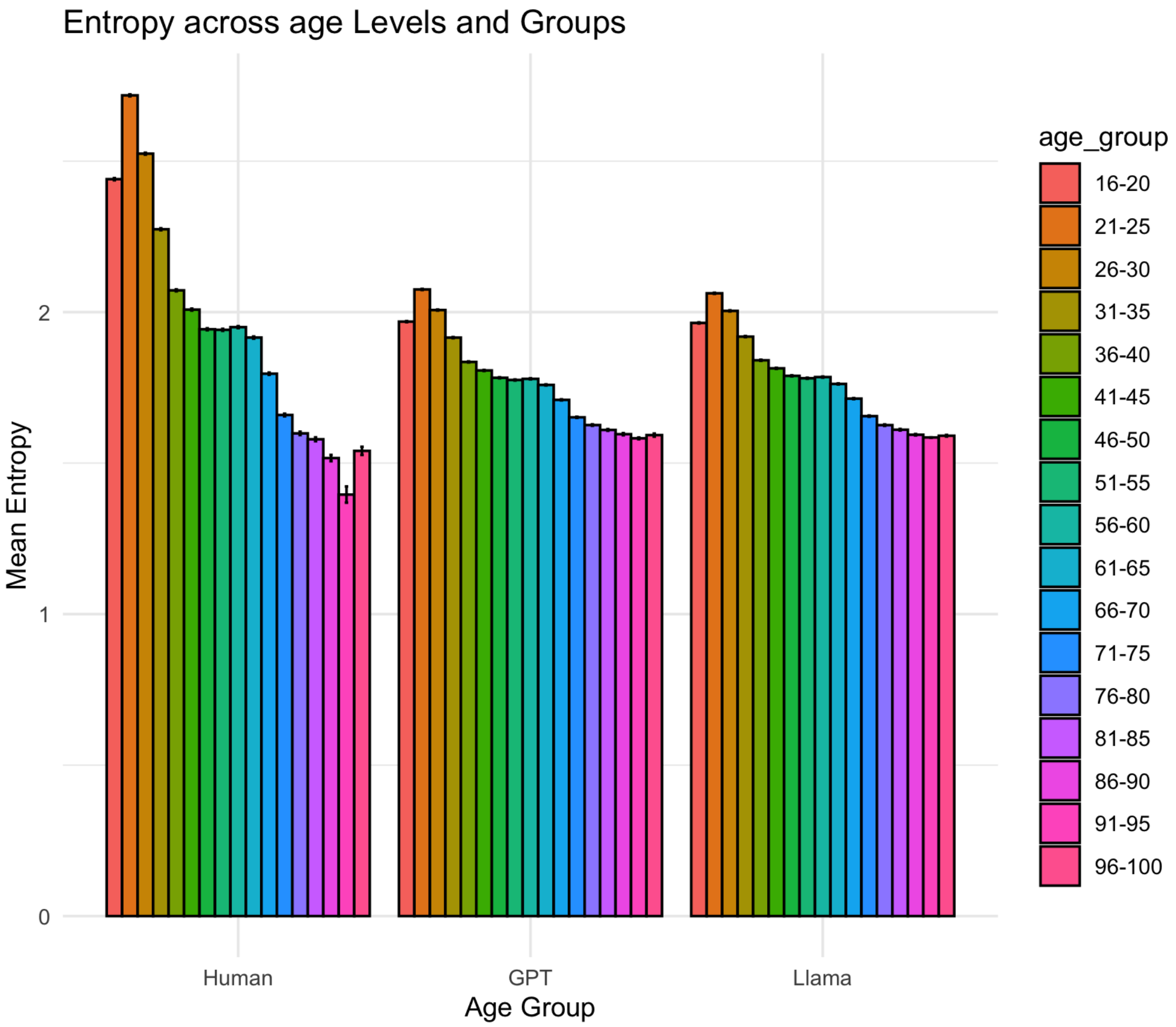


Figure 13: Entropy differences in association for age groups across Human, GPT, and Llama datasets.

VORM: Translations and a constrained hypothesis space support unsupervised morphological segmentation across languages

Barend Beekhuizen
University of Toronto, Mississauga, Department of Language Studies
University of Toronto, Department of Linguistics
`barend.beekhuizen@utoronto.ca`

Abstract

This paper introduces VORM, an unsupervised morphological segmentation system, leveraging translation data to infer highly accurate morphological transformations, including less-frequently modeled processes such as infixation and reduplication. The system is evaluated on standard benchmark data and a novel, typologically diverse, dataset of 37 languages. Model performance is competitive and sometimes superior on canonical segmentation, but more limited on surface segmentation.

1 Introduction

While supervised neural models achieve near-ceiling performance on morphological segmentation (Batsuren et al., 2022), unsupervised systems leave ample room for improvement, despite recent progress (Virpioja et al., 2013; Narasimhan et al., 2015; Eskander et al., 2020; Xu et al., 2020). Supervised techniques can furthermore only be used for several dozen languages, whereas corpus data and word lists are available for many more. Progress on unsupervised learning is thus desirable to improve the cross-linguistic scope of morphological segmentation. The downstream benefit of morphological segmentation for training language models has been debated (Sälevä and Lignos, 2023), but morphological segmentation has further applications in comparative linguistics: for instance, to study patterns in massively parallel corpora (Liu et al., 2023), or to support semi-automated interlinear-glossing methods (McMillan-Major, 2020).

Contribution #1 of this paper is an unsupervised morphological segmentation system that leverages parallel translation data and best-first heuristics inspired by Lignos (2010) to constrain the hypothesis space. This allows it to accurately infer a broader array of morphological processes (infixation, reduplication). The system outperforms, for metrics that reflect canonical than surface segmentation, state-of-the-art unsupervised morphological models for many languages.

With those linguistic goals in mind, evaluation on a more diverse set of languages is further desirable. Existing benchmark datasets reflect only a small part of the diversity in morphological typology, with notable absences of extremely common processes like reduplication (Todd et al., 2022). Furthermore, all languages come from the Eurasian continent, thus reflecting an areally narrow set of languages. **Contribution #2** of this paper is to present a method of using a corpus of interlinearly-glossed fieldwork data in 37 typologically and areally more diverse languages (Seifart et al., 2024) to generate (both supervised and unsupervised) training data as well as evaluation data with a reproducible training/development/test split.

Materials for the project are at `https://github.com/dnrb/vorm`. After further introducing the backgrounds to this work (§2), I will introduce the novel system (§3) and the cross-linguistic data (§4). The experimentation will be set out in §5, with its empirical results in §6.

2 Background

2.1 Unsupervised morphological segmentation

The Morfessor model (Virpioja et al., 2013) forms a baseline for unsupervised morphological segmentation. It leverages word-internal statistical patterns of character sequences, leading to surface segmentations of the input string. A recent, linguistically inspired, model that similarly leads to surface segmentations is Eskander et al. (2020)'s MorphAGram, which trains Adaptor Grammars (Johnson et al., 2006) on surface strings, representing segmentation as a context-free grammar parsing problem.

Other unsupervised models leverage the insight that morphological processes do not merely carve up a surface string, but transform base forms into

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 602–626
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

derived forms, that are often not just superstrings of the base form – transforming `believe` into `believing` requires dropping the `e`.[1] Modeling such processes accurately would allow us to represent the canonical segmentation (Kann et al., 2016) of a surface string, i.e., recognizing that `believe` in the (surface segmented form) `believ+ing` contains the same canonical morpheme as `believe+s`.

An early exponent of this class of models is Morsel (Lignos, 2010), which uses a best-first heuristic that maximizes the data coverage of the inferred transformations, leading to derivations consisting of chains of transformations. A similar model pair, leveraging more global optimization over the search space of transformations, is Morphochains (Narasimhan et al., 2015) and Morphoforests (Luo et al., 2017). Like Morphoforests, ParaMA2 (Xu et al., 2020) explicitly considers paradigms, groups of transformations that co-occur as a further building block to their model, on top of using the idea that transformations form chains.

Here, I adopt many of the premises of the cited works: heuristic search, constrained by using word pairs and paradigms, and representing morphological processes as transformations.

2.2 Leveraging translations

Parallel translation data has, in several domains, been proven to help guide (otherwise) unsupervised models towards the right regions of the hypothesis space. Most pertinently, Rice et al. (2024) use translations of a target language to a reference language to provide an additional semantic signal in a supervised system, in similar ways to Narasimhan et al. (2015) and Schone and Jurafsky (2001), to determine morphological segmentation: formally overlapping words in the target language translating to the same or semantically similar words in the reference language are thus more likely to be segmented similarly.

Beyond morphology, translation data has been used to project structure of a better-resourced reference language to a target language – examples are PoS tagging and grammatical structure (Johannsen et al., 2016). Word-sense disambiguation has been shown to benefit from using translation data, given that distinct senses often translate differently (Apidianaki, 2008; Hauer and Kondrak, 2023). Shared between all cases is the idea that a reference language provides insight in the latent structure (semantic distinctions, grammatical relations, shared morphological material) of the target language, either through the projection of that structure or through the variation in the patterns of translation themselves. My approach leverages this latter type of signal.

2.3 Morphological typology

When we approach unsupervised morphological segmentation as a task of being able to induce *for any language* the (canonical or superficial) morphological segments without having access to the correct segments to train on, it is paramount to consider the variation in morphological processes across languages. A typologically-oriented overview of morphology is Haspelmath and Sims (2010), who draw on the distinction between free morphemes (which can occur as a word by themselves) and bound morphemes (which cannot) to list the following basic processes:

First, **affixation** involves concatenating bound morphemes to a free morpheme, such as `believe` + `-ing`. This includes infixation, whereby a bound morpheme is located inside the free morpheme – such as the Tagalog ‘agent trigger’ morpheme `-um-` forming `s-um-alat` ‘wrote’ out of `salat` ‘write’. Next, **compounding** involves concatenating two or more free morphemes, like `boathouse` from `boat` and `house`. Third, **reduplication** means reproducing a part of a free morpheme on either end of that morpheme – marginal in English (e.g., `house house` ‘a real house’), but widely productive in other languages, e.g. `duhp` ‘dive’ → `du-duhp` ‘be diving’ (Ponapean). Fourth, **base modification** involves changing the string ‘inside of’ the free morpheme, e.g. English ablaut (`gave` as the past tense of `give`) or stem-internal gemination as the morphological causative in Standard Arabic (`waqafa` ‘stop (intransitive)’ → `waqqafa` ‘stop (transitive)’). Finally, in **conversion** the form is unaltered but the grammatical category changes, e.g., English `hammer` can be used as a noun or verb.

Given this diversity, the focus on non-reduplicative affixation alone is narrow. Reduplication is, for instance, extremely common: over 80% of languages have some form of it (Rubino, 2013). A smaller set of languages has stem-internal modifications such as ablaut or tone change (Bickel and Nichols, 2013) – Yu (2007) finds infixation in 111 languages from 26 language families.

Surface segmentation models such as Morfessor and MorphAGram inherently rule out infixation

---

[1] Character strings are represented throughout with the `typewriter` font.

w_r	c_t	m
cảm	`$danke$`	`danke`
cảm	`fuehl`	`bauchgefuehl ehrgefuehl fuehl fuehle fuehlen fuehlst fuehlt fuehlte fuehlten gefuehl gefuehle gefuehlen gefuehllos (40 more)`

Table 1: Examples of extracted morphological families. Orthography follows the Morphochallenge 2010 format.

and base modification, and typically do not identify reduplication as distinct from regular affixation (but see Todd et al., 2022). Most models of canonical segmentation do not consider processes of reduplication and base modification, with notable exceptions being ParaMA2 (Xu et al., 2020). The present work develops this line of research.

3 The VORM model

The proposed model, VORM ('Vertaling Ondersteunt Redelijke Morfologie' – Dutch for 'Translation supports reasonable morphology') is a heuristic system that leverages translation equivalency in a reference corpus to find an initial set of morphological transformations, which it then applies more broadly. The model consists of three steps: Determining potential morphological families **[S1]**, which guide the learning of productive morphological transformations **[S2]**. Next, the learned transformations are applied beyond the potential morphological families by propagating the inferred transformations to the full vocabulary **[S3]**. Figure 1 presents a simplified illustration of the model to follow along with the technical introduction.

3.1 S1: Determining morphological families

One common challenge in unsupervised systems that use word pairs (Narasimhan et al., 2015; Xu et al., 2020) is to avoid oversegmentation. Recurrent phonotactic or orthographic patterns may lead to the induction of spurious transformations. Narasimhan et al. (2015) use distributional semantic information to nudge the model away from unrelated pairs and towards related pairs, building on the insight of Schone and Jurafsky (2001) that distributional semantic representations are often similar for morphological variants. Here, I propose to use another way to constrain the comparison, namely bitexts and word alignments.

The general procedure is as follows: we consider a bitext B of translations between a target language t and a reference language r, defined as $B = [\langle u_r^1, u_t^1\rangle, \langle u_r^2, u_t^2\rangle, \dots \langle u_r^n, u_t^n\rangle]$, meaning that B consists of an ordered list of paired translation-equivalent utterances $\langle u_r, u_t\rangle$. Let further the utterances $u_l^1 \dots u_l^n$ for a language l be made up of words from some vocabulary V_l.

The goal is to retrieve sets of word types in t that are likely morphologically related to each other, to feed into the next step. We call such a set a 'morphological family' (cf. Nagy et al., 1989), denoted $m \in M$, where M is a set of morphological families. Several functions could be defined mapping the bitext B onto a set of morphological families M. Word alignment models are a sensible candidate, except for the fact that morphologically rich target languages have a long tail of morphologically complex words which risk not getting accurately aligned, as indeed found by Beekhuizen (2025).

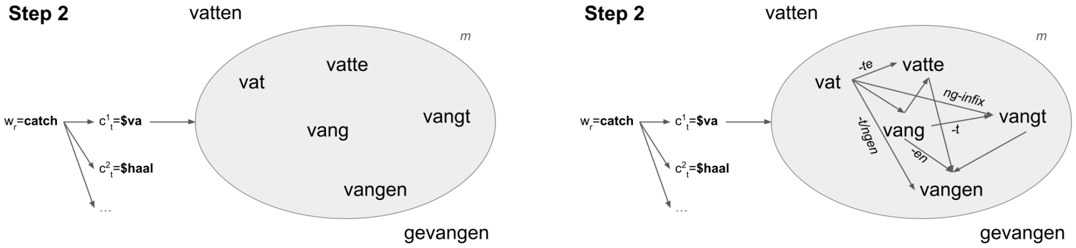
Instead, I use here the forward step of the Conceptualizer model of Liu et al. (2023), which, given a seed word w_r in the reference language r, iteratively finds character substrings $[c_t^1, c_t^2, \dots c_t^n]$ of words in t whose distribution across the utterances in B is statistically most strongly associated with the distribution of w_r. Each such substring c_t defines a morphological family m as all word types $w_t^1, w_t^2, \dots w_t^n$ that (1) contain c_t as a substring, and (2) occur in an utterance u_t^i whose aligned counterpart in r, u_r^i, contains the seed word w_r.

Table 1 presents examples of morphological families, using the seed language (Vietnamese) and corpora introduced below. Vietnamese cảm 'feel' has two c_t: `$danke$` ($ = word boundary) and `fuehl`. The morphological family of `$danke$` definitionally only contains `danke` itself, whereas `fuehl` matches many (related) words in the bitext in which it co-occurs with cảm. Figure 1a presents a morphological family found for an English-to-Dutch mapping, used here as our guiding example.

3.2 S2: Learning productive transformations

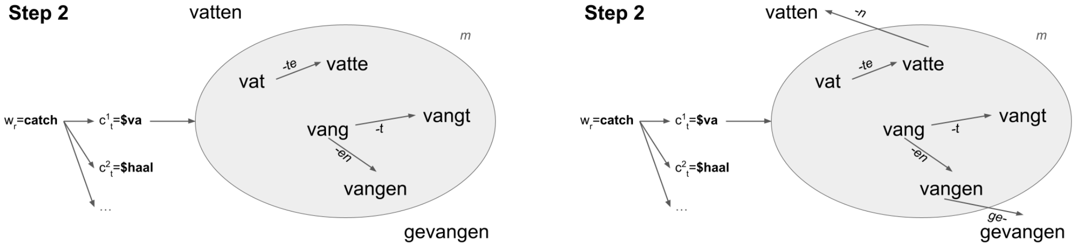
The morphological families are next used to learn productive transformations in Step 2. This procedure closely follows Morsel (Lignos, 2010). This step starts with initializing a set F of candidate transformations $f_1, f_2, \dots, f_n$. The procedure iterates over all $m \in M$. For each m, each possible pair $\langle w_t^i, w_t^j\rangle$ in m is considered. All transformations from a set of allowed transformations F_{all} that transform w_t^i into w_t^j are added to F.

F_{all} is defined to represent the typological diversity of morphological processes. The following are



(a) Step 1: Determining morphological families.

(b) Step 2: Retrieving all possible pairwise transformations.



(c) Step 2: Best-first pass determining derivations.

(d) Step 3: Expanding the transformations.

Figure 1: Simplified illustration of the 3 steps of VORM, given English (reference) and Dutch (target).

the allowed types of transformations on the right edge of the string (mirrored transformations are defined for the left edge):

Suffixation: add characters to the right edge of w_t^i so that the result is w_t^j. For instance: `belief`–`beliefs` is modeled by `-s` suffixation;

Suffixation with assimilation: remove 1 or 2 characters from the right edge of w_t^i and then add any string of characters to the (new) right edge, so that the result is w_t^j: `believe`–`believing` is modeled by `-e/ing` suffixation;

Full right reduplication; a string of length n on the right edge of w_t^i is suffixed to w_t^i to form w_t^j: The Fanbyak pair `ini`–`inini` 'to shoot' is modeled by full right reduplication of ∼`ni` (with tildes representing reduplication);

Partial-V right reduplication; all strings of one or more vowels[2] in w_t^i and w_t^j are replaced by a wildcard symbol `@`, forming the new strings $w_t^{i'}$ and $w_t^{j'}$. Next, a string s of the length n on the right edge of $w_t^{i'}$ is suffixed to $w_t^{i'}$ to form $w_t^{j'}$: Gorwaa `guus`–`guusas` are modeled this way, reduplicating the final consonant `s`, preceded by `a`.

