NALOMA 2025

The 5th Workshop on Natural Logic Meets Machine Learning (NALOMA)

Proceedings of the Workshop

August 4 - 8, 2025
Bochum, Germany

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 979-8-89176-287-9

Introduction

Welcome to the 5th edition of the Natural Logic Meets MAchine Learning workshop (NALOMA).

NALOMA continues to serve as a venue dedicated to bridging the gap between machine/deep learning approaches on the one hand, and symbolic/logic-based approaches to natural language understanding and reasoning on the other. A central focus of the workshop remains the development of hybrid approaches and the exploration of theoretical insights that shape and guide computational models of reasoning.

NALOMA took place in August 4-8 during ESSLLI 2025, hosted at Ruhr University Bochum, Germany. We are deeply grateful to the ESSLLI organizers for their support. The workshop was held over a period of five days, with time slots of about one and a half hours. This year's program featured three inspiring keynotes, five regular talks with accompanying archival papers included in this proceedings, and three contributed talks based on non-archival submissions.

We would like to thank all authors of archival or non-archival submissions, as well as the dedicated members of the program committee whose careful reviews ensured the quality of the workshop. Our thanks also go to our keynote speakers for sharing their expertise and vision.

As in prior years, NALOMA serves as a platform connecting the symbolic AI and logic communities with the machine learning community, with the dual purpose of promoting discussion and fostering joint research initiatives. We look forward to the collaborations and insights that will arise from this year's event.

NALOMA is endorsed by the Special Interest Group on Computational Semantics (SIGSEM), for which we are grateful.

Lasha Abzianidze and Valeria de Paiva, Program Co-Chairs

Organization

Program Co-Chairs

Lasha Abzianidze, Utrecht University

Valeria de Paiva, Topos Institute

Program Committee

Stergios Chatzikyriakidis, University of Crete

Katrin Erk, University of Texas at Austin

Hai Hu, Shanghai Jiao Tong University

Thomas Icard, Stanford University

Aikaterini-Lida Kalouli, Bundesdruckerei GmbH

Lawrence S. Moss, Indiana University

Hitomi Yanaka, University of Tokyo and Riken Institute

Keynote Talk

Understanding Complex Situation Descriptions

Aaron Steven White
University of Rochester

Abstract: We use natural language to convey information about situations: things that happen or stuff that is true. This ability is supported by systematic relationships between the way we conceptualize situations and the way we describe them. These systematic relationships in turn underwrite inferences that go beyond what one strictly says in describing a situation. The question that motivates this talk is how to design systems that correctly capture the inferences we draw about situations on the basis of their descriptions.

Classical approaches to this question-exemplified in their modern form by graph-based representations, such Uniform Meaning Representation-attempt to capture the situation conceptualization associated with a description using a symbolic situation ontology and to draw inferences on the basis of rules stated over that ontology. An increasingly popular alternative to such ontology-factored approaches are ontology-free approaches, which attempt to directly represent inferences about a situation as natural language strings associated with a situation description, thereby bypassing the problem of engineering a situation ontology entirely.

I discuss the benefits and drawbacks of these two approaches and present case studies in synthesizing them that focus specifically on how best to capture inferences about complex situations-i.e. situations, like building a house, that themselves may be composed of subsituations, like laying the house's foundations, framing the house, etc. I argue that we should ultimately strive for ontology-free representations but that the challenges inherent to reasoning about complex situations highlight the persistent benefits of situation ontologies in providing representational scaffolding for the construction and evaluation of such representations.

Bio: Aaron Steven White is an Associate Professor of Linguistics at the University of Rochester, with a secondary appointment in Computer Science and an affiliation with the Goergen Institute for Data Science. He directs both the Center for Language Sciences and the FACTS.lab (Formal and Computational Semantics Lab) at the University of Rochester. His research focuses on the development of large-scale, theoretically informed semantic annotation frameworks and natural language understanding systems.

Keynote Talk

How Can Large Language Model Become More Human?

Mehrnoosh Sadrzadeh
University College London

Abstract: Psycholinguistic experiments reveal that efficiency of human language use is founded on predictions at both syntactic and lexical levels. Previous models of human prediction exploiting LLMs have used an information theoretic measure called surprisal, with success on naturalistic text in a wide variety of languages, but under-performance on challenging text such as garden path sentences. This paper introduces a novel framework that combines the lexical predictions of an LLM with the syntactic structures provided by a dependency parser. The framework gives rise to an Incompatibility Fraction. When tested on two garden path datasets, it correlated well with human reading times, distinguished between easy and hard garden path, and outperformed surprisal.

Bio: Mehrnoosh is a professor of Computer Science at University College London. She holds a Royal Academy of Engineering Research Chair and leads a lab on mathematical and quantum methods in AI. Her research mainly focuses on studying logical and mathematical models of natural language, in particular, uisng algebraic grammars for syntax modeling and tensor spaces for semantics, often these methoids incorporating machine learning and quantum methods.

Keynote Talk

Understanding the Logic of Generative AI through Logic

Kyle Richardson
Allen Institute for AI

Abstract: Symbolic logic has long served as the de-facto language for expressing complex knowledge throughout computer science, owing to its clean semantics. Symbolic approaches to reasoning that are driven by declarative knowledge, in sharp contrast to purely machine learning-based approaches, have the advantage of allowing us to reason transparently about the behavior and correctness of the resulting systems. In this talk, we focus on the broad question: Can the declarative approach be leveraged to better understand and formally specify algorithms for large language models (LLMs)? We focus on formalizing recent direct preference alignment (DPA) loss functions, such as DPO, that are currently at the forefront of LLM alignment. Specifically, we ask: Given an existing DPA loss, can we systematically derive a symbolic expression that characterizes its semantics? We outline the details of a novel formalism we developed for these purposes. We also discuss how this formal view of preference learning sheds new light on both the size and structure of the DPA loss landscape and makes it possible to derive new alignment algorithms from first principles. Our framework and approach aim not only to provide guidance for the AI alignment community, but also to open up new opportunities for researchers in formal semantics to engage more directly with the development and analysis of LLM algorithms.

Bio: Kyle Richardson is a senior research scientist at the Allen Institute for AI (AI2) in Seattle. He works at the intersection of NLP and Machine Learning on the Aristo team, with a particular focus on generative AI and language models. Recently, he has been interested in using formal methods to better understand and specify algorithms for large language models. Prior to AI2 he was at the IMS and the University of Stuttgart, where he obtained his PhD in 2018.