Partial-C right reduplication; all strings of one or more consonants in w_t^i and w_t^j are replaced by the rightmost consonant in the string, forming the new strings $w_t^{i'}$ and $w_t^{j'}$. Next, a string s of the length n on the right edge of $w_t^{i'}$ is suffixed to $w_t^{i'}$ to form $w_t^{j'}$. Partial-C left reduplication is more common: Pangasinan (Rubino, 2001) transforms `plato` 'plate' into `paplato` 'plates' by taking the leftmost single consonant and vowel of a string and adding them to the left edge of that string.

Right infixation; for a pair of words w_t^i and w_t^j, removing a string s^i of length n from an anchor a in w_t^i results in a new string $w_t^{i'}$, and removing a string s^j of length m from the same anchor a in w_t^j results in a string $w_t^{j'}$. If $w_t^{i'}$ is identical to $w_t^{j'}$, the pair of words is modeled by a-anchored right infixation. Anchors are structural positions in the orthographic string constraining where the infix is combined (Yu, 2007), and I use 4 here: before vs. after the last consonant cluster, and before vs. after the last vowel cluster. English `give`-`gave` are modeled by replacing s^i = `i` for s^j = `a`, given that $w_t^{i'}$=$w_t^{j'}$=`gve`, anchored on a = before-last-consonant-cluster.

Figure 1b illustrates the set of transformations (labels on the directed edges) for the guiding example: the morphological family `$va` reflects two 'real' lemmas: `vang` 'catch [someone]' and `vat` 'catch [a disease]'. Not 'knowing' this, the model tries all possible transformations (as defined below) between any pair of word forms, such as `-te` suffixation between `vat` and `vatte`, but also (incorrectly) `ng`-infixation between `vat` and `vangt`.

Next, a best-first heuristic extracts a set of productive transformations $F_p \subseteq F$. The intuition here is that a productive morphological transformation is one that models many word pairs. Let P be the set of all word pairs $\langle w_t^i, w_t^j \rangle$ such that there

---

[2] Vowels are characters that, when stripped of diacritics, are one of {`a`,`e`,`i`,`o`,`u`,`y`}. Any other character is a consonant.

is at least one morphological family m for which $w_t^i \in m \wedge w_t^j \in m$, and P_f all such word pairs modeled by a transformation f. We then define the best transformation $f_{\text{best}} = \arg\max_f |P_f|$.[3] Once f_{best} is found, the word pairs in $P_{f_{\text{best}}}$ are removed from P, as are all other word pairs whose second word is modeled by f_{best}. The procedure is repeated until $|P_{f_{\text{best}}}| < \theta_f$, where θ_f is a pre-set threshold.

Figure 1c illustrates a possible resulting state in our example: common suffixes like `-t` and `-en` are extracted to form derivations between `vang` and `vangt` or `vangen`, while transformations with fewer instances, such as `-t/ngen` modeling the transformation from `vat` to `vangen`, are eliminated at this stage.

The derivations found through the best-first heuristic afford two sources of constraints on the application of F_p in the full vocabulary in the next Step. First, derivations form **chains**: `bookings` may have been derived from `booking` with `-s` suffixation, after which `booking` was derived from `book` through `-ing` suffixation. We denote the chain or derivation d as ⟨`-ing`, `-s`⟩, and we collect all attested chains of transformations. Secondly, chains co-occur with other chains – this can similarly help prevent oversegmentation in ways set out below. For now, we define a pair of chains of transformations d_i, d_j to **co-occur** if there is at least one base form that both models some w_i through d_i and some other w_j through d_j.

An additional procedure allows us to find **compounds**, using the morphological families. We do so by inferring a set of compound templates, strings of n elements. The template consists of $n-1$ fixed elements, and a blank spot where another word $w_t \in V_t$ can go. We find the set of **reliable compound templates** by iterating over all $m \in M$. For each word $w \in m$, we find all of its exhaustive splits w^i, w^j for which $w^i \in V_t \wedge w^j \in V_t$ and $w^i \in m \vee w^j \in m$. The latter constraint provides evidence that this is indeed a compound. For example, `bauchgefuehl` in Table 1 yields two potential compound patterns ⟨`bauch + _`⟩ and ⟨`_ + gefuehl`⟩, as both `bauch` 'belly' $\in V_f$ and `gefuehl` $\in V_f$, with the latter moreover being part of m as well (as can be seen in the table). If a pair w^i, w^j is found that forms a reliable compound template, we recursively apply the procedure to each element of the pair to see if further splits can be found. The count of the reliable compound templates is tracked across M, and all reliable compound templates with a frequency of θ_c or greater are kept to constrain compounding in Step 3.

3.3 S3: Propagation to the full word list

The derivations obtained in Step 2 are typically accurate, but only capture a small part of a language's vocabulary. First, not all morphologically related words in the bitext are found in the same morphological family m (such as Dutch `gevangen`, the past participle of `vang` in Figure 1d), but perhaps more importantly, we would like the unsupervised model to be able to generalize beyond the bitext itself. As such, Step 3 models the propagation of the productive transformations F_p, constrained by the set of chains and chain co-occurrences, to a wordlist L, where L may consist of all words in B, or some external source.

First, for each word $w \in L$, all transformations chains that can apply to it are extracted and added to a set of potential analyses $A(w)$ of w. A chain $d = \langle f_1, f_2, \ldots, f_n \rangle$ is applicable to a word w if, for every transformation f, a new string w' can be derived by removing the string added by f from the previously derived string w, where new strings do not have to be in V_t. The resulting new string after successfully applying d to w is denoted s for stem, and is added to a list of potential stems S.

Every stem $s \in S$ now defines a set of words $D(s) = \{w_i, \ldots, w_n\}$, each of which derives s through the application of a chain d. However, some s with very large $D(s)$ did not reflect coherent morphologically related groups of words. For that reason, we impose a further constraint, such that every derivational chain d modeling the relation between a word $w \in D(s)$ and s has to be found to co-occur, as defined in Step 2, with the derivational chains of at least half the other words in $D(s)$. If this is not the case, the word whose derivation co-occurs with the fewest derivations of the other words of $D(s)$ is removed from $D(s)$. This procedure is repeated until the set consists of one member, or the derivations of all words in $D(s)$ co-occur with at least half the other words in $D(s)$.

The central mechanism of this step is a **best first pass**, similar to Step 2, except the model now iteratively finds the stem s_{best} that models the largest $D(s)$ (with ties broken by stem length, preferring shorter stems). Once found, all words in $D(s_{\text{best}})$ are removed from $D(s')$ for all stems $s' \in S$, and a new s_{best} is determined. Figure 1d illustrates: the

---

[3] Ties are broken first by morphological type, where the ordering given above is followed, then by affix length (longer affixes are preferred).

words `vatten` and `gevangen` are not part of the morphological family but can be modeled with productive transforms that form attested chains from words that are in the morphological family.

After this pass is done, compounds are extracted over all extracted s_{best} by applying the **reliable compound templates** from Step 2. If the substring s filling the blank is a word in V_t, compounding applies, and the new derivation has more than one stem (potentially each with their own derivations).

4 DoReCo-morph: crosslinguistic data

The representational potential of Vorm, including reduplication and infixation, exceeds the set of morphological phenomena present in the datasets typically used. Reduplication and infixation are absent from widely used benchmark sets such as Morphochallenge 2010 (Kurimo et al., 2010). A corpus that can be fill this gap is DoReCo (`https://doreco.huma-num.fr/`; Seifart et al., 2024), consisting of collections of transcribed fieldwork materials in 52 languages. Much of these materials have interlinear glosses, exemplified in Table 3, where for each word, the morphological analysis is given. Such data allow us to derive a list of words with their morphological analyses, which in turn can be used to train (un)supervised morphological segmentation systems and evaluate them.

The Supplemental Materials for this paper contain a script for deterministically transforming the corpus data into a dataset in the same format as the Morphochallenge data, with word types linked to their canonical and surface segmentation(s). In particular, the unique words (the **w** layer in Table 3) are linked to all their morphological analyses, represented as combinations of the morphemes (**m**) and the glosses (**g**). An analysis of Savosavo `ghavilighue` would thus be: '`ghavi`:paddle `-li`:3SG.M.O `-ghu`:NMLZ `=e`:EMPH'. Some preprocessing to normalize orthography and glossing was applied.

These data can be readily used for computational morphology (and perhaps other tasks such as interlinear gloss induction, cf. McMillan-Major, 2020). The script also generates a train/development/test split over the data to facilitate experimentation. While the derived data cannot be published under the corpus licence, their generation is exactly reproducible. The datasets used, along with relevant statistics on the derived data, are presented in Table 2 in the Appendices. This table also gives the citation for each individual language, required as part of the user agreement of the corpus.

Morphological profiles of the 37 languages are presented in Appendix A (alongside similar profiles for the MC10 and MC22 data, for comparison). The average number of morphemes ranges between 1.17 and 3.26 per language in the DoReCo-Morph dataset, representing a broad variety of morphological complexity. Moreover, for all languages, there is at least some difference between the canonical forms and the surface strings (cf. Figure 5), suggesting that more than the mere placement of surface string boundaries is necessary to adequately represent the morphological structure of most languages. While little evidence of (the annotation of) infixation or base modification was found among the languages, reduplication is extensively represented in the corpus: a majority (22/37) of languages display some form of reduplication, with some languages having over 10% of their word types display reduplication. This underscores the point of Todd et al. (2022) that being able to represent reduplication is necessary for a truly multilingual unsupervised morphological model.

5 Evaluation

Evaluation data First, Vorm is compared with other models on two benchmarks: Morphochallenge 2010 (MC10; Kurimo et al., 2010), with gold data for English, Finnish, Turkish, and German canonical and surface (for all but German) segmentation, and the SIGMORPHON 2022 task on surface segmentation (SGM22; Batsuren et al., 2022) for eight languages. Next, we consider the novel DoReCo-morph dataset of 37 languages.

Evaluation metrics The standard metrics were applied. First, EMMA-2 (Virpioja et al., 2011) measures the precision and recall between the gold (canonical) segmentation and the predicted segmentation by inferring mappings between gold and predicted morpheme labels that optimize Precision and Recall, thus solving the problem of potentially differing morpheme labels. It is applied to both datasets with canonical forms: MC10 and DoReCo-morph. Second, Boundary Precision and Recall (BPR; Batsuren et al., 2022) measures the F1 score of the proportion of predicted boundaries found in the gold data (precision) and conversely the proportion of gold data boundaries predicted (recall) for all datasets. While commonly

language (glottocode; family; area; reference)
Yali (apah1238; Nuclear Trans New Guinea; PNS; Riesberg, 2024)
Arapaho (arap1274; Algic; NAM; Cowell, 2024)
Baïnounk Gubëeher (bain1259; Atlantic-Congo; AFR; Cobbinah, 2024)
Beja (beja1238; Afro-Asiatic; AFR; Vanhove, 2024)
Bora (bora1263; Boran; SAM; Seifart, 2024)
Cabécar (cabe1245; Chibchan; NAM; Quesada et al., 2024)
Cashinahua (cash1254; Pano-Tacanan; SAM; Reiter, 2024)
Dolgan (dolg1241; Turkic; ERS; Däbritz et al., 2024)
Evenki (even1259; Tungusic; ERS; Kazakevich and Klyachko, 2024)
Goemai (goem1240; Afro-Asiatic; AFR; Hellwig, 2024)
Gorwaa (goro1270; Afro-Asiatic; AFR; Harvey, 2024)
Hoocak (hoch1243; Siouan; NAM; Hartmann, 2024)
Jahai (jeha1242; Austroasiatic; ERS; Burenhult, 2024)
Jejuan (jeju1234; Koreanic; ERS; Kim, 2024)
Kakabe (kaka1265; Mande; AFR; Vydrina, 2024)
Kamas (kama1351; Uralic; ERS; Gusev et al., 2024)
Komnzo (komn1238; Yam; PNS; Döhler, 2024)
Movima (movi1243; Isolate; SAM; Haude, 2024)
Dalabon (ngal1292; Gunwinyguan; AUS; Ponsonnet, 2024)
language (glottocode; family; area; reference)
Nisvai (nisv1234; Austronesian; PNS; Aznar, 2024)
N——ng (nngg1234; Tuu; AFR; Güldemann et al., 2024)
Northern Kurdish (nort2641; Indo-European; ERS; Haig et al., 2024)
Northern Alta (nort2875; Austronesian; PNS; Garcia-Laguia, 2024)
Fanbyak (orko1234; Austronesian; PNS; Franjieh, 2024)
Pnar (pnar1238; Austroasiatic; ERS; Ring, 2024)
Daakie (port1286; Austronesian; PNS; Krifka, 2024)
Ruuli (ruul1235; Atlantic-Congo; AFR; Witzlack-Makarevich et al., 2024)
Sanzhi Dargwa (sanz1248; Nakh-Daghestanian; ERS; Forker and Schiborr, 2024)
Savosavo (savo1255; Isolate; PNS; Wegener, 2024)
Nafsan (sout2856; Austronesian; PNS; Thieberger, 2024)
Sümi (sumi1235; Sino-Tibetan; ERS; Teo, 2024)
Tabasaran (taba1259; Nakh-Daghestanian; ERS; Bogomolova et al., 2024)
Teop (teop1238; Austronesian; PNS; Mosel, 2024)
Texistepec Popoluca (texi1237; Mixe-Zoque; NAM; Wichmann, 2024)
Mojeño Trinitario (trin1278; Arawakan; SAM; Rose, 2024)
Urum (urum1249; Turkic; ERS; Skopeteas et al., 2024)
Vera'a (vera1241; Austronesian; PNS; Schnell, 2024)

Table 2: Languages in the DoReCo-morph dataset. The macroareas are: PNS = Papunesia, NAM = North America, SAM = South America, AFR = Africa, ERS = Eurasia, AUS = Australia.

w	melo	bo	lo	ghavilighue.
m	melo	bo	lo	ghavi -li -ghu =e
g	tuna	go	3SG.M	paddle -3SG.M.O -NMLZ =EMPH
f	"he went and fished bonito with it."			

Table 3: Interlinear Gloss; Savosavo (Wegener, 2024)

used, it is a less linguistically insightful metric, as (per Figure 5) non-identity between the canonical morphemes and the surface string is crosslinguistically extremely common.

Training data The bitexts used for MC10 and SGM22 were (up to) a million words of bitext from Opus2018 (Lison and Tiedemann, 2016) subtitles from `www.opensubtitles.org/`. Vietnamese was chosen as the reference language as it has little morphology. Bitexts for German and Turkish were orthographically normalized with the test data. For the DoReCo-morph experiment, bitexts were generated from the corpora, using the **w** and **f** layers (cf. Table 3). Free translations were mostly in English, with some in Malay, Spanish, and others.

Comparison models For the MC10 and SGM22, I compare Vorm against published results, but add Morfessor2 (Virpioja et al., 2013) to the latter as an unsupervised baseline. For DoReCo-morph, I run Morfessor2, ParaMA2 (Xu et al., 2020), and MorphAGram (Eskander et al., 2020) (in the language-independent setting) as unsupervised models, and Chipmunk (Cotterell et al., 2015), as a supervised model. The unsupervised models were trained on the full wordlists, and Chipmunk on the training split (48% of the data), and were tested on the test split (40% of the data).

Tuning Models were tuned on each dataset individually, using the standard splits of MC10 and SGM22, and the proposed split (12% of the data of each language) for DoReCo-morph. To better understand the performance of the Vorm model, an ablation experiment was furthermore run, leaving out Step 1 ('-S1') by instead having one single morphological family containing all vocabulary items, not extracting compounds ('-C'), and leaving out Step 3 ('-S3'). As the optimal hyperparameters for these settings may differ from the unablated version of Vorm, tuning was done on each individual ablation variant. Details and results (hyperparameter settings and accuracy metrics) of the tuning for all models and ablation variants can be found in Appendix B. Below, I will report test data results on the best-tuned model per model/ablation variant.

6 Results

6.1 Results by dataset

MorphoChallenge 2010 results. Table 4 presents the results for MC10. First, we focus on the metric for canonical segmentation, EMMA-2. Across the four languages, Vorm has the highest average F_1 score at 90.0. For the individual languages, we find that Vorm presents a substantial improvement over MorphAGram and Morfessor for Finnish, German, and Turkish, but not for English, where MorphAGram outperforms Vorm. Considering the ablation steps, we find that for some languages not using the translation equivalences in Step 1 ('-S1') or not finding compounds ('-C') improves

	EMMA-2						BPR					
	morf	AG	VORM	-C	-S1	-S3	morf	AG	VORM	-C	-S1	-S3
English	85.9	88.7	84.1	84.1	**91.2**	56.6	75.2	**80.0**	54.0	52.9	43.1	40.6
Finnish	73.4	77.7	94.9	**95.0**	92.9	46.4	62.8	**71.1**	24.8	23.6	25.1	40.6
German	80.9	85.9	93.7	**93.9**	93.2	41.3	n/a					
Turkish	61.3	69.3	**87.3**	86.0	78.9	28.3	64.6	**78.9**	24.2	23.3	23.2	19.0
avg.	75.3	80.4	**90.0**	89.7	89.0	43.1	67.5	**76.7**	34.3	33.3	30.5	24.1

Table 4: Model comparison on the development sets for Morphochallenge 2010 [MC10], comparing Morfessor (Morf) and the best MorphAGram (AG) model against VORM with ablation variants, on EMMA-2 and BPR F_1 scores. The best result per language and per metric is boldfaced.

	DeepSPIN-3	morf	VORM	-C	S1	S3
Czech	93.84	**28.71**	28.18	27.12	25.75	6.34
English	93.63	**49.90**	41.85	33.17	40.63	10.80
French	95.73	**23.63**	20.33	21.67	20.31	3.99
Hungarian	98.72	**34.47**	34.43	33.67	32.44	32.44
Italian	97.43	11.84	11.35	**12.50**	11.40	2.39
Latin	99.39	**17.77**	12.92	13.25	12.98	4.10
Russian	99.35	11.46	15.60	**18.56**	14.42	0.65
Spanish	99.04	9.23	17.99	**19.06**	17.96	1.25
avg.	97.29	**23.38**	20.82	20.67	20.07	7.41

Table 5: Model comparison on the tests sets for the SIGMORPHON 2022 challenge comparing DeepSPIN-3 (supervised) and Morfessor2 against VORM and its ablation variants on the Batsuren et al. (2022) evaluation measure. The best unsupervised result per language is boldfaced; the best result overall underlined.

the quality of the model, suggesting that further development of these components might be necessary. Removing the extension to the full vocabulary ('-S3') is, however, consistently detrimental.

On the surface segmentation measure of BPR, VORM is substantially outperformed by Morfessor and MorphAGram. This effect may be due to the differences between the metrics: EMMA-2 favours canonical morpheme identity, but does not penalize allomorphy, which is indistinguishable from undersegmentation to the model. The same undersegmentation leads to extremely low (often single digit) recall scores on the BPR measure for VORM.

SIGMORPHON 2022 results. For the SGM22, only surface segmentation is considered, using the metric provided by the task. The results are presented in Table 5. While no unsupervised model performs even close to the supervised models (here, the best-performing supervised model DeepSPIN-3, Peters and Martins, 2022, is given as a reference point), VORM without compounding ('-C') occasionally outranks Morfessor2 in its performance. This further underscores the previous observation that VORM does not excel in surface segmentation.

DoReCo-morph. Finally, Table 6 present the aggregated results for VORM and comparison models over the 37 DoReCo-morph languages, with Table 15 in the appendices presenting the EMMA-2 scores per language. For the EMMA-2 scores, unablated VORM outperforms the other unsupervised models for 20/37 languages (32 if considering the ablated variants). MorphAGram is the optimal model for 1 language. Considering average model performance, we find VORM outperforming other unsupervised models by a substantial margin, coming within a 2% range of the supervised Chipmunk model. Notably, the language VORM performs worst on still reaches an EMMA-2 score of 78.0, while Chipmunk only scores 69.9 on its worst case – with Morfessor also performing robustly at 77.1. In the ablation experiment, we find that the effects of leaving out compounding ('-C') are negligible, and that not having Step 1 in many cases *improves* performance (indeed, the worst case *without* Step 1 is slightly better than the worst case of unablated VORM). Omitting Step 3 ('-S3') in all cases leads to a substantial drop in performance.

For surface segmentation (BPR), however, the performance is more mixed: here, Chipmunk is the best overall model with a large margin, with ParaMA2 being the best model for 15 languages, and VORM for 21. Notably, leaving out Step 3 here frequently leads to an improvement for VORM, owing perhaps to the fact that these are small datasets for which the complete vocabulary is captured in Step 1 (as opposed to the MC10 data, where the corpus data only contained a subset of the test data), and that as such extension beyond the morphological families leads to more Precision errors than improvement of Recall.

6.2 Discussion

On the whole, the results suggest that VORM is a competitive model of **canonical segmentation**. On

	EMMA-2								BPR							
	chip	morf	para	AG	vorm	-S1	-C	-S3	chip	morf	para	AG	vorm	-S1	-C	-S3
max?		4		1	**20**	11		1		1	**15**		5	1		**15**
avg.	91.4	86.1	80.3	84.6	**89.9**	88.6	89.9	69.1	86.9	56.9	57.0	34.3	58.8	46.6	58.2	**60.4**
worst	69.9	77.1	69.7	71.3	78.0	**78.3**	78.0	35.3	65.6	31.0	**35.7**	13.3	32.9	29.4	32.7	29.6

Table 6: Aggregated EMMA-2 & BPR F_1 scores for the DoReCo-morph dataset for [chip]munk (supervised), [Morf]essor2, [Para]MA2 , Morph[AG]ram, and Vorm with its ablation variants. Best unsupervised results in bold; best overall results underlined.

word	`babarak`	`vivirigěm`
gold	`ba∼`:RED∼ `bara`:long `-k`:TAM1	`vi∼`:RED∼ `virigě`:rush `-m`:TAM1
chip	`babarak`	`vivirig` + `ěm`
morf2	`babara` + `k`	`vivi` + `rig` + `ěm`
para	`babara` + `-k`	`vi_rig` + `-vi-` + `-em`
AG	`babara` + `k`	`vivi` + `rig` + `ěm`
vorm	`ba∼` + `bara` + `-k`	`vi∼` + `virigě` + `-m`

Table 7: Examples of reduplication in Vera'a (Schnell, 2024) and their analysis across models. Underscores mark the infix slot; tildes mark reduplicative affixes.

morphologically complex languages like Finnish and Turkish, its improvement over other unsupervised models is substantial. The ablation experiments paint a complicated picture of what leads to these results – the addition of a compounding component, and the 'narrowing' of the hypothesis space through the use of morphological families in Step 1 have only a small, and sometimes even a negative, effect. The Precision-oriented focus of the compounding component may lead to limited extraction of compounds. Step 1 may be redundant with the filtering mechanisms of Step 2: when all words are compared with each other, low-frequency transforms will be eliminated by the frequency threshold, and frequent, but spurious, transforms may be weeded out by being pre-empted by a more frequent transform in the best-first pass. However, for some languages (Turkish in MC10, several DoReCo-morph languages), the omission of Step 1 does come at a cost, suggesting that narrowing by translation equivalence is not always redundant.

On **surface segmentation** Vorm does not perform as competitively. This can be attributed to the lower Recall the model achieves here, and its focus on canonical segmentation leading to variable boundaries on the surface string. Importantly, this contrast suggests that canonical and surface segmentation are substantially different tasks.

The examples in Table 7 demonstrate the model's capacity to analyze reduplication. We see that only vorm analyses the forms correctly, both in its surface segmentations as well as in its canonical analysis, i.e., recognizing `ba∼` and `vi∼` as reduplicative morphemes. Other models either undersegment the left edge of the words, or missegment the word (paraMA, Morfessor).

None of our languages has productive base modification processes, but German has some, in nominal plurals and past tense. Given the low type frequency of such Ablaut processes, the tuned model did not learn these patterns, but a model with a lower $\theta_f = 30$, did analyze `huehnerbesitzer` 'chicken owner' correctly as `hu_hn + -e- + -er + besitz + -er` and `geldbetraege` 'sums of money' as `geldbetra_g -e- + -e`.

7 Conclusion

This paper introduces Vorm, a novel unsupervised morphological segmentation system, which uses translation-equivalency to narrow down the set of word pairs on which the inferred morphological transformations are based. Aside from affixation, the model can represent base-modifying transformations and reduplication. Generalizing models are induces through a pair of heuristic, best-first processes. In doing so, the model stands in a tradition of unsupervised morphological segmentation that does not consider very large parts of the hypothesis space (Lignos, 2010; Xu et al., 2020) in order to maintain high precision.

Further exploration on the DoReCo-morph dataset could identify more specific modeling challenges by breaking down the full dataset into linguistically interesting subsets (cases with reduplication, cases where the canonical form deviates substantially from the surface form through assimilation processes, etc.). Through such exploration, and more detailed analysis of model performance on different challenges, the landscape of what unsupervised learners have to contend with might become more clear. With this paper, I hope to have made a first move in that direction.

References

Marianna Apidianaki. 2008. Translation-oriented word sense induction based on parallel corpora. In *Language Resources and Evaluation (LREC)*.

Jocelyn Aznar. 2024. Nisvai DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtskỳ, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, et al. 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.

Barend Beekhuizen. 2025. Token-level semantic typology without a massively parallel corpus. In *The 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*.

Balthasar Bickel and Johanna Nichols. 2013. Fusion of selected inflectional formatives (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Natalia Bogomolova, Dmitry Ganenkov, and Nils Norman Schiborr. 2024. Tabasaran DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Niclas Burenhult. 2024. Jahai DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Alexander Yao Cobbinah. 2024. Baïnounk Gubëeher DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Ryan Cotterell, Thomas Mueller, Alexander Fraser, and Hinrich Schütze. 2015. Labeled morphological segmentation with semi-markov models. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 164–174.

Andrew Cowell. 2024. Arapaho DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Chris Lasse Däbritz, Nina Kudryakova, Eugénie Stapert, and Alexandre Arkhipov. 2024. Dolgan DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Christian Döhler. 2024. Komnzo DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Ramy Eskander, Francesca Callejas, Elizabeth Nichols, Judith L Klavans, and Smaranda Muresan. 2020. Morphagram, evaluation and framework for unsupervised morphological segmentation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 7112–7122.

Diana Forker and Nils Norman Schiborr. 2024. Sanzhi Dargwa DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Michael Franjieh. 2024. Fanbyak DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Alexandro Garcia-Laguia. 2024. Northern Alta DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Valentin Gusev, Tiina Klooster, Beáta Wagner-Nagy, and Alexandre Arkhipov. 2024. Kamas DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Tom Güldemann, Martina Ernszt, Sven Siegmund, and Alena Witzlack-Makarevich. 2024. Nng DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Geoff Haig, Maria Vollmer, and Hanna Thiele. 2024. Northern kurdish (kurmanji) doreco dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Iren Hartmann. 2024. Hoocak DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Andrew Harvey. 2024. Gorwaa DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Martin Haspelmath and Andrea Sims. 2010. *Understanding morphology*. Routledge.

Katharina Haude. 2024. Movima DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Bradley Hauer and Grzegorz Kondrak. 2023. One sense per translation. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 442–454.

Birgit Hellwig. 2024. Goemai DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Anders Johannsen, Željko Agić, and Anders Søgaard. 2016. Joint part-of-speech and dependency projection from multiple sources. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 561–566.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2006. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. *Advances in neural information processing systems*, 19.

Katharina Kann, Ryan Cotterell, and Hinrich Schütze. 2016. Neural morphological analysis: Encoding-decoding canonical segments. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 961–967.

Olga Kazakevich and Elena Klyachko. 2024. Evenki DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Soung-U Kim. 2024. Jejuan DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Manfred Krifka. 2024. Daakie DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden. Association for Computational Linguistics.

Constantine Lignos. 2010. Learning from unseen data. In *Proceedings of the Morpho Challenge 2010 Workshop*, pages 35–38, Helsinki, Finland. Aalto University School of Science and Technology.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yihong Liu, Haotian Ye, Leonie Weissweiler, Philipp Wicke, Renhao Pei, Robert Zangenfeind, and Hinrich Schütze. 2023. A crosslingual investigation of conceptualization in 1335 languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12969–13000.

Jiaming Luo, Karthik Narasimhan, and Regina Barzilay. 2017. Unsupervised learning of morphological forests. *Transactions of the Association for Computational Linguistics*, 5:353–364.

Angelina McMillan-Major. 2020. Automating gloss generation in interlinear glossed text. *Society for Computation in Linguistics*, 3(1).

Ulrike Mosel. 2024. Teop DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

William Nagy, Richard C Anderson, Marlene Schommer, Judith Ann Scott, and Anne C Stallman. 1989. Morphological families in the internal lexicon. *Reading Research Quarterly*, pages 262–282.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM Comput. Surv.*, 33(1):31–88.

Ben Peters and Andre F. T. Martins. 2022. Beyond characters: Subword-level morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 131–138, Seattle, Washington. Association for Computational Linguistics.

Maïa Ponsonnet. 2024. Dalabon DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Juan Diego Quesada, Stavros Skopeteas, Carolina Pasamonik, Carolin Brokmann, and Florian Fischer. 2024. Cabécar DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Sabine Reiter. 2024. Cashinahua DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Enora Rice, Ali Marashian, Luke Gessler, Alexis Palmer, and Katharina Wense. 2024. Tams: Translation-assisted morphological segmentation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6752–6765.

Sonja Riesberg. 2024. Yali (apahapsili) doreco dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Hiram Ring. 2024. Pnar DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Françoise Rose. 2024. Mojeño Trinitario DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Carl Rubino. 2001. Pangasinan. In Jane Garry and Carl Rubino, editors, *Encyclopedia of the World's Languages: Past and Present*, pages 539–542. H.W. Wilson Press, New York / Dublin.

Carl Rubino. 2013. Reduplication (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Jonne Sälevä and Constantine Lignos. 2023. What changes when you randomly choose bpe merge operations? not much. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 59–66.

Stefan Schnell. 2024. Vera'a doreco dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Patrick Schone and Dan Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Second meeting of the north american chapter of the association for computational linguistics*.

Frank Seifart. 2024. Bora DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Frank Seifart, Ludger Paschen, and Matthew Stave. 2024. *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Stavros Skopeteas, Violeta Moisidi, Nutsa Tsetereli, Johanna Lorenz, and Stefanie Schröter. 2024. Urum DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Amos Teo. 2024. Sümi DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Nick Thieberger. 2024. Nafsan (south efate) doreco dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

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 on Computational Research in Phonetics, Phonology, and Morphology*, pages 12–22.

Martine Vanhove. 2024. Beja DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

Sami Virpioja, Ville T. Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.

Alexandra Vydrina. 2024. Kakabe DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Claudia Wegener. 2024. Savosavo DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Søren Wichmann. 2024. Texistepec Popoluca DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

Alena Witzlack-Makarevich, Saudah Namyalo, Anatol Kiriggwajjo, and Zarina Molochieva. 2024. Ruuli DoReCo dataset. In Frank Seifart, Ludger Paschen, and Matthew Stave, editors, *Language Documentation Reference Corpus (DoReCo) 2.0*. Laboratoire Dynamique Du Langage (UMR5596, CNRS & Université Lyon 2), Lyon.

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.

Alan C. L. Yu. 2007. 67pivot theory and the typology. In *A Natural History of Infixation*. Oxford University Press.

A Morphological profiles of the languages

A.1 Number of morphemes

The number of morphemes, as given in the gold standard datasets is presented in Figures 2 (for DoReCo-Morph), 3 (for MC10) and 4 (for MC22).

A.2 Number of insertions and deletions

Only for the DoReCo-Morph data do we have both the surface forms and the canonical forms; for MC10, despite canonical morphemes being given, the inflectional morphemes are mostly given in a featural notation (‘+PL’, ‘+SUP’) and as such a canonical string of phonological/orthographic segments cannot be faithfully extracted. For MC22, only suface string segmentations are given.

The difference between the surface form and the canonical form was calculated by running a Wagner-Fisher algorithm (see Navarro, 2001, for the complexity of authorship attribution of this algorithm) on the two strings to compute the path of maximal string overlap, allowing only for character insertion and deletion operations. Any insertions (the canonical form has more characters than the surface form; e.g., `believing` for `believe` + `-ing`) and deletions (the surface form has more characters than the canonical form; e.g., `barring` for `bar` + `-ing`) were counted.

Figure 5 presents the number of insertions and deletion going from the surface form to the canonical form.

A.3 Prevalence of morpheme types

Per language, it was determined heuristically whether a morpheme was free, reduplicative and bound, or affixal and bound. Reduplicative morphemes are consistently tagged with a tilde (‘$\sim$’) on their left or right edge in the DoReCo-Morph data (and are absent from the MC10 data). Affixal morphemes are marked with capitalized glosses in DoReCo-Morph and either a grammatical feature-style notation starting with ‘+’ (e.g., ‘+PL’) or a grammatical category marking as ‘p’ (prefix) or ‘s’ (suffix) in the MC10 data. Morpheme types are undefined for the MC22 data.

Figure 6 presents the counts for the DoReCo-Morph data, while Figure 7 presents the counts for the MC10 data.