Table of Contents

Program

Monday, August 4, 2025

17:00 - 17:05 *Opening Remarks*

17:05 - 18:00 *Keynote: Understanding Complex Situation Descriptions*
Aaron Steven White

18:00 - 18:25 *In the Mood for Inference: Logic-Based Natural Language Inference with Large Language Models*
Bill Noble, Rasmus Blanck and Gijs Wijnholds

Tuesday, August 5, 2025

17:00 - 17:30 *Implementing a Logical Inference System for Japanese Comparatives*
Yosuke Mikami, Daiki Matsuoka and Hitomi Yanaka

17:30 - 17:55 (non-archival) *MERGE: Minimal Expression-Replacement GEneralization Test for NLI*
Mădălina Zgreabăn, Tejaswini Deoskar and Lasha Abzianidze

Wednesday, August 6, 2025

17:00 - 17:55 *Keynote: How Can Large Language Model Become More Human?*
Mehrnoosh Sadrzadeh

17:55 - 18:20 *Unpacking Legal Reasoning in LLMs: Chain-of-Thought as a Key to Human-Machine Alignment in Essay-Based NLU Tasks*
Yu Ying Chu, Sieh-chuen Huang and Hsuan-Lei Shao

Thursday, August 7, 2025

17:00 - 17:25 *Dataset Creation for Visual Entailment using Generative AI*
Rob Reijtenbach, Suzan Verberne and Gijs Wijnholds

17:25 - 17:50 *Building a Compact Math Corpus*
Andrea Ferreira

17:50 - 18:15 (non-archival) *Automatic Evaluation of Linguistic Validity in Japanese CCG Treebanks*
Asa Tomita, Hitomi Yanaka and Daisuke Bekki

Friday, August 8, 2025

17:00 - 17:55 *Keynote: Understanding the Logic of Generative AI through Logic*
Kyle Richardson

17:55 - 18:20 (non-archival) *How Often does Natural Logic Actually Meet Machine Learning?*
Lasha Abzianidze

18:20 - 18:25 *Closing Remarks*

Unpacking Legal Reasoning in LLMs: Chain-of-Thought as a Key to Human-Machine Alignment in Essay-Based NLU Tasks

Ying-Chu Yu[1], Sieh-Chuen Huang[1], Hsuan-Lei Shao[2]*
[1]College of Law, National Taiwan University, Taipei, Taiwan
[2]Graduate Institute of Health and Biotechnology Law, Taipei Medical University, Taipei, Taiwan
eangelyu1278@gmail.com, schhuang@ntu.edu.tw, hlshao@tmu.edu.tw

Abstract

This study evaluates how Large Language Models (LLMs) perform deep legal reasoning on Taiwanese Status Law questions and investigates how Chain-of-Thought (CoT) prompting affects interpretability, alignment, and generalization. Using a two-stage evaluation framework, we first decomposed six real legal essay questions into 68 sub-questions covering issue spotting, statutory application, and inheritance computation. In Stage Two, full-length answers were collected under baseline and CoT-prompted conditions. Four LLMs—ChatGPT-4o, Gemini, Grok3, and Copilot—were tested. Results show CoT prompting significantly improved accuracy for Gemini (from 83.2% to 94.5%, $p < 0.05$) and Grok3, with moderate but consistent gains for ChatGPT and Copilot. Human evaluation of full-length responses revealed CoT answers received notably higher scores in issue coverage and reasoning clarity, with ChatGPT and Gemini gaining +2.67 and +1.92 points respectively. Despite these gains, legal misclassifications persist, highlighting alignment gaps between surface-level fluency and expert legal reasoning. This work opens the black box of legal NLU by tracing LLM reasoning chains, quantifying performance shifts under structured prompting, and providing a diagnostic benchmark for complex, open-ended legal tasks beyond multiple-choice settings.

1 Introduction

Legal reasoning presents unique challenges for Large Language Models (LLMs) due to the logic-intensive and statute-bound nature of legal texts. Existing evaluations often focus on multiple-choice formats that fail to capture the stepwise reasoning required in legal analysis. This study introduces a Chain-of-Thought (CoT) prompting strategy tailored for legal essay questions. By guiding LLMs through decomposed sub-questions, we aim to reveal how structured prompting enhances interpretability, aligns with human reasoning, and supports complex legal inference.

We propose a two-stage diagnostic evaluation to assess how CoT affects legal reasoning generalization. Stage One decomposes six real legal exam questions into 68 sub-questions evaluating fact recognition, statutory application, and logic chaining. Stage Two compares full-length responses under baseline and CoT-prompted conditions. Four LLMs are tested, and answers are scored by both a professor and a student, enabling analysis of human-machine agreement and misalignment across reasoning dimensions.

By moving beyond answer correctness toward an analysis of legal reasoning structure, this study contributes new methods for evaluating alignment, generalization, and interpretability in legal NLU tasks. It offers a scalable benchmark and experimental protocol to guide the development of more transparent and human-aligned legal language systems.

2 Related Work

With the advancement of Large Language Models (LLMs), their applications in the legal domain have grown rapidly, yielding promising results in tasks such as contract analysis, judgment summarization, legal consultation, and case prediction . To promote research in legal language processing, several benchmark datasets and evaluation platforms for LLMs have emerged in recent years, including LexGLUE (Chalkidis et al., 2021, 2020), LegalBench (Guha et al., 2023), and the COLIEE competition on statutory entailment and retrieval. These benchmarks primarily cover tasks such as multiple-choice questions, case classification, statute matching, and legal question answering. However, most of them focus on English-language corpora and closed-form problems, lack-

* Corresponding author. ORCID: 0000-0002-7101-5272

Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA), pages 1–7

ing the design needed to evaluate the type of open-ended, reasoning-intensive essay questions encountered in real-world legal practice. As noted by the creators of LegalBench, current benchmarks "still fall short of comprehensively evaluating the open-ended reasoning required in law school exams and legal writing assignments" (Guha et al., 2023), which often involve deep statutory subsumption and integrated legal analysis.

Chain-of-Thought (CoT) prompting has recently emerged as a promising strategy for improving multi-step reasoning and computation, validated on tasks such as math word problems and commonsense reasoning (Wei et al., 2022). CoT prompting has to be effective even without in-context examples: simple natural language cues like "Let's think step by step" can activate internal reasoning chains and significantly improve performance in zero-shot settings (Kojima et al., 2022). CoT has since been widely applied in mathematical reasoning (e.g., GSM8K, MATH), logic puzzles, scientific domains, and programming tasks. Prior studies consistently find CoT especially useful for tasks that require intermediate inference steps, as it helps maintain contextual coherence and supports longer, structured chains of reasoning.

Although legal reasoning itself is inherently a multi-step logical task, systematic analysis of CoT prompting in the legal domain remains limited. Some notable attempts include the KIS team's Interpretable CoT strategy in the COLIEE 2024 entailment task, which enhances interpretability in statutory subsumption through structured prompting (Fujita et al., 2024). Another example is the LegalGPT framework, which integrates CoT modules within a multi-agent architecture to simulate the collaborative logic of real-world legal practice (Shi et al., 2024).

Mainstream approaches to evaluating legal LLMs typically rely on automated metrics (e.g., accuracy, BLEU, F1-score) or multiple-choice style datasets to compare model performance. However, such closed-form evaluations fail to reflect the logical depth and reasoning quality required for open-ended generative tasks. Recent studies have begun to incorporate human evaluation to better assess consistency and subsumption performance in long-form legal QA. Representative systems such as Length-Controlled AlpacaEval (Dubois et al., 2024), MT-Bench (Zheng et al., 2023) , and PromptBench (Jiang et al., 2023) use human preference ratings or expert judgments as quality signals, augmented by ranking-based metrics, weighted averages, or Elo-style comparisons.

we conduct qualitative analysis by selecting cases with high inter-rater agreement to compare the logical structure of reasoning chains, thereby supplementing the current lack of process transparency and error traceability in legal LLM evaluation.

3 Experiment Design

3.1 Stage 1: Decomposed Reasoning Evaluation

3.1.1 Test Set Design

The test set used in this study consists of six essay questions, all adapted from previous Judicial Officer Examinations and the National Taiwan University Graduate Law School entrance exams in the field of Status Law. These questions cover a range of key topics, including the validity of marriage, division of marital property, legal guardianship, inheritance, bigamy, and adoption.