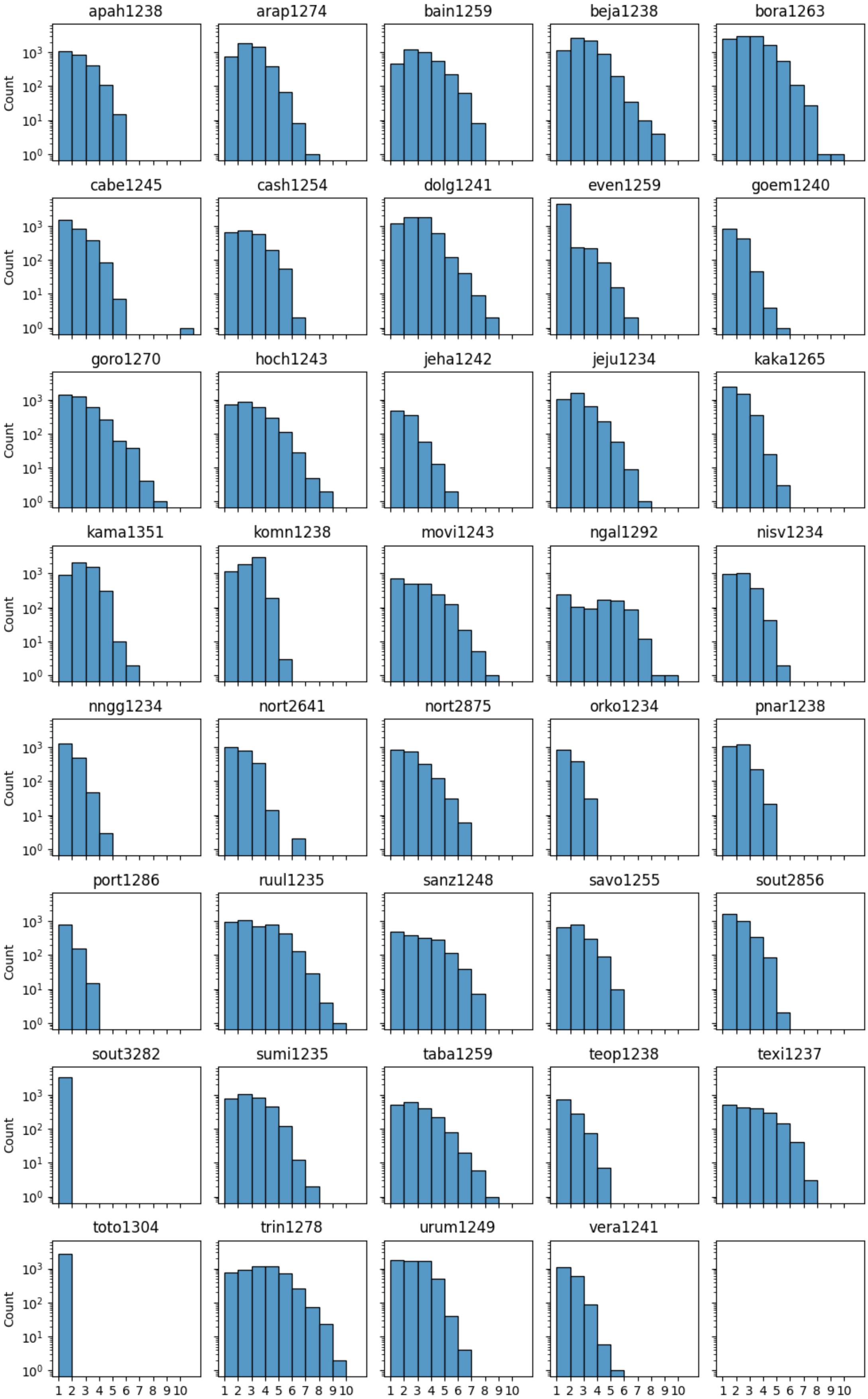


Figure 2: Histogram of the number of morphemes in the DoReCo-Morph data.

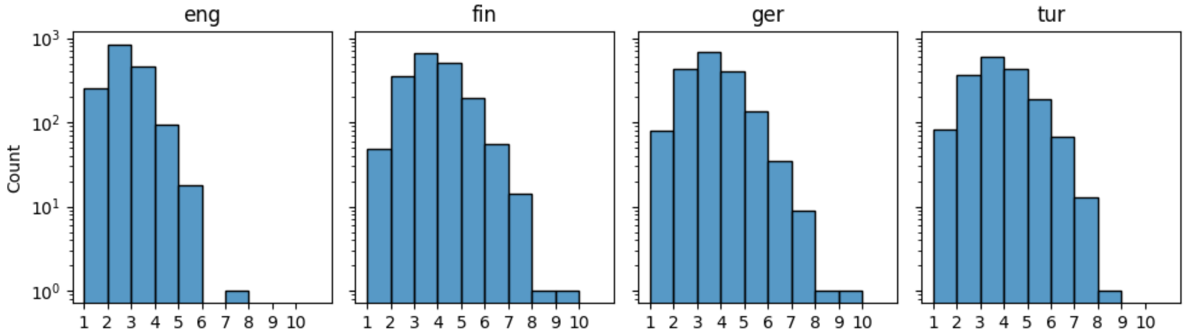


Figure 3: Histogram of the number of morphemes in the MC10 data.

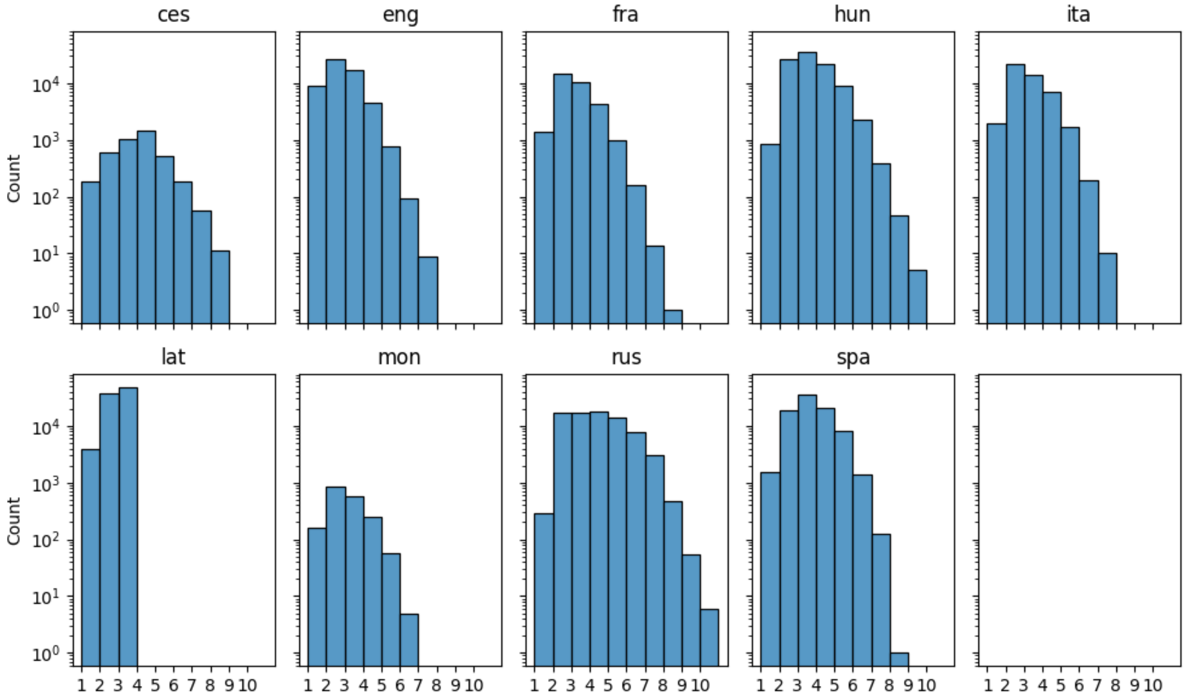


Figure 4: Histogram of the number of morphemes in the MC22 data.

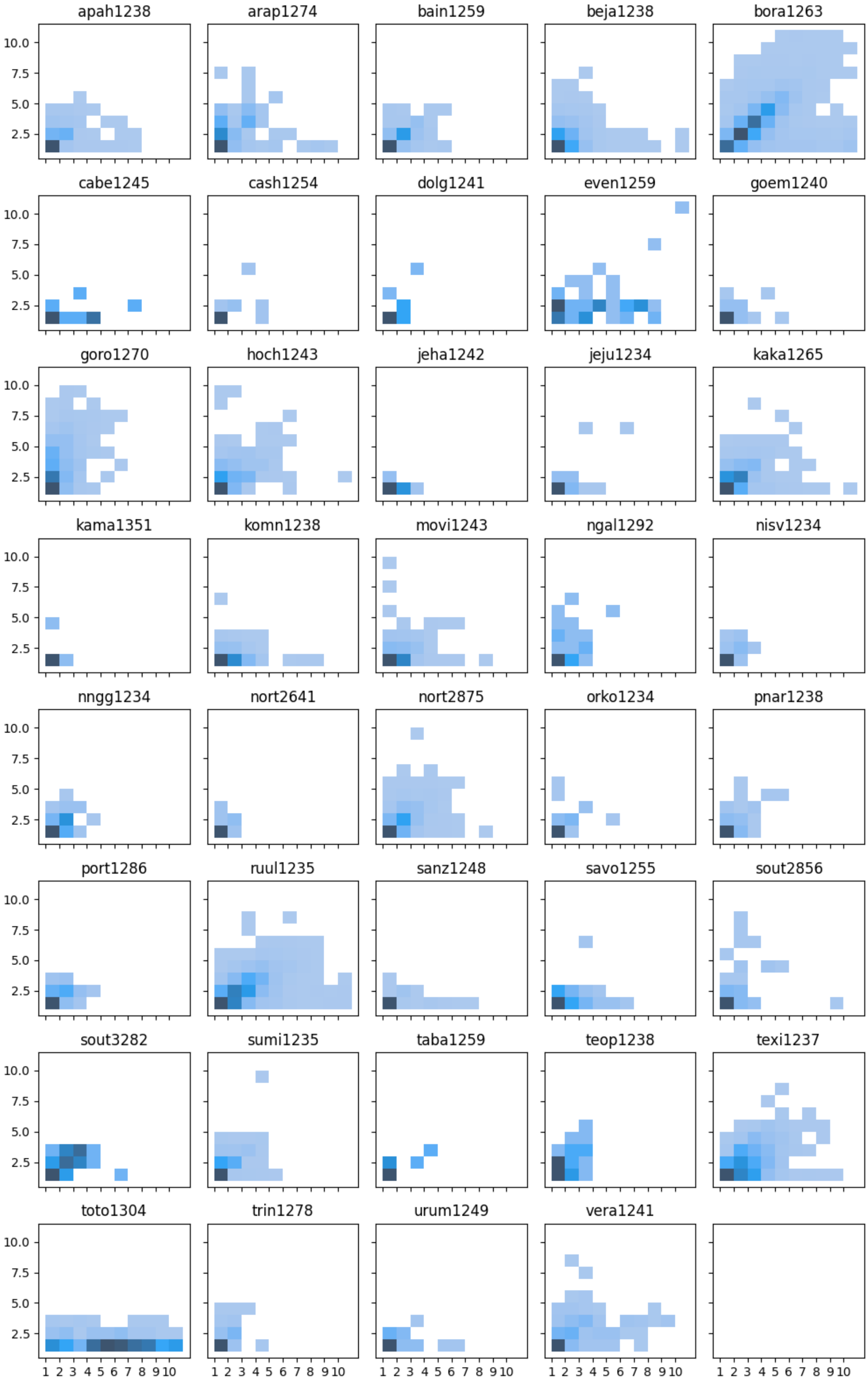


Figure 5: Heatmap of the number of insertions (rows) and deletions (columns) going from the surface form to the canonical form in the DoReCo-Morph data (darker means more instances).

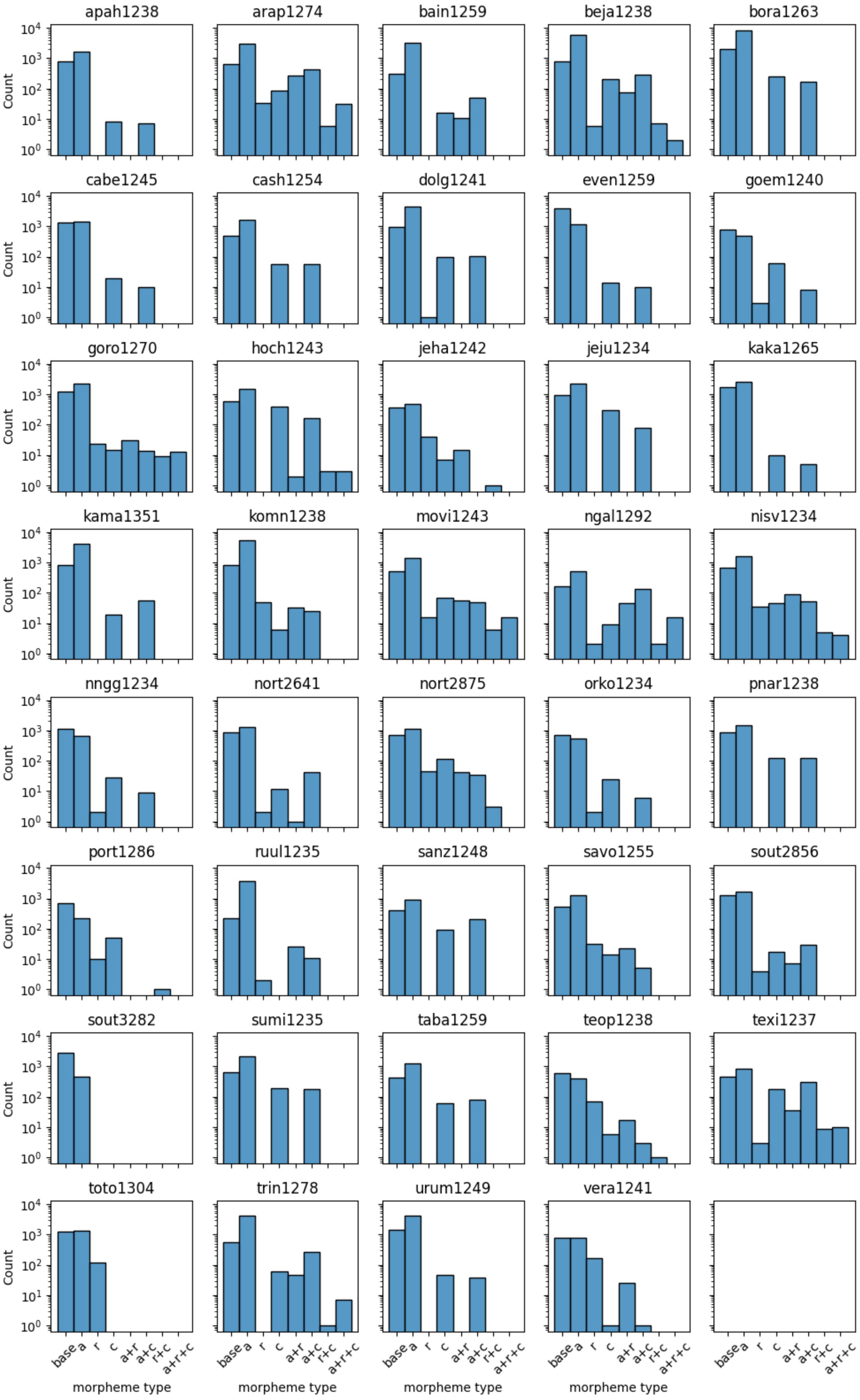


Figure 6: Histogram of the combinations of three morphological types in the lexical items of the DoReCo-Morph languages. 'base' = morphologically symplex, 'a' = affixation; 'r' = reduplication; 'c' = compounding

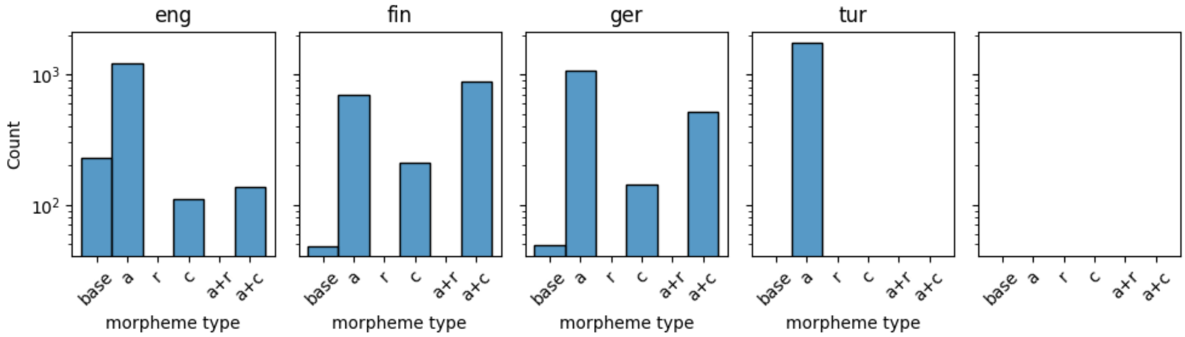


Figure 7: Histogram of the combinations of three morphological types in the lexical items of the DoReCo-Morph languages. ‘base’ = morphologically symplex, ‘a’ = affixation; ‘r’ = reduplication; ‘c’ = compounding

B Tuning experiments

The model was tuned on the development split (12% of the data for each language) in the DoReCo-morph data, the training split for MC10 and the development split for SGM22. For each task, and for each metric (EMMA-2 or BPR), the best-performing set of hyperparameters of each model (and of each ablation variant of the Vorm model) was selected.

B.1 DoReCo-Morph data

For **VORM**, the free parameters $\theta_f \in \{3, 5, 10, 20\}$ (minimum number of word pairs modeled by a transformation in Step 2) and $\theta_c \in \{1, 5, 10, 20\}$ (minimum number of compound template occurrences for it to be used in Step 3) using a grid search over the values. The ablation over model components (-CMPD: no compounding, -S1: no Step 1, i.e. comparing all of the vocabulary in Step 2, -S3: no Step 3) was done simultaneously, as optimal values for θ_f and θ_c can be expected to vary across ablation steps.

Tuning scores are given in Table 8. As the optimal parameter settings do not line up across the two measures, the parameter setting with the highest average across the two scores was selected per ablation setting. In the main text, we report on $\theta_f = 20, \theta_c = 1$ for no ablation and -CMPD, $\theta_f = 20, \theta_c = 20$, for -S1, and $\theta_f = 3, \theta_c = 1$ for -S3.

For **Morfessor**, the model was tuned on the three ways of using token counts (token counts or: 'token', no counts or: 'type', and 'log-counts'). Scores are given in Table 9. Log-counts is on average the best-performing setting.

For **ParaMA** (Xu et al., 2020), I varied the minimum stem length ($\in \{1, 3\}$) and whether the model

ablation	θ_c	θ_f	EMMA-2	BPR
	1	3	90.48	42.48
	1	5	90.60	46.57
	1	10	89.62	51.51
	1	20	85.73	**56.62**
	5	3	90.63	42.43
	5	5	90.80	46.47
	5	10	89.78	51.37
	5	20	85.82	56.42
	10	3	90.65	42.43
	10	5	**90.83**	46.46
	10	10	89.80	51.25
	10	20	85.86	56.29
	20	3	90.65	42.43
	20	5	**90.83**	46.46
	20	10	89.80	51.22
	20	20	85.88	56.15
-CMPD	n/a	3	90.65	42.43
-CMPD	n/a	5	**90.83**	46.46
-CMPD	n/a	10	89.80	51.22
-CMPD	n/a	20	85.88	**56.09**
-S1	1	3	87.25	30.02
-S1	1	5	87.88	30.21
-S1	1	10	88.52	33.21
-S1	1	20	88.71	**40.80**
-S1	5	3	87.42	30.03
-S1	5	5	88.53	30.24
-S1	5	10	89.39	33.23
-S1	5	20	89.21	40.70
-S1	10	3	87.42	30.03
-S1	10	5	88.53	30.24
-S1	10	10	89.69	33.25
-S1	10	20	89.52	40.63
-S1	20	3	87.42	30.03
-S1	20	5	88.53	30.24
-S1	20	10	89.69	33.25
-S1	20	20	**89.77**	40.52
-S3	1	3	**76.71**	**59.74**
-S3	1	5	76.48	59.53
-S3	1	10	76.17	59.06
-S3	1	20	75.55	58.42
-S3	5	3	76.70	59.70
-S3	5	5	76.48	59.45
-S3	5	10	76.17	59.00
-S3	5	20	75.55	58.39
-S3	10	3	76.70	59.69
-S3	10	5	76.47	59.45
-S3	10	10	76.16	58.93
-S3	10	20	75.56	58.33
-S3	20	3	76.70	59.69
-S3	20	5	76.47	59.44
-S3	20	10	76.17	58.92
-S3	20	20	75.55	58.27

Table 8: Average EMMA-2 and BPR scores on DoReCo-Morph tuning data for the VORM model. Best model scores per ablation (none, -CMPD, -S1, -S3) boldfaced, best overall score (per metric) underlined.

parameters	EMMA-2	BPR
token	86.6	**51.5**
type	85.6	33.2
log-counts	**88.2**	48.4

Table 9: Average EMMA-2 and BPR scores on DoReCo-Morph tuning data for the Morfessor model. Best model scores boldfaced

parameters	EMMA-2	BPR
$-$compound, min stem ≥ 1	78.7	41.5
$+$compound, min stem ≥ 1	78.7	41.5
$-$compound, min stem ≥ 3	83.3	53.5
$+$compound, min stem ≥ 3	**83.3**	**53.5**

Table 10: Average EMMA-2 and BPR scores on DoReCo-Morph tuning data for the ParaMA model. Best model scores boldfaced

parameters	EMMA-2	BPR
default	**86.7**	**23.1**
reported	86.1	22.2
vaguer-G	85.8	22.4
sharper-G	86.2	22.1

Table 11: Average EMMA-2 and BPR scores on DoReCo-Morph tuning data for the MorphAGram model. Best model scores boldfaced

tried to find compounds. Table 10 presents the results. Compounding and a minimal stem length of ≥ 3 leads to the best setting on both metrics.

For **MorphAGram** (Eskander et al., 2020), the primary two settings were the model defaults and the reported values (in which the Gamma parameters of the py-cfg model (Johnson et al., 2006) were set to pyb-gamma-s $= 10$ and pyb-gamma-c $= 0.1$. As these parameters were found to be effective before, tuning was undertaken in both directions, resetting them to the default ('vaguer-G': pyb-gamma-s$= 1$ and pyb-gamma-c$= 1$) and making them more extreme (as suggested in the py-cfg documentation: 'sharper-G': pyb-gamma-s$= 100$ and pyb-gamma-c$= 0.01$). No tuning of the Alpha and Beta parameters of the py-cfg model was done as the optimal tuned in the paper was the default. Table 11 presents the results; the default setting was consistently the optimal one.

For **Chipmunk**, no parameters were found that would lead to differences in model performance.

B.2 MC10

For the MC10 data, only novel results were generated for Vorm, with the other results being cited from other papers. The free parameters $\theta_f \in \{30, 60, 100\}$ (minimum number of word pairs modeled by a transformation in Step 2) and $\theta_c \in \{10, 30, 60\}$ (minimum number of compound template occurrences for it to be used in Step 3) were tuned using a grid search over the values. As with the DoReCo-morph data, the ablation variants were tuned separately. The results are given in Table 12. The best average parameter settings used for the test phase were $\theta_f = 100, \theta_c = 10$ for no-ablation, both metrics, $\theta_f = 100$ for -C, $\theta_f = 100, \theta_c = 100$ for -S1, EMMA-2, and $\theta_f = 100, \theta_c = 30$ for -S1, BPR, and finally $\theta_f = 30, \theta_c = 100$ for -S3, EMMA and $\theta_f = 60, \theta_c = 100$ for -S3, BPR.

B.3 MC22

For the MC22 data, the free parameters of VORM $\theta_f \in \{30, 60, 100\}$ (minimum number of word pairs modeled by a transformation in Step 2) and $\theta_c \in \{10, 30, 100\}$ (minimum number of compound template occurrences for it to be used in Step 3) were tuned using a grid search over the values. The results are given in Table 13. For EMMA-2, values of $\theta_f = 100, \theta_c = 100$ were found to be on average optimal for the no-ablation variant, -S1 and -C, and $\theta_f = 30, \theta_c = 100$ for the

			EMMA-2					BPR			
ablation	θ_c	θ_f	eng	fin	ger	tur	avg.	eng	fin	tur	avg.
	10	30	**92.93**	96.17	**95.77**	90.48	93.84	42.34	26.09	23.40	30.61
	10	60	92.59	96.26	95.37	92.84	94.26	53.11	**26.95**	22.45	34.17
	10	100	92.22	96.27	94.46	94.16	**94.28**	**55.87**	25.60	**24.05**	**35.17**
	30	30	**92.93**	96.17	**95.77**	90.48	93.84	42.34	26.09	23.40	30.61
	30	60	92.59	96.26	95.37	92.84	94.26	53.11	**26.95**	22.45	34.17
	30	100	92.22	96.27	94.46	94.16	**94.28**	**55.87**	25.60	**24.05**	**35.17**
	100	30	**92.93**	96.17	**95.77**	90.48	93.84	42.34	26.09	23.40	30.61
	100	60	92.59	96.26	95.37	92.84	94.26	53.11	**26.95**	22.45	34.17
	100	100	92.18	**96.28**	94.47	**94.21**	**94.28**	**55.87**	25.60	**24.05**	**35.17**
-C	n/a	30	**92.97**	96.26	**95.86**	90.46	93.89	42.05	25.49	23.21	30.25
-C	n/a	60	92.62	96.26	95.41	92.81	94.28	52.32	**26.14**	21.98	33.48
-C	n/a	100	92.20	**96.36**	94.49	**94.11**	**94.29**	**54.90**	24.66	**23.30**	**34.29**
-S1	10	30	93.87	90.09	91.93	82.60	89.62	34.69	25.55	20.27	26.84
-S1	10	60	**94.29**	94.14	93.55	87.01	92.25	38.83	24.50	21.05	28.13
-S1	10	100	92.90	95.37	95.13	**90.31**	93.43	**43.78**	**25.65**	23.23	**30.89**
-S1	30	30	93.87	90.09	91.93	82.60	89.62	34.69	25.55	20.27	26.84
-S1	30	60	**94.29**	94.14	93.55	87.01	92.25	38.83	24.50	21.05	28.13
-S1	30	100	92.90	95.37	95.13	**90.31**	93.43	**43.78**	**25.65**	23.23	**30.89**
-S1	100	30	93.87	90.09	91.93	82.60	89.62	34.69	25.55	20.27	26.84
-S1	100	60	92.88	94.14	93.55	87.01	91.90	39.23	24.50	21.05	28.26
-S1	100	100	92.89	**95.65**	**95.24**	90.16	**93.48**	**43.78**	25.62	**23.34**	24.48
-S3	10	30	**73.40**	**58.59**	**53.77**	43.00	**57.19**	**39.94**	**14.73**	21.05	25.24
-S3	10	60	73.36	58.58	**53.77**	42.97	57.17	39.69	14.72	**21.39**	25.27
-S3	10	100	73.07	58.58	53.76	43.02	57.11	39.24	14.37	21.12	24.91
-S3	30	30	**73.40**	**58.59**	**53.77**	43.00	**57.19**	**39.94**	**14.73**	21.05	25.24
-S3	30	60	73.36	58.58	**53.77**	42.97	57.17	39.69	14.72	**21.39**	25.27
-S3	30	100	73.07	58.58	53.76	43.02	57.11	39.24	14.37	21.12	24.91
-S3	100	30	**73.40**	**58.59**	**53.77**	43.00	**57.19**	**39.94**	**14.73**	21.05	25.24
-S3	100	60	73.36	58.58	**53.77**	42.97	57.17	39.69	14.72	**21.39**	**25.27**
-S3	100	100	73.11	58.50	53.76	**43.15**	57.13	39.24	14.37	21.06	24.89

Table 12: EMMA-2 and BPR scores on MC10 tuning data for the VORM model. Best model scores per ablation variant and per metric boldfaced

-S3 variant. For BPR, $\theta_f = 100, \theta_c = 100$ was found to be the optimal setting for no-ablation and -C, and $\theta_f = 100, \theta_c = 30$ for -S1 and -S3.

For **Morfessor**, the model was again tuned on the three ways of using token counts (token counts or: 'token', no counts or: 'type', and 'log-counts'). Scores are given in Tables 14. Across languages, the 'type' setting performed the best.

ablation	θ_c	θ_f	ces	eng	fra	hun	ita	lat	rus	spa	avg.
	10	30	27.68	33.61	**21.65**	**35.21**	12.44	**13.22**	**18.95**	**19.14**	22.74
	10	60	**30.93**	35.49	20.78	34.10	**12.47**	12.67	16.30	18.08	22.60
	10	100	28.18	**41.85**	20.33	34.43	11.35	12.92	15.60	17.99	**22.83**
	30	30	27.68	33.61	**21.65**	**35.21**	12.44	**13.22**	**18.95**	**19.14**	22.74
	30	60	**30.93**	35.49	20.78	34.10	**12.47**	12.67	16.30	18.08	22.60
	30	100	28.18	**41.85**	20.33	34.43	11.35	12.92	15.60	17.99	**22.83**
	100	30	27.68	33.61	**21.65**	**35.21**	12.44	**13.22**	**18.95**	**19.14**	22.74
	100	60	**30.93**	35.49	20.78	34.10	**12.47**	12.67	16.30	18.08	22.60
	100	100	28.18	**41.85**	20.33	32.44	11.35	12.92	15.60	17.99	22.58
-CMPD	n/a	30	27.12	33.17	**21.67**	**33.67**	**12.50**	**13.25**	**18.56**	**19.06**	**22.38**
-CMPD	n/a	60	**28.86**	34.24	20.73	32.34	12.48	12.72	15.20	17.95	21.82
-CMPD	n/a	100	25.32	**40.62**	20.34	31.77	10.50	12.89	14.42	17.96	21.73
-S1	10	30	21.55	30.41	17.12	34.06	**16.70**	**26.67**	13.19	13.19	21.61
-S1	10	60	22.92	29.22	16.74	34.67	16.19	23.42	14.82	17.37	21.92
-S1	10	100	24.77	28.24	17.07	**34.97**	16.53	19.51	**17.34**	16.05	21.81
-S1	30	30	21.55	30.41	17.12	34.06	**16.70**	**26.67**	13.19	13.19	21.61
-S1	30	60	22.92	29.22	16.74	34.67	16.19	23.42	14.82	17.37	21.92
-S1	30	100	24.77	28.24	17.07	**34.97**	16.53	19.51	**17.34**	16.05	21.81
-S1	100	30	21.55	30.41	17.12	34.06	**16.70**	**26.67**	13.19	13.19	21.61
-S1	100	60	22.92	29.22	16.74	34.67	16.19	23.42	14.82	17.37	21.92
-S1	100	100	**25.75**	**40.63**	**20.31**	32.44	11.40	12.98	14.42	**17.96**	**21.99**
-S3	10	30	**6.50**	**10.91**	**4.05**	**2.70**	**2.42**	**4.10**	**0.68**	**1.27**	4.08
-S3	10	60	6.45	10.85	4.02	2.69	2.40	4.09	0.67	1.26	4.05
-S3	10	100	6.34	10.80	3.99	2.67	2.39	**4.10**	0.65	1.25	4.02
-S3	30	30	**6.50**	**10.91**	**4.05**	**2.70**	**2.42**	**4.10**	**0.68**	**1.27**	4.08
-S3	30	60	6.45	10.85	4.02	2.69	2.40	4.09	0.67	1.26	4.05
-S3	30	100	6.34	10.80	3.99	2.67	2.39	**4.10**	0.65	1.25	4.02
-S3	100	30	**6.50**	**10.91**	**4.05**	**2.70**	**2.42**	**4.10**	**0.68**	**1.27**	4.08
-S3	100	60	6.45	10.85	4.02	2.69	2.40	4.09	0.67	1.26	4.05
-S3	100	100	6.34	10.80	3.99	32.44	2.39	**4.10**	0.65	1.25	**7.75**

Table 13: BPR scores on MC22 tuning data for the VORM model. Best model scores per language and per ablation boldfaced.

	ces	eng	fra	hun	ita	lat	rus	avg.
morflogtoken	14.79	46.75	23.59	**35.03**	**12.14**	17.70	**12.34**	23.19
morftoken	10.37	40.52	20.94	34.13	11.23	17.68	10.60	20.78
morftype	**28.71**	**49.90**	23.63	**34.47**	11.84	**17.77**	11.46	**25.39**

Table 14: BPR scores on MC22 tuning data for Morfessor. Best model scores boldfaced

C Further quantitative breakdown of results

This Appendix supplements section 6 with the results broken down along several axes.

- Table 15 displays the results on the DoReCo-Morph dataset, broken down per language.

	chip	morf	para	AG	vorm	S1	-C	S3
apah1238	90.2	85.8	83.8	80.3	86.6	88.9	86.6	69.0
arap1274	92.9	90.4	70.2	90.6	93.8	90.8	93.8	64.1
bain1259	95.8	77.1	80.3	93.1	92.6	93.8	92.6	53.1
beja1238	89.8	85.0	80.9	89.7	95.0	95.2	95.0	50.8
bora1263	87.3	80.4	67.3	87.2	93.5	94.3	93.5	35.4
cabe1245	94.2	82.6	82.7	77.0	92.5	89.5	92.5	82.5
cash1254	95.0	87.0	81.9	88.1	91.6	91.9	91.6	65.9
dolg1241	95.9	87.2	81.6	88.6	92.2	92.2	92.2	65.9
even1259	69.9	81.5	81.4	79.4	84.8	85.9	84.8	72.2
goem1240	95.5	90.7	84.2	80.5	94.3	87.3	94.3	88.4
goro1270	82.6	82.0	79.4	80.7	88.1	87.6	88.1	71.8
hoch1243	90.4	89.0	79.0	87.1	90.6	86.4	90.6	60.2
jeha1242	93.4	92.8	90.0	85.4	91.7	91.5	91.7	87.1
jeju1234	93.8	86.4	82.0	86.5	90.7	91.0	90.7	65.7
kaka1265	82.7	83.6	82.5	80.0	87.7	88.6	87.7	73.2
kama1351	95.3	90.7	87.5	91.6	95.5	95.3	95.5	68.8
komn1238	92.5	86.7	78.8	91.8	93.5	93.6	93.5	57.7
movi1243	89.8	86.2	76.1	85.1	89.2	87.0	89.2	62.6
ngal1292	94.8	89.2	67.9	87.7	83.6	78.3	83.6	60.5
nisv1234	94.7	89.4	85.7	87.2	90.3	90.1	90.3	75.9
nngg1234	91.6	89.5	80.5	71.3	87.3	82.9	87.3	87.8
nort2641	93.6	85.3	85.5	84.5	91.1	91.0	91.1	79.9
nort2875	86.8	84.2	79.6	84.7	89.8	89.0	89.8	69.0
orko1234	88.9	84.5	78.2	76.2	86.7	82.2	86.7	86.1
pnar1238	95.1	90.4	84.1	85.9	91.3	89.9	91.3	73.4
port1286	90.3	84.8	83.6	73.9	90.0	84.9	90.0	89.9
ruul1235	91.9	86.8	74.0	87.8	90.7	90.1	90.7	53.1
sanz1248	94.4	86.3	76.3	85.1	78.0	80.2	78.0	63.7
savo1255	90.9	88.3	83.8	88.4	90.7	90.5	90.7	70.0
sout2856	92.7	89.1	84.7	84.3	89.3	87.6	89.3	77.8
sumi1235	94.2	87.0	85.1	86.6	92.1	93.7	92.1	57.8
taba1259	91.8	82.5	81.3	86.4	88.2	83.0	88.2	64.6
teop1238	89.5	84.1	77.2	74.0	84.3	81.8	84.3	86.0
texi1237	92.2	80.0	76.2	85.5	87.9	85.8	87.9	62.9
trin1278	96.7	85.4	73.0	91.1	90.0	90.1	90.0	50.5
urum1249	95.8	89.3	86.1	86.4	92.1	92.4	92.1	69.1

Table 15: EMMA-2 results for the DoReCo dataset for Chipmunk (supervised), Morfessor, ParaMA2, MorphAGram, and Vorm (with ablation variants).

Do large language models solve verbal analogies like children do?