The test set comprises six essay questions adapted from Taiwan's judicial exams and law school admissions, focusing on key topics in Status Law such as marriage validity, inheritance, and guardianship. Each question was decomposed into multiple sub-questions—68 in total—targeting factual analysis, statutory application, and logic chaining. Status Law was selected due to its blended demands of symbolic reasoning, legal interpretation, and numerical computation, making it an ideal domain for evaluating LLMs' integrated legal reasoning capabilities.

The six essay questions selected for this study cover the following legal topics: **A. Validity of marriage, B. Division of residual marital property, C. Limitations on parental rights in relation to children's interests, D. Limited succession and creditor claims, E, Legal consequences of bigamy, F. Collation issues in inheritance distribution.**

The sub-questions are primarily framed as numerical problems and binary (yes/no) questions, with a few short-answer questions. Each sub-question presents a specific legal scenario and requires the LLM to provide a definitive judgment or calculation. To ensure stricter evaluation, this study adopts a rigorous scoring standard: if the model arrives at the correct final answer but misidentifies roles, relationships, or inheritance rankings within its reasoning, the response is marked incor-

rect. This prevents models from "guessing correctly" and emphasizes the need for accurate legal reasoning and comprehension.

3.1.2 Model Selection

The study evaluates four mainstream LLMs: ChatGPT-4o (OpenAI), Grok 3 (xAI), Gemini 1.5 Flash (Google), and Copilot (Microsoft, based on GPT-4-turbo). These models were selected to represent the current state-of-the-art offerings from the four major LLM platforms, balancing accessibility, popularity, and architectural diversity. All models were tested under identical formats and prompt templates to ensure fairness.

While some LLMs—such as ChatGPT or Gemini—exhibit emergent Chain-of-Thought (CoT) reasoning capabilities without explicit prompting, our experiments show that these models still often display under-reasoning behaviors, skipping intermediate legal logic steps or prematurely concluding without sufficient statutory justification.

Rather than designing a universal prompt for all models or modifying LLM parameters, our approach centers on a specialist-in-the-loop interaction. Here, a legal expert provides real-time validation at each reasoning stage, enabling targeted scaffolding and reducing cumulative error. This controlled feedback setup aims not merely to improve answer correctness, but to systematically expose and align the model's legal reasoning flow with doctrinal logic. Such design reflects pedagogical principles of legal education and offers a practical pathway for integrating LLMs into high-stakes, explainable legal workflows.

These sub-questions were manually constructed to reflect three core legal reasoning types: factual classification, statutory mapping, and inheritance computation. Each sub-task is framed as either a binary (yes/no), numerical, or short-form reasoning prompt. This design captures both surface-level legal recognition and deeper multi-step logic.

3.2 Stage Two: Comparison of CoT vs. Direct Answering

3.2.1 Response Format and Prompting Design

In Stage Two, models were tested on the full legal questions under two conditions: baseline answering (one-shot response) and Chain-of-Thought (CoT) prompting.

Responses were evaluated by a professor and a law student, blind to the model type and prompting condition. Scoring followed a 0–10 rubric covering issue coverage, statutory accuracy, and reasoning clarity. This design simulated realistic exam settings while enabling structured analysis of model reasoning quality.

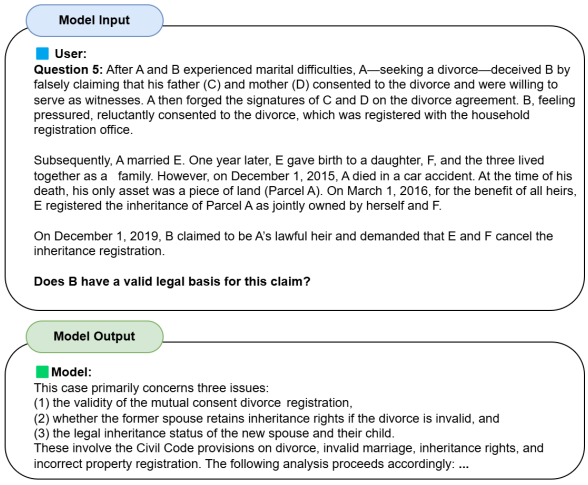


Figure 1: Example of Baseline Prompting

In contrast, Figure 2 depicts the Chain-of-Thought (CoT) prompting setting. The model is prompted to answer each sub-question step by step, with immediate user feedback at each stage to confirm or correct the response. This interactive scaffolding prevents early-stage reasoning errors from propagating and encourages structured legal analysis. After completing all sub-steps, the model is prompted to synthesize a full-length answer based on the verified intermediate results.

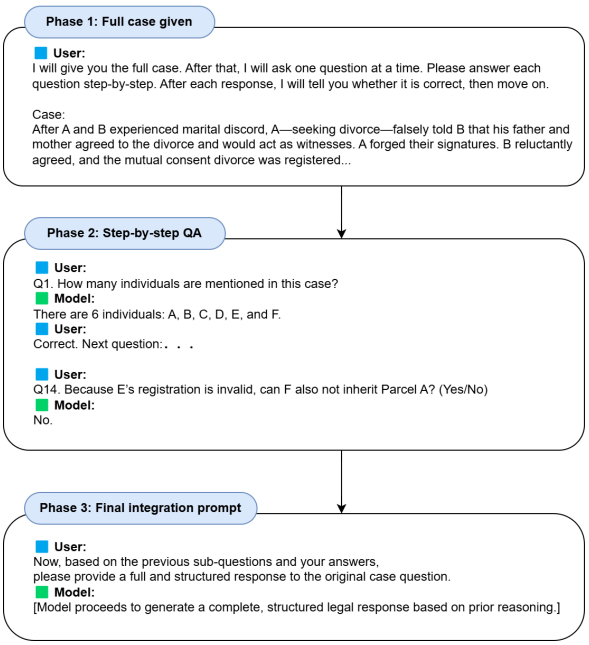


Figure 2: Example of CoT Prompting

This contrast illustrates how CoT prompting transforms the model's reasoning process from a monolithic, opaque response into an interactive, modular sequence of logic steps, allowing for clearer observation and evaluation.

3.2.2 Scoring Mechanism

Both response versions were evaluated by scorers with formal legal training, who rated each answer holistically on a 0–10 scale, with higher scores reflecting better overall quality. Scoring was based on three key criteria:

1. Issue coverage: Whether the model identified and addressed the key legal issues and factual disputes in the question.

2. Accuracy of statutory application: Whether the cited or applied legal provisions were correct and logically relevant.

3. Clarity of legal reasoning: Whether the reasoning was coherent, structured, and logically sound.

4 Experiment Result

4.1 Stage One Results: Accuracy in Decomposed Reasoning Evaluation

This stage also examined the effect of Chain-of-Thought (CoT) prompting on answer accuracy. The table below presents the accuracy performance of the four LLMs under baseline and CoT conditions, along with results from a paired-sample t-test:

Model	Raw Accuracy	CoT Accuracy	t-value	p-value
ChatGPT	0.842	0.866	-0.92	0.398
Gemini	0.833	0.9445	-3.71	0.013
Copilot	0.822	0.864	-2.14	0.089
Grok3	0.843	0.895	-2.98	0.031

Table 1: Accuracy comparison between raw and CoT prompting across LLMs

Overall, even though the unit of observation in this stage was the model's performance on decomposed sub-questions, the results clearly demonstrate that CoT prompting significantly enhanced accuracy for certain models. The effect was particularly pronounced in question types that involved multi-step reasoning and structured analysis, such as inheritance calculation, classification of legal status relationships, and precise statutory mapping. These findings provide a quantitative foundation for the holistic answer evaluations conducted in Stage Two.