Tamar Johnson$^\diamond$, **Mathilde ter Veen**$^\diamond$, **Rochelle Choenni**†,
Han L. J. van der Maas$^\diamond$ **Ekaterina Shutova**† **and Claire E. Stevenson**$^\diamond$,
$^\diamond$Psychological Methods, University of Amsterdam, the Netherlands
†ILLC, University of Amsterdam, the Netherlands
`t.johnson@uva.nl`
`c.e.stevenson@uva.nl`

Abstract

Analogy-making lies at the heart of human cognition. Adults solve analogies such as *horse belongs to stable like chicken belongs to . . . ?* by mapping relations (*kept in*) and answering *chicken coop*. In contrast, young children often use association, e.g., answering *egg*. This paper investigates whether large language models (LLMs) solve verbal analogies in A:B::C:? form using associations, similar to what children do. We use verbal analogies extracted from an online learning environment, where 14,006 7-12 year-olds from the Netherlands solved 872 analogies in Dutch. The eight tested LLMs performed at or above the level of children, with some models approaching adult performance estimates. However, when we control for solving by association this picture changes. We conclude that the LLMs we tested rely heavily on association like young children do. However, LLMs make different errors than children, and association doesn't fully explain their superior performance on this children's verbal analogy task. Future work will investigate whether LLMs associations and errors are more similar to adult relational reasoning.

1 Introduction

Analogy-making, using what you know about one thing to infer knowledge about a new, somehow related instance, lies at the heart of human intelligence and creativity and forms the core of educational practice (Gentner, 1988; Hofstadter, 1997; Holyoak, 2012). Given how important analogical reasoning is to learning and generalization, much research has focused on how this seemingly unique human ability emerges, develops, and can be improved (Goswami, 1991; Sternberg and Nigro, 1980; Stevenson and Hickendorff, 2018) as well as emulated in machines (Gentner and Forbus, 2011; Mitchell, 2021). Recently, large language models (LLMs), such as GPT-3 (Brown et al., 2020), have demonstrated surprisingly good performance in verbal analogy solving (e.g., *table is to legs as tree is to . . . ? chair, leaves, branches or roots?*) (Lu et al., 2022; Webb et al., 2023). The question then arises *how* LLMs solve these analogies. Is it similar to adult humans using relational mapping? Or perhaps more similar to the associative processes children tend to use?

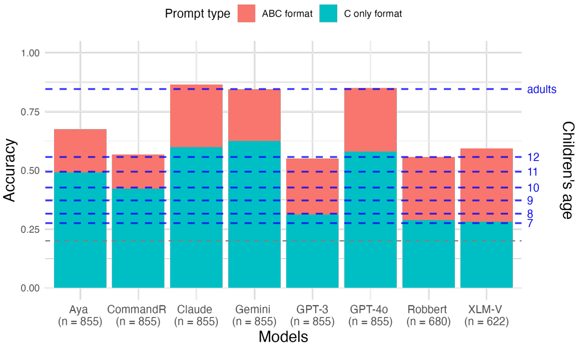


Figure 1: How well does each LLM perform? We see that when prompted with A:B::C:? many LLMs outperform children. However, LLMs can also solve most items by association, evidenced by correctly solving analogies when only prompted with C:?.

Earlier work shows that language models largely rely on semantic similarity between analogy terms to solve analogies (Rogers et al., 2020; Ushio et al., 2021b), which would indicate solving by association. In this paper we investigate whether LLMs use association or analogy to solve a set of Dutch verbal analogies. First, we examine how LLM performance compares to children and find that the best models outperform 12-year-olds, approaching adult performance estimates. Second, we examine whether LLM performance is influenced by the same item characteristics that affect children's analogy solving, where results confirmed that this is indeed the case, especially for lower performing models. Third, through a series of prompting experiments we show that these LLMs appear to use association to solve a large proportion of analogies.

Proceedings of the 29th Conference on Computational Natural Language Learning, pages 627–639
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

Fourth, we compare error patterns of children with LLMs and find that LLMs are far more similar to each other (and those of similar architecture and size) than to children.

This paper contributes to the study of analogical reasoning in LLMs in three ways: (1) it is the first to directly compare LLM verbal analogy solving performance to that of children; (2) we use experiments to tap into whether LLMs solve analogies using association like young children; and (3) we use Dutch rather than English language items and examine performance in multilingual LLMs.

2 Theoretical Background

2.1 The Analogical Reasoning Process

Although there are different cognitive models of analogical reasoning—varying in the order of processing steps and whether these occur sequentially or in parallel—there is a general consensus on which processes are involved. Taking the example of *"body is to feet as tree is to . . . ?"* (or more abstractly, A:B::C:?), the basic analogy information processing steps are generally considered to be: (1) encoding relevant information about the base (A:B) and target (C) domains; (2) searching and retrieving relationships and similarities between the analogy elements in the base domain, A and B (e.g., *"stands on"* for body and feet); (3) aligning the base and target domains (*"body and tree are things that stand"*) and mapping the mostly likely relationship between A and B, to the target domain, C, to come up with D; and (4) evaluating the validity of the predicted solution (Gentner and Hoyos, 2017; Sternberg, 1977; Thibaut and French, 2016).

2.2 Factors Affecting People's Verbal Analogy Solving

The basic analogy solving steps are consistently found in people from about 12 years and up (Thibaut and French, 2016). When adults make mistakes there are three main factors that lead to errors: (1) the relation type (causal is more difficult than categorical), (2) a large conceptual distance between analogy base and target domains, and (3) salient distractors amongst the multiple-choice options (Jones et al., 2022).

Type of Relation Jones et al. (2022) grouped analogical relations into three types: categorical, causal and compositional. They found that adults perform better on categorical analogies (e.g., *tarantula:spider::bee:insect*) than causal (e.g., *fracture:cast::incision:scar*) or compositional (e.g., *fingernail:finger::knee:leg*) analogies. Children's performance follows a similar pattern, assuming sufficient domain knowledge is in place (e.g., Sternberg and Nigro, 1980; Goswami and Brown, 1990; Alexander and Kulikowisch, 1991).

Conceptual Distance Between Base and Target Domains The greater the distance between an analogy base and target domain the more difficult the analogy is for adults and children to solve (Jones et al., 2022; Thibaut and French, 2016). For example, *bowl:dish::spoon:silverware* is easier for people to solve than *wrench:tool::sad:mood*.

Distractor Salience People are sometimes lured to choose a distracting incorrect response in multiple choice verbal analogies, and are most easily distracted by answer options that have a strong semantic association with the C term (Kucwaj et al., 2022). Jones et al. (2022) defines distractor salience as the relation between C:D relative to each of the C:D', where D' represents each distractor option. Distractor salience is high, when the semantic similarity between C and one of the incorrect answers D' is greater than the semantic similarity between C and the correct answer D. High distractor salience leads to lower performance in adults (Ichien et al., 2020; Jones et al., 2022) and this is even more apparent in children (Richland et al., 2006; Thibaut and French, 2016).

2.3 Analogical Reasoning Development

Children's verbal analogical reasoning improves with age, where a gradual shift occurs around 4-8 years of age from reasoning based on surface similarities and associations to reasoning based on (abstract) relations (Gentner, 1988; Stevenson and Hickendorff, 2018; Gentile et al., 1977). For example, if we ask a four-year-old *"horse belongs to stable like chicken belongs to . . . ?"* they may use association and reply *"egg"*, relying on the strong connection between the words chicken and egg to solve the problem. In contrast, older children and adults will likely give the intended relational response *"chicken coop"*, using the underlying relation structure to solve the analogy.

However, even when children can solve these analogies, two main factors that seem to affect the transition from associative to relational reasoning are increased domain knowledge (Goswami and

Brown, 1990; Gentner, 1988; Alexander and Kulikowisch, 1991) and improved executive functions (working memory and inhibition control; Doumas et al., 2018; Thibaut and French, 2016).

Children tend to fail in analogy solving if they are unfamiliar with the elements or relations in the analogy (Gentner and Hoyos, 2017; Goswami and Brown, 1990; Goddu et al., 2020). If children are shown to possess the required domain knowledge and are provided clear instructions on how to solve the task then they can successfully solve verbal analogies (in the form of pictures) as early as 3-years-old (Goswami, 1991; Goddu et al., 2020).

However, even when children can solve these analogies, evidence from scene analogy problems (Richland et al., 2006) and eye-tracking studies (Thibaut and French, 2016) shows that children up to 8 years-old tend to focus first on the C term when solving analogies, sometimes ignoring A and B altogether (Thibaut and French, 2016). This appears to be related to limited working memory capacity (Richland et al., 2006; Stevenson et al., 2013; Stevenson, 2017) and limits in inhibition- and executive control (Thibaut and French, 2016; Doumas et al., 2018). Performance improves when interventions are used that support children's processing capacities (Stevenson and Hickendorff, 2018) and when children are forced to focus first on the A:B pair (Glady et al., 2017).

2.4 Verbal Analogy Solving in LLMs

The extent to which LLMs can solve analogies is a subject of debate. Most of this work has focused on comparing models in terms of overall accuracy on benchmarks such as the Bigger Analogy Test Set (BATS; Mikolov et al., 2013b) and verbal analogies from the Scholastic Assessment doTest (SAT; Turney et al., 2003) and investigating the types of relations they can solve (e.g., syntactic versus semantic). More importantly, when LLMs demonstrate analogy solving abilities, it is unclear how they achieved these solutions (e.g., Webb et al., 2023), whether this is through relational reasoning or another process, such as the associative strategy often employed by young children.

Word embeddings Over a decade ago, Mikolov et al. (2013b) published their seminal paper showing that pre-trained word embeddings (e.g., Word2Vec Mikolov et al., 2013a) could be used to solve verbal analogies in the form of A:B::C:? using vector arithmetic, the most famous example being: $embed(king) - embed(man) + embed(woman) \approx embed(queen)$, where $embed$ represents the word embedding obtained from the pre-trained neural network. This milestone was tempered by Gladkova et al. (2016), who made clear that this method was limited in the breadth of relations that it could process. For example, the capitol-country relation was solved quite successfully, but others such as animal-sound and part-whole, were solved less successfully.

Transformer language models With the rise of the Transformer architecture, featuring language models such as BERT (Devlin et al., 2018), verbal analogy solving remained a challenge. Earlier work transferred the verbal analogy datasets, such as the BATS to the sentence level, and showed that BERT-based models and GPT-2 (Radford et al., 2019) performed at a similar level to GloVe (Pennington et al., 2014), a word embedding model, on analogies containing relations such as capitol-country and male–female pairs (Zhu and de Melo, 2020). More recently, Czinczoll et al. (2022) developed a dataset containing scientific and metaphor analogies (SCAN). Here there was a clear advantage of transformer models over analogy solving with word embeddings, where GPT-2, BERT and M-BERT outperformed GloVe on the analogy items containing metaphors such as *career:mountain::success:ascent*. Also, Petersen and van der Plas (2023) showed that by changing the training objective of LLMs to maximize relational similarity, LLM performance improves. Yet, the general conclusion remained that verbal analogy solving is more challenging for LLMs than people.

People versus LLMs in analogy solving Recent research has shown that LLMs can solve verbal analogies with similar accuracy to people. For example, Ushio et al. (2021b) showed that LLMs such as GPT-2 and RoBERTa generally perform well on analogies designed for 4th to 10th graders (9-16 year-olds). Also, Webb et al. (2023) concluded that GPT-3 and GPT-4 generally perform around the same level as adults on two verbal analogy datasets.

Item factors affecting LLM verbal analogy solving There has been some research on the effect of *relationship type* on LLM's verbal analogy solving performance. Ushio et al. (2021a) showed that fine-tuned RoBERTa models performed slightly better on categorical relations (hypernymns) than compositional ones (meronymns). And Webb et al. (2023)

found that categorical relations in the SAT verbal analogies were easier for GPT-3 than compositional (function) relations and also that categorical relations were easier than both compositional and causal relations on the items from Jones et al. (2022). Similarly, Linford et al. (2022) found that categorical relations were easier for BERT models than causal relations, although performance on both was far lower than for human adults.

Similarly to people, LLMs have more difficulty as the *conceptual distance* between the domains in the analogy increases. For example, the LLMs in Czinczoll et al. (2022) performed better on the BATS analogies than on their SCAN dataset comprising scientific and metaphor based analogies, where the semantic distance between the base and target domains was greater. In addition the scientific analogies were solved better by LLMs than those based on metaphors, which was explained by there being a clearer correspondence between base and target domains in scientific analogies. Also, Webb et al. (2023), used the items from Jones et al. (2022) to investigate whether, like in people, a near conceptual distance between the base and target domains made analogies easier to solve for GPT-3 than far analogies; this was indeed the case. Interestingly,do humans outperformed GPT-3 on the far analogies.

There is less research on the effect of *distractor salience* on LLM analogy solving. In Petersen and van der Plas (2023) their best performing trained model appeared unaffected by low versus high distractor salience. In Musker et al. (2024), analogy tasks presented in an in-context-learning setting with interleaved distractors affected LLMs more than human adults. We expect that salient distractors, i.e. multiple-choice options that are semantically more similar to the analogy terms than the correct response, will have a greater chance of being "selected" by the LLMs.

3 Research Questions

In this study, with pre-registered hypotheses and methods, we examine how 8 multilingual LLMs solve 872 verbal analogies, also solved by 14,006 in an online learning environment.

RQ1: How well do LLMs perform compared to children ages 7-12 in verbal analogy solving? We expected recent LLMs to solve the analogies with similar accuracy to older children (12-year-olds) as this is similar to adult performance (hypothesis 1; Webb et al., 2023; Ushio et al., 2021a).

RQ2: Which item characteristics influence children's and LLM performance on verbal analogies? We expected the pattern of results found in adults also to be found in children and in LLMs. A growing strand of work shows that children, from a very young age, are remarkably sensitive to distributional regularities in their input and are adept at learning from this type of information (e.g., Saffran et al., 1996; Bresnan, 2007; Clark, 2014). Given the similarity (albeit to a limited extent) to how LLMs extract and track information from their input, we investigate whether the two learners are affected by shared item characteristics. First, we expect performance on categorical relations to be better than compositional and causal relations for both children (Sternberg and Nigro, 1980, hypothesis 2a1) and LLMs (Webb et al., 2023, hypothesis 2a2). Second, we expect analogies with a near conceptual distance between A:B to be easier than far analogies for children (Thibaut and French (2016); Hypothesis 2b1) and LLMs (Czinczoll et al., 2022; Webb et al., 2023, hypothesis 2b2). Third, we expect higher distractor salience to lead to more errors in children (Thibaut and French, 2016, hypothesis 2c1) and LLMs (Ushio et al., 2021b, hypothesis 2c2).

RQ3: Do LLMs choose associative or analogical solutions? We investigate this through a series of experiments comparing LLM performance on alternative formulations of the verbal analogies, where we control for associative responses.

4 Methods

LLM data and code and a selection of the children's data is publicly available. The full dataset is available upon request from Prowise Learn, the company that provided the children's data on the verbal analogies dataset.

4.1 Prowise Learn's Verbal Analogies Game

Prowise Learn is an online adaptive learning environment for elementary school children.

Verbal analogies is one of the games on the platform (see Appendix A for a screenshot of the game). The analogies are presented as text in "A:B::C:?" format, and the children must choose among five answer options, all five of which are semantically associated with C. For more information see Appendix A.

Data Collection with Children For this study, we extracted information on 14,006 7-12 year-old's (M = 10.73, SD = 1.15 years) performance on 872 verbal analogies from the Prowise Learn database. We applied three selection criteria when extracting the children's data (on June 19, 2021): (1) children solved at least 20 items to ensure stable ability estimates, (2) children had last played the game on or after September 1st 2020, the start of the school year and 4 months after the launch of the game, when item difficulty estimates were verified to have small standard errors and (3) children were ages 7-12 to avoid confounds in performance (i.e., younger children most likely did not have sufficient reading abilities and older children had most likely repeated a grade). This data collection was approved by the University of Amsterdam's Ethics Review Board with id FMG-3037.

Data Collection with Adults To provide an estimate of adult-level performance on this children's verbal analogy task, we collected data from 120 Dutch-speaking adults (M = 29.20, SD = 9.96 years) through Prolific's academic participant recruitment system. Each person solved 30 analogies presented in a similar format to those of children, with the aim of having each item solved by 4 adults to estimate item-level performance. Also, to test to what extent adults solve analogies by association we administered 30 additional items in C-only format (see 7.1 for a description). We applied two inclusion criteria before analyzing the adults' data. First, we included adults that solved >=50% of items correctly (i.e., achieved at least average children's performance), which led to 3 participants being excluded. Second, we included adults who explicitly stated that they did not use AI-tools to solve the analogies (1 person excluded). This data collection was approved by the University of Amsterdam's Ethics Review Board with id FMG-3105.

Item Selection The game contained three types of verbal reasoning problems; verbal analogies was one of them. From the initial set of 872 verbal analogies, we checked all items that were outliers (>1.5 SD) on the item difficulty scale and removed 17 items that were judged by two independent raters to contain errors (e.g., multiple correct solutions, requiring domain knowledge likely unfamiliar to children). This resulted in 855 items for data analysis.

4.2 Item characteristics

Relation Type Relationship type refers to how the A and B term are related. This relationship is applied to the C-term to find D. Table 2 provides a selected overview of relation types in the analogy task[1]. For analyses related to RQ2 we selected 302 items that fall into the following three categories defined by Jones et al. (2022):

- **Categorical**: one of the A:B terms defines the category and the other word is an example of this category. For example, "yellow" is part of the category "color".
- **Causal**: one of the A:B terms is the cause and the other is the effect. For example, "stumbling" will result in "falling".
- **Compositional**: one of the A:B terms is part of the other term. For example, "leaf" is part of a "tree".