4.2 Stage Two Results: Human Evaluation of Full-Length Responses Under CoT Prompting

The goal of the Stage Two experiment was to simulate realistic legal exam conditions and assess whether the overall quality of LLM-generated responses to unsegmented, full-length legal questions could be improved by introducing Chain-of-Thought (CoT) prompting. The same six Status Law questions used in Stage One were employed, but this time they were presented in their entirety—without decomposition—requiring the model to generate a complete answer in one go.

Model	**Raw** *Ave. Score*	**CoT** *Ave. Score*	**Average Improv.**	**Improv. Stud.** *Rater*	**Improv. Prof.** *Rater*	**Scoring Consistency** *(Pearson's r)*
ChatGPT	6.50	9.17	**+2.67**	+3.00	+2.33	0.716
Gemini	6.12	8.04	+1.92	+2.83	+1.00	0.853
Copilot	5.83	7.42	+1.58	+2.00	+1.17	0.752
Grok3	6.25	8.08	+1.83	+2.17	+1.50	0.835

Table 2: Human evaluation results under baseline vs. CoT prompting

4.2.1 Overall Model Scoring Results

The results show that all models demonstrated improved performance when CoT prompting was applied. Among them, ChatGPT exhibited the largest improvement (+2.67 points) and the most consistent performance across questions. Gemini and Grok3 also showed marked improvements, each with gains exceeding 1.8 points. Although Copilot lagged behind the other models in terms of raw scores, it too displayed consistent improvement under CoT prompting.

In terms of inter-rater agreement, Pearson correlation coefficients ranged from 0.71 to 0.85 across the four models, indicating a moderate to high level of scoring consistency between the two raters. Notably, Gemini and Grok3 achieved the highest consistency, suggesting particularly stable performance as evaluated by both expert and student raters.

Q.	Raw Avg.	CoT Avg.	Score Gain	p-value
1	4.50	7.75	+3.25	0.068
2	6.12	7.12	+1.00	0.430
3	6.25	6.38	+0.12	0.919
4	3.88	5.50	+1.62	0.080
5	3.88	7.00	+3.12	0.002
6	4.63	7.50	+2.88	0.011

Table 3: Average scores for each legal question under raw vs. CoT prompting

4.3 Qualitative Analysis of Model Responses on a Representative Question

To illustrate model reasoning differences, we selected Question 5 as a representative case based on high inter-rater agreement. Gemini's baseline response exhibited flawed assumptions and incorrect citations, such as misapplying Article 92 instead of the correct divorce statute. Its CoT-prompted version showed clearer structure and partial legal improvement, yet still missed key statutes and overgeneralized inheritance logic. This contrast highlights CoT's benefit in structuring legal reasoning, though gaps remain in precise statutory application and doctrinal subsumption (see Appendix A for question details).

Table 4 provides a visual comparison of key statutes that should be cited in an ideal answer like Figure 3.

Issue	Original Reasoning	CoT Reasoning
Divorce	Assumes validity; cites §92 (intent defect)	Finds formal defect; cites §1050
Remarriage	Treats E as lawful spouse	Void due to bigamy; E not in good faith
F's Inheritance	Assumes F is legitimate heir	F is non-marital but legally recognized
Registration	Both E and F must cancel	Only E is void; F retains right

Table 4: Comparison of original and CoT reasoning on four legal issues.

The original version erroneously cited Article 92, which pertains to the revocation of declarations of intent due to fraud. This reflects a misunderstanding of the legal nature of the problem—it misclassified the issue as a defect in intent rather than a formal defect that invalidates the divorce. Furthermore, it failed to mention several key statutes, including those governing bigamy and non-marital inheritance.

The CoT version correctly cited Articles 1050 and 1138, capturing part of the statutory logic. However, it omitted Article 767 and did not clearly reference the provisions governing non-marital children, resulting in a fragmented presentation of the statutory framework.

5 Conclusion and Limitations

5.1 CoT-on-CoT

This paper presents a two-stage diagnostic framework to evaluate how Large Language Models

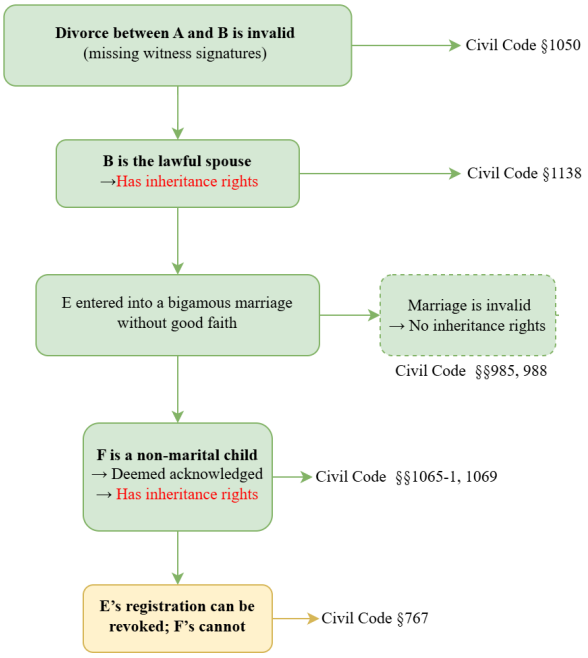


Figure 3: Correct legal reasoning flowchart

(LLMs) reason through legal essay questions in Taiwan's Status Law. By decomposing real exam questions into 68 sub-tasks and comparing full-length responses under baseline and Chain-of-Thought (CoT) prompting, we assess both micro-level reasoning and holistic legal understanding. Results show that CoT significantly improves accuracy in issue spotting, statutory application, and inheritance calculation, particularly for Gemini and Grok3.

Human evaluation further reveals enhanced clarity and structure in CoT-generated answers, though alignment gaps with expert persist. While our focus is not on prompt universalization, but explore CoT-on-CoT designs using models already trained with internal reasoning strategies, to examine whether reasoning stability or redundancy effects emerge. The CoT prompting can effectively improve the logical structure and issue coverage in model-generated responses.

Rather than fully opening the black box of LLM reasoning, our approach traces the model's internal chains of thought by eliciting and examining intermediate steps, thereby making its legal decision path more interpretable. We release a legally grounded benchmark and propose a generalizable evaluation methodology for open-ended, multi-step reasoning tasks.

5.2 Limitations and Future Work

This study has several limitations that open avenues for future work. First, while our results demonstrate that LLMs often produce incorrect reasoning even when their final answers are right, we do not yet offer a systematic typology of such reasoning failures. Future research could develop finer-grained error categories and explore how these missteps relate to different legal domains or prompt structures.

Second, although our methodology involves decomposing legal questions into sub-tasks, we have not formalized a reusable guideline for annotators or model developers to construct reasoning flowcharts. Creating such a protocol—potentially in the form of annotation templates or instructional schemas—could support replicability and improve human-LLM alignment in legal diagnostics.

Third, while we propose three evaluation dimensions (issue coverage, statutory application, reasoning clarity), we have not validated their generalizability across legal domains beyond Taiwanese Status Law. We believe these dimensions are transferable, but further experiments on multilingual or cross-jurisdictional datasets (e.g., U.S. torts, Japanese family law) are needed to assess the framework's robustness and scalability.

Finally, our current evaluation focuses on a limited set of essay-style questions, and has not been tested at scale. Integration with existing legal benchmarks (e.g., LegalBench, COLIEE) could allow broader adoption, while future work may automate sub-question generation or integrate CoT supervision into fine-tuning pipelines.