Conceptual Distance Between Base and Target Domains We used three vector-based language models[2] to compute the semantic distance (1 - cosine similarity) between the A:B and the C:D pair. We used the mean value over the three vector-based models as the selected category for each item for analysis.

Distractor Salience Distractor salience was measured by the cosine similarity between C and D minus the cosine similarity between C and each incorrect answer D'. We used the same three vector-based models from Section 4.2 to compute the cosine distances between embeddings for C and each of the five D's. Then we determined distractor salience per item for each vector model and used the mean value for analysis.

4.3 Analogy completion with LLMs

Pretrained Language Models We studied how 8 transformer-based multilingual LLMs solved the same set of verbal analogies as the children.

Two of the LLMs are BERT-based masked language models. **RobBERT** (Delobelle et al., 2020) was pretrained on Dutch data only, and a multilingual variant **XLM-V** (Liang et al., 2023) was

---

[1] These labels were chosen and annotated by the Prowise Learn item developers.

[2] Word2Vec trained by CLIPS on different Dutch corpora (Tulkens et al., 2016), Word2Vec trained by the Nordic Language Processing Laboratory on the CoNLL17 corpus (Kutuzov et al., 2017), and FastText trained on Common Crawl and Wikipedia (Grave et al., 2018).

trained on 116 languages.[3] Identical to BERT (Devlin et al., 2018), both models contain 12 layers with 12 attention heads each.

The other LLMs are autoregressive transformer-decoder based language models. The open-source models we use are **Aya** and **Command-R**, both accessed through the Cohere API. The proprietary models we use are Anthropic's **Claude Sonnet-3.5**, Google's **Gemini-2.0-flash**, and Open AI's **GPT-3** and **GPT-4o**, each accessed through the API provided by the respective company.

Analogy completion We wanted to mimic the way the children solved the analogies in the best way possible. This was especially important because we investigate whether an associative response is more likely in the presence of a correct response. Therefore, we prompted the generative LLMs with the full analogy and asked them to choose from the five response options. For example, "tripping is to falling as picking up is to ? Choose clean, junk, mess, room, or thrift store." The response options were presented in random order.

However, this method was not possible to implement for the BERT-based models. Therefore, for the RobBERT and XLM-V models we used the masked language model approach and fed the models '`A` is to `B`, as `C` is to `D`', replacing `D` with each possible multiple-choice solution. The `D` option with the highest probability for the completion was considered the selected response.

5 Results RQ1: How well do LLMs perform compared to children?

Figure 1 shows performance per model on the 872 items. We see that all tested models, both BERT-based and autoregressive transformer-decoder based language models, perform at or above the level of children on the multiple choice question verbal analogy task. Children already at the age of 7 perform higher than chance level (gray dashed line), with Aya , Command-R , GPT-3 , RobBERT and XLM-V around the same level as 12 years old, whereas Claude , Gemini and GPT-4o outperform all children and other models, and perform at the level of adults on this task.

We analyzed how many of the items LLMs could solve by word association and report their performance on the C:? task (Experiment 1, see also 7). Results show that for the autoregressive transformer-decoder based models, word association can explain most of their success, but also in other models a large portion of items can be solved solely by association (Figure 1, blue portion of the bars). See 7 for further details and conclusions.

6 Results RQ2: Which item factors influence analogy solving?

For RQ2, we tested the effects of solver (children, LLMs) and/or item characteristics on accuracy. 9 logistic regression models (one per each solver type) predicted the performance on each item by relation type, semantic distance between base and target domains and distractor salience. We also included by-item intercepts as random effects.

Relation Type Logistic regression analyses of children's performance revealed significant effect of relation type ($beta = -0.49, z = -2.20, p < 0.05$) such that compositional items were easier than causal items for children to solve, following a similar pattern previously found in adults (Jones et al., 2022). Relation type did not significantly influence performance in most models with the exception of Gemini that performed better on items with compositional than categorical relations ($\beta = -0.9, z = -658.5, p < 0.001$), and on items with compositional than causal relations ($\beta = -0.19, z = -144.4, p < 0.001$). RobBERT showed similar significant effect of compositional versus categorical relation items ($\beta = -0.64, z = -2.58, p < 0.001$). GPT-3 , however, performed better on categorical than compositional items ($\beta = 0.45, z = 2.06, p < 0.05$).

Semantic Distance between Base and Target Domains Logistic regression of children's performance revealed a significant effect of the semantic distance between the base and target domains. Items with shorter distance were easier for children to solve ($\beta = -0.54, t = -4.05, p < 0.001$). A similar pattern was observed in most LLMs including Aya ($\beta = -1.79, z = -2.16, p < 0.05$), Command-R ($\beta = -3.07, z = -3.90, p < 0.001$),Gemini ($\beta = -6.23, z = -4536.2, p < 0.001$), GPT-4o ($\beta = -2.84, z = -2.67, p < 0.01$), GPT-3 ($\beta = -2.64, z = -3.31, p < 0.001$), RobBERT ($\beta = - - 4.15, z = -4.38, p < 0.001$) and XLM-V ($\beta = -5.13, z = -4.67, p < 0.001$). Only Claude showed no significant effect of se-

---

[3] We found XLM-V to be more suitable than mBERT or XLM-R as it suffers less from overtokenization in Dutch and thus covers more of our test words.

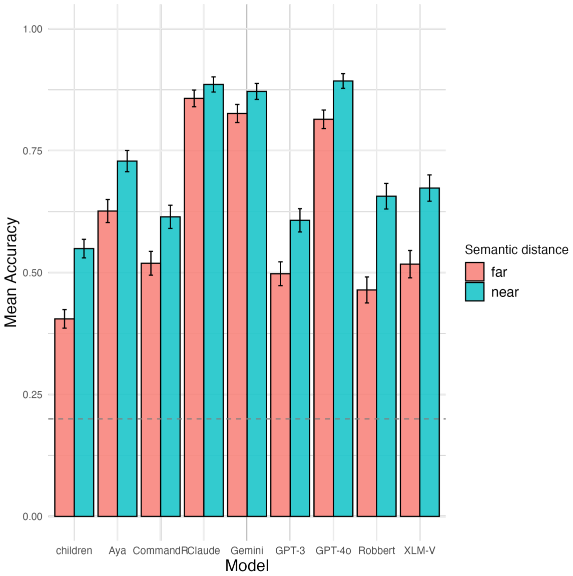


Figure 2: Near analogies are often easier to solve than far analogies for both children and LLMs. Note: for purpose of clarity, in the plot we binned semantic distance into *near* and *far* categories, where *near* is < median semantic distance and *far* is >= median semantic distance.

mantic distance on performance ($p = 0.95$)(see Figure 2).

Distractor Salience As can be seen in Figure 3, items with lower distractor salience were significantly easier to solve than those with high distractor salience for children ($beta = -2.65, z = -4.33, p < 0.001$) and most LLMs including Aya ($beta = -2.76, z = -4.07, p < 0.001$), Command-R ($beta = -2.63, z = -4.31, p < 0.001$), Gemini ($beta = -1.81, z = -1322.5, p < 0.001$), GPT-4o ($beta = -1.99, z = -2.61, p < 0.01$), GPT-3 ($beta = -3.39, z = -4.99, p < 0.001$), RobBERT ($beta = -2.69, z = -3.94, p < 0.001$) and XLM-V ($beta = -2.64, z = -3.63, p < 0.001$). Only Claude showed no significant effect of distractor salience on performance ($p = 0.89$).

7 Results RQ3: Do LLMs choose associative or analogical solutions?

We investigated whether LLMs choose analogical solutions to verbal analogies, after explicitly testing and controlling for associative responses.

7.1 Experiment 1: C:?

In experiment 1, we prompt the LLMs (and adult participants) with only the C-term, e.g., "C is to [MASK]". If these are solved by association as we expect, then LLMs should still be able to solve a substantial portion of analogies purely by association with C (Ushio et al., 2021b; Poliak et al., 2018); hypothesis 3a). This was indeed the case as can be seen in Table 1, where the generative LLMs solve up to 62% of items without being given A:B. Notably, adults also solved 57% of items with the C-only prompt, where there was 49-71% overlap between models and adults in which items could be solved by association.

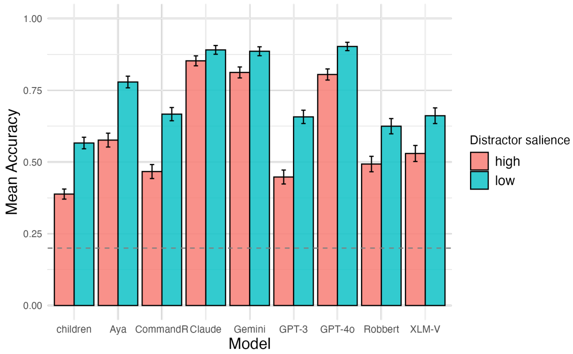


Figure 3: Analogies with low distractor salience are easier to solve than those with high distractor salience for both children and LLMs. Note: for purpose of clarity, in the plot we binned distractor salience into *low* and *high* categories, where *low* is < median distractor salience and *high* is >= median distractor salience.

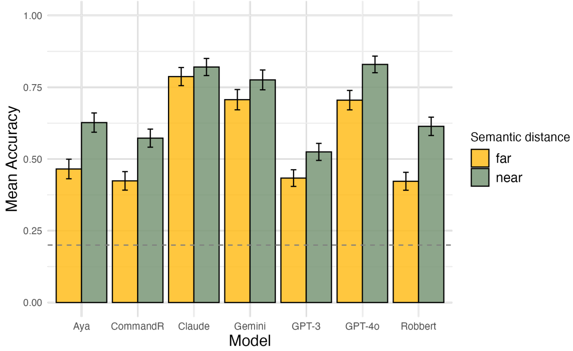


Figure 4: Near analogies are still easier than far analogies, when we control for associative responses (i.e., when filtering out the correctly solved C:? items for each model).

7.2 Experiment 2: A:B::C:? for selected items

We removed items that each model solved correctly with C:? and reevaluated their performance along the same item factors from RQ2. This was done to test the factors affecting the models' performance on items that were not solved by word association alone. We ran logistic mixed effects models predicting the performance of each LLM by the three

	Exp 0 A:B::C:? LLMs		**Exp 1 C:? LLMs**	**Exp 2 filtered A:B::C:?**							
				LLMs		**Children**					
						7-yrs	**8-yrs**	**9-yrs**	**10-yrs**	**11-yrs**	**12-yrs**
model	N items	Acc (SD)	Acc (SD)	N items	Acc (SD)	M (SD)	M (SD)	M (SD)	M (SD)	M (SD)	M (SD)
Aya	855	.67 (.47)	.49 (.50)	435	.54 (.50)	.21 (.37)	.25 (.38)	.30 (.38)	.35 (.38)	.42 (.39)	.49 (.39)
Command-R	855	.57 (.50)	.42 (.49)	494	.50 (.50)	.23 (.39)	.28 (.39)	.34 (.39)	.39 (.39)	.47 (.38)	.53 (.39)
Claude	855	.86 (.34)	.60 (.49)	343	.80 (.40)	.15 (.32)	.19 (.33)	.25 (.34)	.30 (.35)	.37 (.36)	.44 (.36)
Gemini	855	.84 (.36)	.62 (.48)	321	.73 (.45)	.10 (.27)	.14 (.28)	.19 (.29)	.25 (.31)	.33 (.33)	.40 (.35)
GPT-4o	855	.85 (.36)	.58 (.49)	359	.76 (.43)	.15 (.32)	.19 (.33)	.24 (.34)	.30 (.35)	.37 (.36)	.44 (.37)
GPT-3	855	.55 (.36)	.31 (.49)	359	.47(.50)	.24 (.39)	.27 (.40)	.33 (.39)	.38 (.39)	.45 (.39)	.52 (.38)
RobBERT	680	.56 (.50)	.29 (.45)	484	.51 (.50)	.25 (.40)	.30 (.40)	.35 (.40)	.40 (.39)	.48 (.39)	.54 (.38)
XLM-V	622	.59 (.49)	.28 (.45)	447	.51 (.50)	.24 (.39)	.28 (.39)	.34 (.39)	.41 (.39)	.48 (.38)	.55 (.38)

Table 1: LLM Performance on Experiment 0 (original set of A:B::C:? items), Experiments 1 (C:?) and 2 (selection of A:B::C:? after filtering out correct C:? items by each model). Children's mean proportion correct (by age group) on the same selection of items per LLM from Experiment 2.

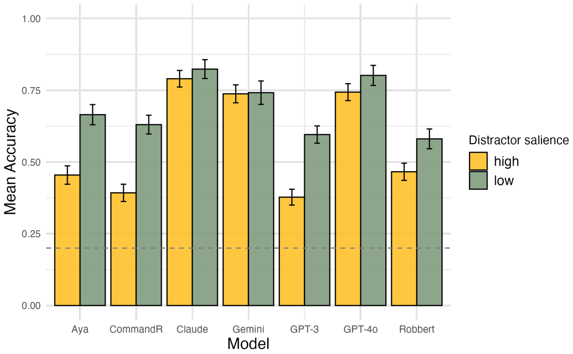


Figure 5: Analogies with low distractor salience are still easier for LLMs, when we control for associative responses.

item characteristics we tested (relation type, semantic distance between base and target domains and distractor salience) on the selected items after filtering out items that were also solved by association only for each model [4]. The models also included by-item intercepts as random effects. Results show that LLMs' performance is still affected by the semantic distance between the base and the target words (see 4) and by distractor salience (see 5). The only exceptions were for GPT-4o and XLM-V , where there were no longer significant effects of distractor salience ($p = 0.43$ and $p = 0.22$, respectively), and for Gemini where there were no significant effects of both semantic distance ($p = 0.57$) and distractor salience ($p = 0.88$) (see Appendix C for the full report of results).

Table 1 shows an overview of model versus children's performance where all items solved correctly with the C:? prompt had been filtered out. We see that BERT-based models solve nearly 30% of analogies correctly when prompted with only "C:?", so without any information about the relation A:B to be mapped. The autoregressive transformer-decoder models solved even greater portions correctly ($40 - 60\%$) with the C-only prompt. Notably, for the youngest children in our dataset, 7-8-year olds, performance dropped to below chance level on the filtered items sets.

7.3 RQ4: Do LLMs choose the same distractors as children do?

In this exploratory analysis we compared LLM errors to those of children and pilot data from adults. For each of the tested models, we looked at the subset of items it answered incorrectly and compared the distractor it chose to the one chosen by most children and piloted adults. We computed Cohen's Kappa coefficient (Cohen, 1960) to test the agreement of distractor choice between each pair of models and between each model and the children (see Figure 6). As can be expected, the Bert-based models, RobBERT and XLM-V , show similar error patterns, while having low agreement with the autoregressive transformer-decoder models. Notably, neither type of model architecture nor adult pilot data showed similar error patterns to those of children. These results suggest that the high performance of LLMs in this task is not driven by the same process as children. However, top-performing models -Gemini and GPT-4o - had similar error patterns to small sample of adults.

8 Discussion

The main goal of this paper was to investigate whether LLMs rely on association to solve verbal analogies, similar to young children. Direct performance comparisons showed that some LLMs

---

[4] This means that the set of A:B::C: selected items, as well as their total number, differ across models.

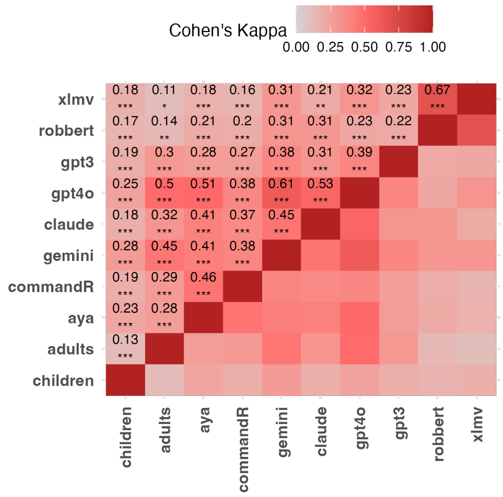


Figure 6: Inter-solver agreement in distractor choice measured using Cohen's Kappa. Values closer to 1 indicate higher agreement. LLMs and children show different error patterns in solving verbal analogies.

perform at the level of 12-year-olds, while top-performing models surpass children and reach adult-level performance. All LLMs appeared to rely heavily on association, though they made different errors than children, and association alone doesn't fully explain their superior performance on this children's analogy task.

To examine whether LLMs are influenced by the same analogy item characteristics as children, we tested the effects of distractor salience, semantic distance between base and target domains, and relation type. Both distractor salience and semantic distance affected LLM performance similarly to children, especially in smaller models. These effects persisted even when association didn't fully account for the reasoning. Relation type, however, did not influence most LLMs in the same way it does children.

A notable finding was that LLMs solved 28%–62% of analogies when prompted with only "C:?", without any information about the relation A:B to be mapped. This experimental manipulation is similar to Ushio et al. (2021b) who found that RoBERTa and BERT only dropped 10 to 15 percentage points in accuracy, still achieving accuracies of 30% or higher on the SAT analogies. In our case, LLMs also dropped around 10 percentage points after filtering out items solved correctly with C:? only. Interestingly, 7-8 year-olds performance often dropped to below chance level on the filtered item sets, which is what was expected as association is the most utilized strategy in this age-group (see Table 1; Thibaut and French (2016); Stevenson and Hickendorff (2018)). A small sample of adults tested on same items also solved 56% of the items when prompted with the C term only.

Our error analysis provides further insight into the similarities in verbal analogical reasoning between children and LLMs. While LLMs exhibit comparable error patterns—particularly among models with the same architecture—their mistakes only loosely align with those made by children. This suggests that there are differences in the way LLMs and children solve verbal analogies. We collected pilot data to examine whether LLM error patterns better coincided with adults to determine whether LLMs resemble more advanced human reasoning or rely on fundamentally different processes. Preliminary results show that the errors of top-performing LLMs, Gemini and GPT-4o - but not Claude , are somewhat similar to those of adults. However, each separate item was solved by only four adults, so future work must determine the reliability of these results.