References

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. 2021. Lexglue: A benchmark dataset for legal language understanding in english. *arXiv preprint arXiv:2110.00976*.

Yuan Chen, Ronglai Shen, Xiwen Feng, and Katherine Panageas. 2024. Unlocking the power of multi-institutional data: Integrating and harmonizing genomic data across institutions. *Biometrics*, 80(4):ujae146.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Masaki Fujita, Takaaki Onaga, and Yoshinobu Kano. 2024. Llm tuning and interpretable cot: Kis team in coliee 2024. In *JSAI International Symposium on Artificial Intelligence*, pages 140–155. Springer.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, and 1 others. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in Neural Information Processing Systems*, 36:44123–44279.

Yue Jiang, Siyu Zheng, Xin Chen, Hao Peng, Xiang Lin, and 1 others. 2023. Promptbench: Towards evaluating the robustness of large language models with prompt-based benchmarks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Juanming Shi, Qinglang Guo, Yong Liao, and Shenglin Liang. 2024. Legalgpt: Legal chain of thought for the legal large language model multi-agent framework. In *International Conference on Intelligent Computing*, pages 25–37. Springer.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Siyu Zheng, Bo Peng, Hao Peng, Zhen Zhang, Yao Shen, Zhuohan Li, Weifeng Du, Tao Yu, Tianyi Zhang, Xiang Lin, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Appendix: Test Set and Sub-question Decomposition

Design Rationale. The test set includes six legal essay questions adapted from Taiwan's Judicial Officer Examinations, National Taiwan University Law Graduate Admissions Exams, and final assessments in Identity Law. Each question targets core legal topics and is decomposed into sub-questions assessing factual comprehension, legal classification, statutory application, and logical reasoning.

Question: Bigamy and Inheritance Disputes

Question A and B registered a consensual divorce based on forged witness signatures. A later married E and had a daughter F. A died in a car crash, and E registered inheritance jointly with F. B later claims as the legal heir. Is her claim valid?

Sub-questions

1. How many individuals are involved?
2. Were A and B legally married? (Yes/No)
3. How many times did A marry?
4. Is divorce valid without witness verification? (Yes/No)
5. Was A and B's divorce legally valid? (Yes/No)
6. If A never divorced B, is marriage to E valid? (Yes/No)
7. If E was unaware, does good faith validate marriage? (Yes/No)
8. Is B a legal heir? (Yes/No)
9. Is E a legal heir? (Yes/No)
10. Are C and D (A's parents) legal heirs? (Yes/No)
11. Is F excluded due to being non-marital? (Yes/No)
12. Does E's co-ownership registration remain valid? (Yes/No)
13. If E is not a legal heir, is her registration invalid? (Yes/No)
14. If E's registration is invalid, does it affect F's inheritance? (Yes/No)

Dataset Creation for Visual Entailment using Generative AI

Rob Reijtenbach
Leiden University
rob.reijtenbach@gmail.com

Suzan Verberne
Leiden University

Gijs Wijnholds
Leiden University

(s.verberne|g.wijnholds)@liacs.leidenuniv.nl

Abstract

In this paper we present and validate a new synthetic dataset for training visual entailment models. Existing datasets for visual entailment are small and sparse compared to datasets for textual entailment. Manually creating datasets is labor-intensive. We base our synthetic dataset on the SNLI dataset for textual entailment. We take the premise text from SNLI as input prompts in a generative image model, Stable Diffusion, creating an image to replace each textual premise. We evaluate our dataset both intrinsically and extrinsically. For extrinsic evaluation, we evaluate the validity of the generated images by using them as training data for a visual entailment classifier based on CLIP feature vectors. We find that synthetic training data only leads to a slight drop in quality on SNLI-VE, with an F-score 0.686 compared to 0.703 when trained on real data. We also compare the quality of our generated training data to original training data on another dataset: SICK-VTE. Again, there is only a slight drop in F-score: from 0.400 to 0.384. These results indicate that in settings with data sparsity, synthetic data can be a promising solution for training visual entailment models.

1 Introduction

Natural language inference (NLI) is a classification problem for pairs of two texts, a premise and a hypothesis. The pair is labeled as *entailment* (the premise entails the hypothesis), *neutral* or *contradiction* (the hypothesis contradicts the premise). In visual entailment (VE) tasks (Xie et al., 2019), the premise is substituted by an image, while the hypothesis is still in text form.

In order to create and train effective models for VE, large datasets are needed. While datasets of images combined with hypotheses and labels do exist, they are relatively small and sparse compared to datasets for textual entailment. Existing datasets are SNLI-VE (Xie et al., 2019) and SICK-VTE (Iokawa et al., 2024) which are both based on NLI datasets and which were created by manual labor leveraging Amazon Mechanical Turk workers. In this paper we evaluate the use of generative AI for VE dataset creation which would allow cheaper and easier dataset creation. This is done by first generating a synthetic dataset, of which we then verify the validity. We introduce a synthetic version of the SNLI-VE dataset called Synthetic-NLI-VE and show how models trained on this dataset have similar performance when tested on real data compared to models trained on real data.

In summary, the contributions of this paper are threefold: (1) we present the new dataset Synthetic-NLI-VE[1]; (2) we find that the performance of models trained on the generated dataset have similar performance compared to models trained on real data; (3) A cross-data evaluation shows that generalizability of visual entailment models to a different dataset is poor, whether or not the training set was generated or original.

2 Related work

Visual entailment and dataset creation The idea of visual entailment was first proposed by Xie et al. (2019). For this task they introduce the Explainable Visual Entailment (EVE) model, based on Attention Visualization. In the same paper the authors introduce the SNLI-VE dataset (Section 3). Antol et al. (2015) introduced a dataset for visual question answering (QA). They used the Microsoft Common Objects in Context (MS COCO) dataset (Lin et al., 2014) as a starting point: ∼200k images of real-world scenes with 5 captions per image. They added 50k images of abstract scenes for which they also collected 5 captions per image.

Marelli et al. (2014) created the SICK dataset. SICK (*sentences involving compositional knowl-*

[1] https://huggingface.co/datasets/robreijtenbach/Synthetic-NLI-VE

Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA), pages 8–17

edge) contains sentence pairs with both relatedness scores and entailment labels. This dataset was created by pairing the Flickr8K dataset (Hodosh et al., 2013) and the SemEval-2012 STS data (Agirre et al., 2012) and having Amazon Mechanical Turk workers annotate them with both similarity scores and entailment labels. Wijnholds and Moortgat (2021) created the Dutch version of SICK using a semi-automatic translation. Bowman et al. (2015) introduced the SNLI dataset on which the aforementioned SNLI-VE was based, with as motivation that the SICK dataset is too small and not balanced enough. For SNLI they created a balanced dataset of around $\sim$500k sentence pairs compared to the $\sim$10k in the SICK dataset.

There are also efforts made to improve existing datasets. This was already the case with Goyal et al. (2017), who improved and extended the VQA dataset resulting in the VQA-v2 dataset. The dataset was improved by, among other things, reducing bias and extended it by adding more images. This has also been done for the SNLI-VE dataset by Do et al. (2021) who created the e-SNLI-VE-2.0.

Synthetic data Unlike the largely human made datasets that were previously discussed, the CLEVR dataset (Johnson et al., 2016) is automatically generated. This dataset contains images of abstract shapes combined with automatically generated questions. The images were created by randomly sampling a scene graph and rendering it using the open-source 3D rendering software Blender.

Yuan et al. (2024) proposed an evaluation framework for assessing synthetic data generated by large language models (LLMs). This framework includes measures for fidelity, utility and privacy. In this work, we only focus on the fidelity and utility of the generated data.

Some research suggests that using synthetic datasets for model training could have a negative effect on performance in the future, if generated datasets are used for training computer vision models (Hataya et al., 2023). As opposed to synthetic datasets used to train generative models, the images that we generate are used to train classification models. Furthermore, these classification models are evaluated on original data, ensuring good real world generalizability.

3 Data

In this work we use two datasets which we briefly describe in this section.

SNLI-VE This was introduced by (Xie et al., 2019), by combining the SNLI dataset (Bowman et al., 2015) with the Flickr30k dataset (Young et al., 2014). The Flickr30k dataset was created by taking 31,783 photos of everyday activities which were harvested from Flickr. Each image receives 5 different captions resulting in 158,915 captions in total. Figure 3 in the appendix shows an example of an image and its captions.

The SNLI dataset (Bowman et al., 2015) is a well known dataset specifically created for natural language inference. In short, it was constructed by having Amazon Mechanical Turk workers generate 3 hypotheses per caption, where captions came from the Flickr30k dataset. From this, Xie et al. (2019) could therefore create the SNLI-VE dataset by replacing each premise by the original corresponding image. The dataset contains a total of 31,783 images, 157,567 premises and 565,286 hypotheses.

SICK-VTE Along the lines of the creation of SNLI-VE, Iokawa et al. (2024) introduces SICK-VTE, a visual entailment version of (a subset of) the SICK dataset (Marelli et al., 2014), but with an additional multilingual component, including also the Dutch (Wijnholds and Moortgat, 2021) and Japanese (Yanaka and Mineshima, 2022) translations of the SICK dataset. The construction of the original SICK dataset was based on sentence transformation rules over image captions instead of human-generated hypothesis. By construction the dataset contains only cases of Entailment and Contradiction: for 488 unique images there are 2,899 sentence pairs, with 1,930 examples of Entailment and 969 examples of Contradiction.

4 Methods

We generate a synthetic dataset as described in §4.1. We then report on the intrinsic evaluation of image quality by comparing the generated images directly with the original images based on a similarity analysis in §4.2. Finally, we perform extrinsic evaluation of synthetic data, comparing it to original data for visual entailment model training in §4.3.

4.1 Image Generation

Our approach for creating the generated dataset is to use the premise text from SNLI as input prompts in a generative model, creating an image for every premise caption. This results in a dataset similar to SNLI-VE, however, instead of multiple premises

referencing the same image, here the resulting dataset has a unique image for every premise. We refer to the generated images as *child images* to express the fact that they were indirectly derived from an original *parent image*. Examples of generated child images are shown in Figure 1.

Our choice of generative model is Stability AI's Stable Diffusion[2]. The ability to run the model locally as opposed to the cloud based solutions from OpenAI and Midjourney was essential for generating the large amount of images necessary for our work.

The chosen resolution was square images of 512x512 pixels as this is the image size Stable Diffusion was trained on and it is close to the average image size of the original SNLI-VE dataset.[3] The checkpoint chosen for this research is Realistic Vision v51[4] which was finetuned for generating photorealistic images.

4.2 Intrinsic evaluation

To assess intrinsic image quality we rely on two measures. As an initial verification we compute pairwise cosine similarity between the CLIP feature vectors of original and generated images and assess the distribution of these values, expecting to see a normal distribution.

Secondly, we use ranked similarity scores over the full dataset to inspect whether, for a given original image, the 5 generated images for it will appear as highly similar or not. We specifically use recall@k and precision@k for evaluation:

In the ranking problem in this work, we take the query to be an original image, and the ranked list of documents to be the 100 most similar generated images as determined by cosine similarity. The relevance function is now binary, returning 1 for an image that was indeed generated from one of the captions of the original image, and 0 otherwise.

For precision@k, we divide the true positives by the number of retrieved images.

4.3 Extrinsic evaluation

We test the validity of the generated images by using them as training data for a classifier to learn the visual entailment classification problem. The approach for this experiment is based on Song et al. (2022) who proposed using CLIP for visual entailment. Their method includes taking the CLIP feature vector of both the premise image and the hypothesis text, fusing these according to Equation 1 and training an MLP on this fused vector representation to output the correct entailment label.

$$\mathsf{fuse}(v_1, v_2) = [v_1, v_2, v_1{+}v_2, v_1{-}v_2, v_1{\cdot}v_2] \quad (1)$$

The input dimension for this perceptron is 2560 which is a direct result of the output size of the fuse function. The fuse function concatenates the feature vector of the image, the feature vector of the hypothesis, the sum of these two vectors as well as the difference between these vectors and finally the product of these vectors. This results in a total of five vectors that are concatenated and with each vector having a size of 512 numbers, the result has a length of $5 * 512 = 2560$.

The resulting vector is used as an input for the MLP which has one hidden layer of size 250. After experimenting with different layer sizes, the size of this hidden layer did not seem to affect the accuracy of the classifier but had an impact on the computational performance. After this one hidden layer the network only has one more layer which is the output layer. This output layer has a size of 3 corresponding to the three possible labels: entailment, neutral, contradiction.

We use this method to train classifiers on both the the original images and the generated images of the SNLI-VE dataset. These classifiers are then tested on the original as well as on the generated test sets, after which their performance is compared. Note that absolute performance of the classifier is not the primary goal. Rather, we are interested in the relative performance of a classifier trained on generated images compared to a classifier trained on real images. We, however, aim for good performance of both as this yields the most accurate data to compare between these two.

5 Experiments and Results

In this section we first report on the results for the intrinsic evaluation (§5.1), after which we discuss the downstream performance in the Visual Entailment task (§5.2), and finally we discuss the results of transferring the Visual Entailment model to the SICK-VTE dataset (§5.3).

[2] https://github.com/Stability-AI/generative-models

[3] The mean width and height were 459 and 395 respectively, and the standard deviations were 67 for width and 74 for height with both having a maximal value of exactly 500.

[4] https://huggingface.co/stablediffusionapi/realistic-vision-v51

A wedding party walks out of a building.

The group of people are assembling for a wedding.

A man and woman dressed for a wedding function.

Figure 1: Three examples of generated images based on three of the captions in Figure 3.

5.1 Intrinsic evaluation

The starting point of our intrinsic comparison is the cosine similarity distribution for images in the development and test set of the SNLI-VE dataset and its generated child images. Each original image is compared to all the generated images and the similarity scores are saved. We found that the similarity values follow a normal distribution for both the development and test set. The mean for both sets is 0.465 with a standard deviation of ~ 0.085 This is also illustrated in Figure 4 in the appendix.

Ranked similarity After assessing the similarity distribution between original and generated images, we report on the recall@k and precision@k curves. Initially, we computed average recall@k and precision@k values for $k = 100$, which reveals that on average only 1.6 of the 100 most similar synthetic images to the real images were based actually generated based on one of the premises accompanying that real image. These results stem from the fact that finding the 100 most similar out of ∼160k generated images will likely not result in finding all of the 5 images that are relevant. This is illustrated in Figure 2 where an image is shown together with the most cosine similar generated image which is not one of its child images. These two images could be considered rather similar by a human. It is likely that there are more images in the collection that are similar than only the child images, making the recall@k measure an underestimation of the real quality of the generated images. The recall@k and precision@k curves for this setting are in Figures 5a and 5b in the appendix.