Our study relies on behavioral methods to evaluate the model's performance on analogy tasks, which, while comparable to the methods used to investigate analogical reasoning in humans, do not allow us to draw definitive conclusions about the underlying mechanisms the model uses to solve these items. In future work, we intend to address this gap by employing mechanistic interpretability techniques—such as visualizing attention patterns—to more directly investigate the process by which LLMs solve analogies.

9 Conclusion

In sum, LLMs perform at or above the level of children on our verbal analogical reasoning task. While word association plays a significant role in LLMs' success, they are able to solve analogies also when this strategy is absent. While LLMs share some similarity to children in the factors that affect performance, the errors they make suggest a different mechanism. Future work can contrast adult-like relational mapping with other possible mechanisms children have been postulated to use such as relational priming (Leech et al., 2008) or partial analogical reasoning (Stevenson and Hickendorff, 2018) to further examine how LLMs solve verbal analogies.

References

P. Alexander and J. Kulikowisch. 1991. Domain knowledge and analogical reasoning ability as predicators of expository text. *Journal of Reading Behavior*, 23(2):165–190.

Joan Bresnan. 2007. Is syntactic knowledge probabilistic? experiments with the english dative alternation. *Roots: Linguistics in search of its evidential base*, 96:77–96.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Alexander Clark. 2014. Distributional learning as a theory of language acquisition. In *Proceedings of the 5th Workshop on Cognitive Aspects of Computational Language Learning (CogACLL)*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Tamara Czinczoll, Helen Yannakoudakis, Pushkar Mishra, and Ekaterina Shutova. 2022. Scientific and creative analogies in pretrained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. Robbert: a dutch roberta-based language model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, page 4171–4186.

Leonidas A. A. Doumas, Robert G. Morrison, and Lindsey E. Richland. 2018. Individual differences in relational learning and analogical reasoning: A computational model of longitudinal change. *Frontiers in Psychology*, 9:1235.

J. R. Gentile, L. Tedesco-Stratton, E. Davis, N. J. Lund, and B. C. Agunanne. 1977. Associative responding versus analogical reasoning by children. *Intelligence*, 1(4):369–380.

D. Gentner. 1988. Metaphor as structure mapping: The relational shift. *Child Development*, 59(1):47–59.

Dedre Gentner and Kenneth D Forbus. 2011. Computational models of analogy. *Cognitive science*, 2(3):266–276.

Dedre Gentner and Christian Hoyos. 2017. Analogy and abstraction. *Topics in cognitive science*, 9(3):672–693.

Nina Gierasimczuk, Han LJ van der Maas, and Maartje EJ Raijmakers. 2013. An analytic tableaux model for deductive mastermind empirically tested with a massively used online learning system. *Journal of Logic, Language and Information*, 22(3):297–314.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

Y. Glady, R. M. French, and J.-P. Thibaut. 2017. Children's failure in analogical reasoning tasks: A problem of focus of attention and information integration? *Frontiers in Psychology*, 8:707.

M. K. Goddu, T. Lombrozo, and A. Gopnik. 2020. Transformations and transfer: Preschool children understand abstract relations and reason analogically in a causal task. *Child Development*, 91(6):1898–1915.

U. Goswami. 1991. Analogical reasoning: What develops? a review of research and theory. *Child Development*, 62(1):1–22.

Usha Goswami and Ann L Brown. 1990. Melting chocolate and melting snowmen: Analogical reasoning and causal relations. *Cognition*, 35(1):69–95.

E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Douglas R Hofstadter. 1997. *Fluid concepts and creative analogies: computer models of the fundamental mechanisms of thought*. Allen Lane, The Penguin Press.

Keith J Holyoak. 2012. Analogy and relational reasoning. *The Oxford handbook of thinking and reasoning*, pages 234–259.

Nicholas Ichien, Hongjing Lu, and Keith J Holyoak. 2020. Verbal analogy problem sets: An inventory of testing materials. *Behavior Research Methods*, 52(5):1803–1816.

Laura L Jones, Matt J Kmiecik, John L Irwin, and Robert G Morrison. 2022. Differential effects of semantic distance, distractor salience, and relations in verbal analogy. *Psychonomic Bulletin & Review*.

Sylvia Klinkenberg, Marthe Straatemeier, and Han L van der Maas. 2011. Computer adaptive practice of maths ability using a new item response model for on the fly ability and difficulty estimation. *Computers & Education*, 57(2):1813–1824.

Hubert Kucwaj, Michał Ociepka, and Adam Chuderski. 2022. Various sources of distraction during analogical reasoning. *Memory & Cognition*, 50(7):1614–1628.

Andrey Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*.

Robert Leech, Denis Mareschal, and Richard P Cooper. 2008. Analogy as relational priming: A developmental and computational perspective on the origins of a complex cognitive skill. *Behavioral and Brain Sciences*, 31(4):357–378.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the Vocabulary Bottleneck in Multilingual Masked Language Models. *arXiv e-prints*, page arXiv:2301.10472.

Bryce Linford, Nicholas Ichien, Keith Holyoak, and Hongjing Lu. 2022. Impact of semantic representations on analogical mapping with transitive relations. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44.

Hongjing Lu, Nicolas Ichien, and Keith J Holyoak. 2022. Probabilistic analogical mapping with semantic relation networks. *Psychological Review*, 129:1078–1103.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

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

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Sam Musker, Alex Duchnowski, Raphaël Millière, and Ellie Pavlick. 2024. Semantic structure-mapping in llm and human analogical reasoning. *arXiv preprint arXiv:2406.13803*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Molly Petersen and Lonneke van der Plas. 2023. Can language models learn analogical reasoning? investigating training objectives and comparisons to human performance. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16414–16425, Singapore. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the 7th Joint Conference on Lexical and Computational Semantics*.

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.

Lindsey E. Richland, Robert G. Morrison, and Keith J. Holyoak. 2006. Children's development of analogical reasoning: Insights from scene analogy problems. *Journal of Experimental Child Psychology*, 94(3):249–273.

Anna Rogers, Olga Kovaleva, Doug Downey, and Anna Rumshisky. 2020. Getting closer to ai complete question answering: A set of prerequisite real tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8722–8731.

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

Robert J Sternberg. 1977. Component processes in analogical reasoning. *Psychological Review*, 84(4):353–378.

Robert J Sternberg and Georgia Nigro. 1980. Developmental patterns in the solution of verbal analogies. *Child Development*, 51:27–38.

Claire E Stevenson. 2017. Role of working memory and strategy-use in feedback effects on children's progression in analogy solving: An explanatory item response theory account. *International Journal of Artificial Intelligence in Education*, 27:393–418.

Claire E Stevenson, Willem J Heiser, and Wilma CM Resing. 2013. Working memory as a moderator of training and transfer of analogical reasoning in children. *Contemporary Educational Psychology*, 38(3):159–169.

Claire E Stevenson and Marian Hickendorff. 2018. Learning to solve figural matrix analogies: The paths children take. *Learning and Individual Differences*, 66:16–28.

Jean-Pierre Thibaut and Robert M French. 2016. Analogical reasoning, control and executive functions: a developmental investigation with eye-tracking. *Cognitive Development*, 38:10–26.

Stéphan Tulkens, Chris Emmery, and Walter Daelemans. 2016. Evaluating unsupervised Dutch word embeddings as a linguistic resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4130–4136, Portorož, Slovenia. European Language Resources Association (ELRA).

Peter D Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03)*.

Akihiro Ushio, Jose Camacho-Collados, and Steven Schockaert. 2021a. Distilling relation embeddings from pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9044–9062. Association for Computational Linguistics.

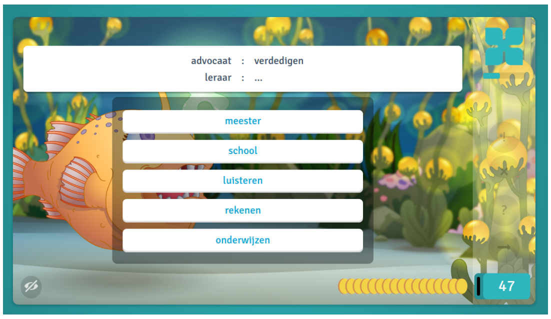
Akihiro Ushio, Luis Espinosa-Anke, Steven Schockaert, and Jose Camacho-Collados. 2021b. Bert is to nlp what alexnet is to cv: Can pre-trained language models identify analogies? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3609–3624. Association for Computational Linguistics.

Sanne HG van der Ven, Marthe Straatemeier, Brenda RJ Jansen, Sharon Klinkenberg, and Han LJ van der Maas. 2015. Learning multiplication: An integrated analysis of the multiplication ability of primary school children and the difficulty of single digit and multidigit multiplication problems. *Learning and Individual Differences*, 43:48–62.

Taylor Webb, Keith J Holyoak, and Hongjing Lu. 2023. Emergent analogical reasoning in large language models. *Nature Human Behaviour*, 7:1526–1541.

Xunjie Zhu and Gerard de Melo. 2020. Sentence analogies: Linguistic regularities in sentence embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3389–3400.

Figure 7: Example analogy "lawyer : defending :: teacher : educating"



A Prowise Learn Verbal Analogies Data

Prowise Learn games are adaptive, so that children solve items that are neither too difficult nor too easy, presenting children with items that they have a 65-85% chance of solving correctly, using response time to improve ability estimates (Klinkenberg et al., 2011). Each time a child solves an item his/her ability score on the game is updated according to an algorithm similar to the adaptive ELO rating system used for chess players (for details see Klinkenberg et al., 2011). At the same time the item's difficulty level is adapted according to the same algorithm. In this way item difficulty is on the same scale as the children's ability, and, as such item difficulties can be used to study children's abilities (see van der Ven et al., 2015; Gierasimczuk et al., 2013, for examples in math and logical reasoning). The ELO algorithm is based on the one-parameter logistic function from item response theory where we estimate the probability a child will solve an item correctly given the child's ability score θ and the item's difficulty level β as shown in Equation 1.

$$P(X = 1|\theta, \beta) = \frac{e^{(\theta-\beta)}}{1 + e^{(\theta-\beta)}} \tag{1}$$

Information extracted per item The following information was extracted per item: question text, answer options, item difficulty rating, standard error of item difficulty rating, type of analogy relation, number of times the item was solved, proportion of times each response option was selected.

B Effect of Relation Type on Children's and LLMs' Performance

B.1 Examples for each Relation Type

See Table 2.

Prowise Learn relations	N	relations*	example
action-result	36	causal	parasol : shadow :: sun : warmth
cause-effect	11	causal	falling : broken :: heating : hot
problem-solution	6	causal	noisy : earplugs :: illness : medicine
same category	28	categorical	lion : tiger :: dog : wolf
classification	51	categorical	lego : toys :: sock : clothes
item-characteristic	45	compositional	skyscraper : high :: lead : heavy
object-function	34	compositional	pan : cooking :: pen : writing
part-whole	51	compositional	gate : city :: door : house
share characteristic	25	compositional	giant : mountain :: dwarf : mouse

Table 2: * Mapping of selected relations in verbal analogies game to those examined in Jones et al. (2022).

C Results for item characteristics on items not solved by word association

Results from Aya show effect of semantic distance($beta = -3.94, z = -3.22, p < 0.01$) and distractor salience ($beta = -3.23, z = -3.34, p < 0.001$). Results from linear model of Command-R show effect of semantic distance ($beta = -4.14, z = -3.97, p < 0.001$) and distractor salience ($beta = -3.31, z = -3.99, p < 0.001$). No significant effect of semantic distance and distractor salience was found for Claude ($p = 0.27$ and $p = 0.73$, respectively). No significant effect of semantic distance and distractor salience was found for Gemini ($p = 0.57$ and $p = 0.88$, respectively). A significant effect of semantic distance was found for GPT-4o ($beta = -3.73, z = -2.55, p < 0.05$) but no significant effect of distractor salience ($p = 0.43$). Results from GPT-3 show effect of distractor salience($beta = -2.90, z = -3.97, p < 0.001$) with no significant effect of semantic distance($p = 0.12$). Results from linear model of RobBERT show effect of semantic distance($beta = -4.28, z = -4.08, p < 0.001$) and distractor salience ($beta = -1.86, z = -2.37, p < 0.05$). A significant effect of semantic distance was found for XLM-V ($beta = -5.01, z = -3.96, p < 0.001$) but no significant effect of distractor salience ($p = 0.22$).

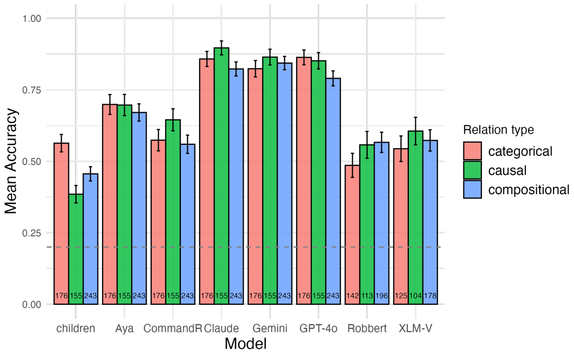


Figure 8: In children (as with adults) compositional relations are easier than causal. Pattern in LLM performance differs per model.

Author Index

Abdelgaber, Nahed, 348
Agerri, Rodrigo, 452
Alacam, Özge, 216
Alishahi, Afra, 126
Amsili, Pascal, 312

Baggio, Giosuè, 469
Bavaresco, Anna, 141
Beekhuizen, Barend, 571, 602
Bengoetxea, Jaione, 452
Briscoe, Ted, 540
Bunzeck, Bastian, 169
Buttery, Paula, 481, 502, 522

Cai, Zhenguang, 586
Candito, Marie, 312
Carvalho, Danilo, 12
Chandak, Yashashree, 201
Cheng, Chung Shing, 1
Cheng, Tsz Chung, 1
Chersoni, Emmanuele, 469
Choenni, Rochelle, 627
Chong, Cheuk Hei, 1
Chou, Hsin-Yu, 30
Cong, Jason, 268
Conlan, Owen, 233

Daelemans, Walter, 68
De Bruyne, Luna, 68
De Souza, Sydelle, 291
Debnath, Alok, 233
Delbari, Zahra, 557
Deng, Yunhan, 47
Doshi, Arham Vinit, 348
Doumas, Leonidas A. A., 291
Duan, Xufeng, 586
Duran, Daniel, 169

El-Naggar, Nadine, 540
Eshuijs, Leon, 105

Fernández, Raquel, 141
Fokkens, Antske, 105
Freitas, Andre, 12

Gamallo, Pablo, 419
Garcia, Marcos, 419
Garg, Amanmeet, 187

Gevers, Ine, 68
Goldman, Omer, 251
Gonzalez-Dios, Itziar, 452
Goriely, Zebulon, 502, 522
Graham, Yvette, 233
gu, Hongyi, 481
Gupta, Advay, 201

Hasan, Mohammad Rashedul, 334
Haslett, David A., 586
Hayat, Ahatsham, 334
Hsieh, Cho-Jui, 268
Hsieh, Shu-Kai, 30

Ishii, Taiga, 437

Jahan, Labiba, 348
Johnson, Tamar, 627
Junker, Simeon, 216

Kennedy, Mary, 377
Keshales, Dorin, 251
Khan, Bilal, 334
Kletz, David, 312
Krishnaswamy, Nikhil, 323
Kuribayashi, Tatsuki, 540

Lai, Kenneth, 323
Lam, Eugene, 1
Lau, Chaak-ming, 1
Lenci, Alessandro, 469
Li, Ying, 47
Lihaoran, Lihaoran, 47

Marez, Victor De, 68
Martinez, Helen, 334
Matsuzaki, Takuya, 397
Miyao, Yusuke, 437
Mollica, Francis, 291

Nath, Abhijnan, 323
Ngo, Hoang, 81
Nguyen, Dat Quoc, 81
Nguyen, Thi-Nhung, 81
Nicholls, Diane, 481
Nosaka, Ryota, 397

Obiso, Timothy, 323

Oseki, Yohei, 93, 407

Pavel, Hamza Reza, 348
Pavlick, Ellie, 156
Phung, Dinh, 81
Pilehvar, Mohammad Taher, 557
Pouw, Charlotte, 126
Prakriya, Neha, 268
Pustejovsky, James, 323

Radaelli, Matteo, 469

Schneider, Nathan, 365
Scivetti, Wesley, 365
Shen, Siqi, 187
Shutova, Ekaterina, 627
Sieker, Judith, 216
Someya, Taiga, 93
Sreekumar, Vishnu, 201
Stevenson, Claire E, 627
Strohmaier, David, 481
Sugimoto, Yushi, 407
Sun, Chen, 156
Sunny, Amal, 201
Suri, Rishi, 348

Temerko, Anna, 419
ter Veen, Mathilde, 627
Tsarfaty, Reut, 251
Tseng, Yu-Hsiang, 30
Tyen, Gladys, 481

van der Maas, Han, 627
Vegner, Ivan, 291
Vu, Thuy-Trang, 81

Wang, Baoxun, 47
Wang, Shihan, 105
Wang, Wenqing, 47
Wu, Bowen, 47

Xiao, Bei, 586

Yanaka, Hitomi, 93
Yat, Wong Chun, 1
Yen, Jui-Nan, 268
Yoshida, Ryo, 93, 407
Yu, Hoi On, 1
Yu, Jingsong, 47
Yuan, Zheng, 481
Yun, Tian, 156

Zarrieß, Sina, 169, 216
Zhang, Jia, 348
Zhang, Yingji, 12
Zuidema, Willem, 126