To get a fairer picture of the similarity evaluation, we recalculate recall@k and precision@k curves for a sampled version of the data which is needed as the train set is large very large compared to the dev and test set, which are only 1000 original images each. We randomly sample 1000 examples from the train set of SNLI-VE, and consequently calculate recall@k and precision@k values for train, development, and test sets separately, each time considering 1000 original images and its ∼5000 generated child images. The resulting plots for the recall@k and precision@k of the samples are in Figure 6b and Figure 6a in the appendix. We find that the average success rate is between 3.5 and 4 out of the five possible relevant images, indicating that most of the relevant real images are found within the first 100 most similar generated images.

For completeness, we include the variance of the recall and precision curves of the samples in Figure 7 in the appendix where one standard deviation above and below each curve is marked.

Train set	Original	Generated
Original	70.3% / 0.703	71.1% / 0.710
Generated	68.9% / 0.686	73.2% / 0.732

Table 1: Accuracies/F1 scores of both models on both test sets of SNLI-VE.

5.2 Extrinsic Evaluation: Classification

We train both a model on the dataset of original images, and a model on the dataset of generated images, using the same train/dev/test split as suggested for the SNLI-VE dataset. We trained the model for 100 epochs and selecting the epoch for which the model performs highest on the development set, which was saved for evaluating on the test set. The accuracy and loss on the training set and dev set are shown in the appendix in Figure 8a and 8b and Figure 9a and 9b respectively.

We report accuracies and F1 scores in Table 1. We observe the best overall performance when using the model trained on generated data evaluated on the generated data as well. This suggests that

(a) Original

(b) Generated

Figure 2: An example of an image and a generated image which looks similar but is not considered relevant as the generated image is not a child of the original image in this evaluation. The original image (a) had 5 captions in the dataset written by 5 different workers. Image (b) was generated for the caption "A group of young men have finished their drinks while sitting at a table in a restaurant ."

the generated images and their classification has less variability compared to the original data. We also see that the model trained on original images performs better on the generated test set than it does on the original test set. This could suggest that the generated test set is "easier" to classify. Lastly, and most importantly, we do see that the model trained on generated data and tested on original data has a somewhat lower performance in this experiment, but the difference is small. It suggests that synthetic training data results in slightly worse performance in real world tasks.

5.3 Cross-data generalizability

The final part of the experiments evaluate the performance of the trained models when they are tested on another dataset, in this case the SICK-VTE dataset. As discussed in Section 3, SICK-VTE and its synthetic counterpart do not contain any neutral examples. To train visual entailment models, having neutral examples would be essential however for the purpose of testing the generalizability pretrained models, a dataset with neutral examples is preferred.

The experimental setup is similar to that of the classification experiment in Section 5.2, except that we now reuse the trained models from the prior experiment as we assess transfer capabilities. Both of the trained models were tested on the original SICK-VTE dataset and, for completeness, also on the generated version of SICK-VTE. Similar to the previous experiment, we report both accuracy and F1 scores in Table 2. Note that, in contrast to the results on the SNLI-VE dataset, accuracy and F1 scores diverge, due to label imbalance in SICK-VTE.

Train set	Original	Generated
Original	50.7% / 0.400	51.4% / 0.391
Generated	47.2% / 0.384	47.6% / 0.384

Table 2: Accuracies/F1 scores of both models on the SICK-VTE datasets.

We find that performance is relatively poor, given a majority baseline of 0.6657 for a model only predicting Entailment. This result is in line with the findings of Talman and Chatzikyriakidis (2019), who found similar issues when transfering models trained on the SNLI dataset to the SICK dataset. Secondly, we can conclude that the model trained on generated data performs slightly worse compared to the model trained on original data. This is in line with the findings in the previous experiment (§5.2).

6 Conclusion

In this paper we introduced a synthetic VE dataset Synthetic-NLI-VE. The dataset proved to have similar utility compared to the dataset it was based on while being far less costly to create. This also proves the viability of using generative AI to create datasets for the VE task, whereby we pave the way for future research into using synthetic data for VE dataset creation. As future work we propose changing the single set of parameters for the generation model to a variety of different values. Secondly, generating more than one image per caption could result in better training data compared to the one image per caption dataset we generated. Lastly, evaluating different classification algorithms could further strengthen the findings.

Limitations

Our experiments are limited evaluation for the CLIP model, and the findings might be different for other visual entailment models.

We investigated cross-data generalizability in synthetic VTE datasets. One limitation of our experiments is that both SNLI-VE and SICK-VTE are created based on Flickr30K, which makes them relatively more similar to each other than datasets based on other sources, such as NLVR and NLVR2.[5] We leave this cross-domain evaluation for future work.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. 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 385–393, Montréal, Canada. Association for Computational Linguistics.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

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.

Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-snli-ve-2.0: Corrected visual-textual entailment with natural language explanations. *CoRR*, abs/2004.03744.

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. *CoRR*, abs/1612.00837:6325–6334.

Ryuichiro Hataya, Han Bao, and Hiromi Arai. 2023. Will large-scale generative models corrupt future datasets? In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 20498–20508. IEEE.

Micah Hodosh, Peter Young, and Julia Hockenmaier. 2013. Framing image description as a ranking task: data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47(1):853–899.

Nobuyuki Iokawa, Gijs Wijnholds, and Hitomi Yanaka. 2024. Multilingual visual-textual entailment benchmark with diverse linguistic phenomena. *Proceedings of the Annual Conference of JSAI*, JSAI2024:4C3GS1104–4C3GS1104.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2016. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

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).

Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. 2022. CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6088–6100, Dublin, Ireland. Association for Computational Linguistics.

Aarne Talman and Stergios Chatzikyriakidis. 2019. Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy. Association for Computational Linguistics.

Gijs Wijnholds and Michael Moortgat. 2021. SICK-NL: A dataset for Dutch natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1474–1479, Online. Association for Computational Linguistics.

Ning Xie, Farley Lai, Derek Doran, and Asim Kadav. 2019. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706.

Hitomi Yanaka and Koji Mineshima. 2022. Compositional evaluation on Japanese textual entailment and similarity. *Transactions of the Association for Computational Linguistics*, 10:1266–1284.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

[5] https://lil.nlp.cornell.edu/nlvr/

Yefeng Yuan, Yuhong Liu, and Liang Cheng. 2024. A multi-faceted evaluation framework for assessing synthetic data generated by large language models. *Preprint*, arXiv:2404.14445.

Appendix

Additional figures are on the following pages.

- A bearded man, and a girl in a red dress are getting married.
- A wedding party walks out of a building.
- The group of people are assembling for a wedding.
- A man and woman dressed for a wedding function.
- A woman holds a man's arm at a formal event.

Figure 3: One of the $\sim$30k photos and its 5 accompanying captions from the SNLI dataset.

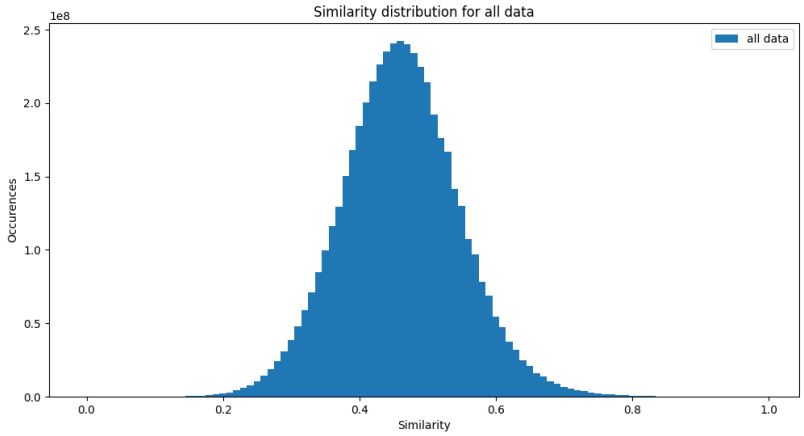
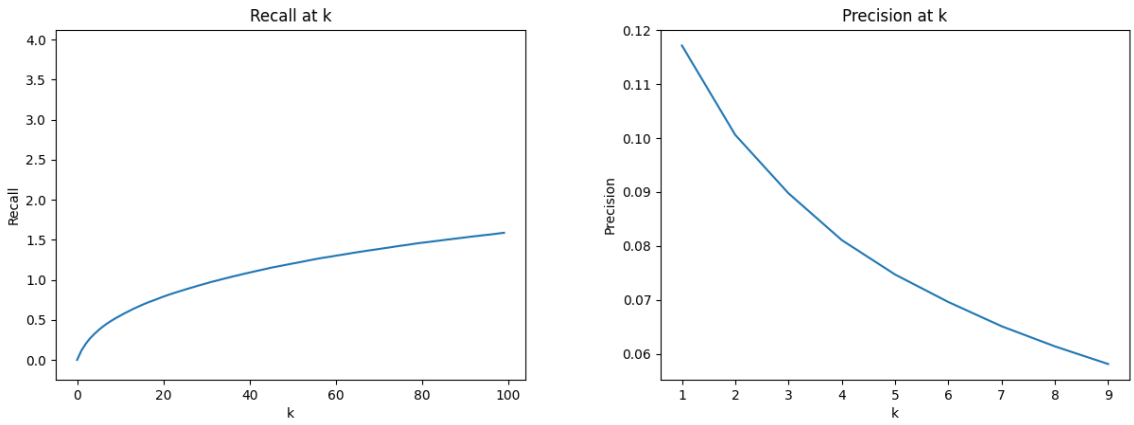
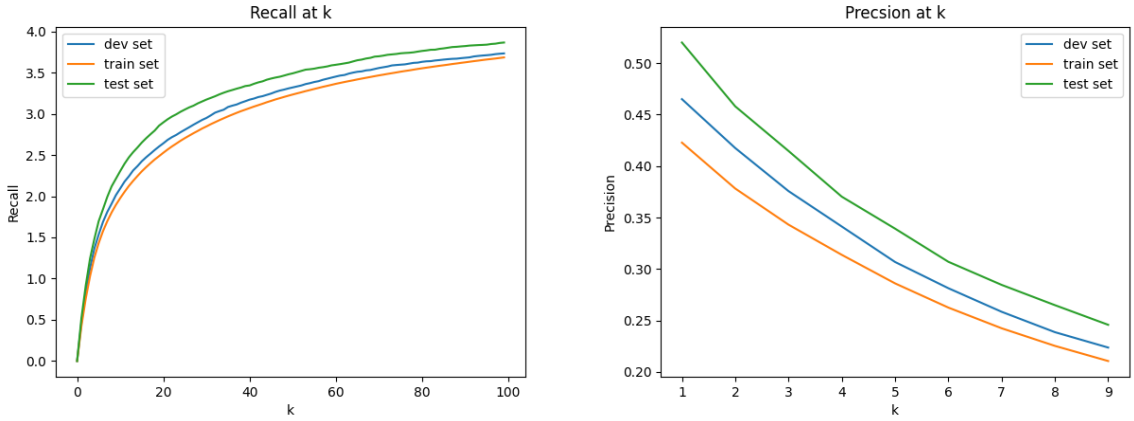


Figure 4: Cosine similarity values for the dataset, showing the expected normal distribution.



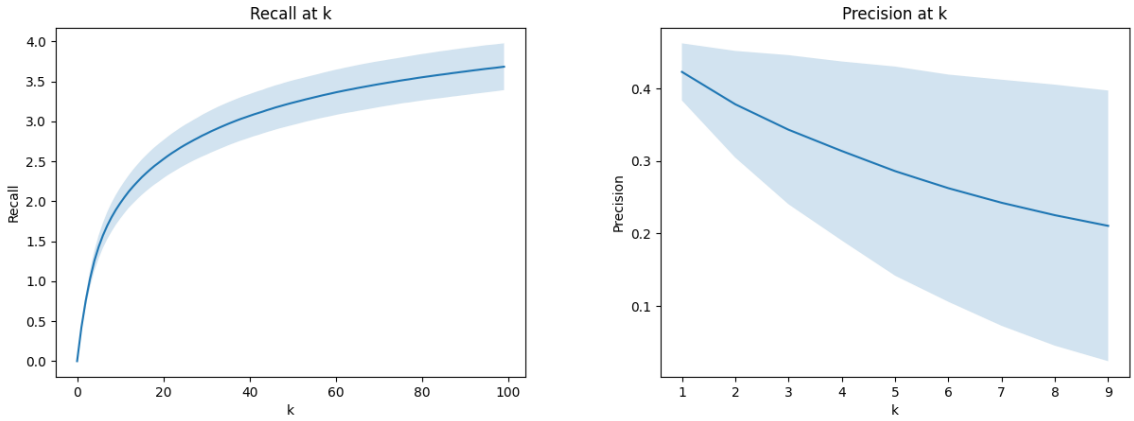
(a) (b)

Figure 5: Recall (a) and precision (b) curves, calculated as averaged over the full dataset of images.



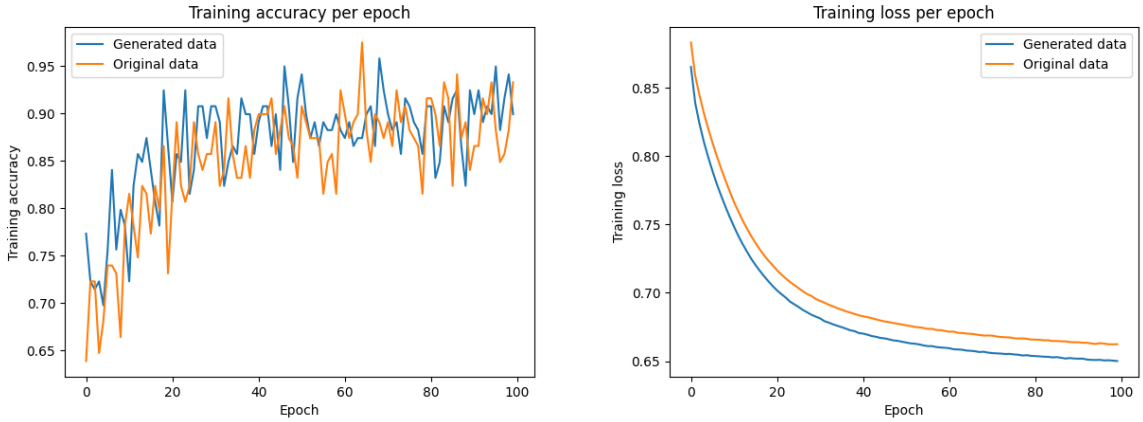
(a) (b)

Figure 6: Precision and recall curves where the train set is sampled in samples of 1000 images.



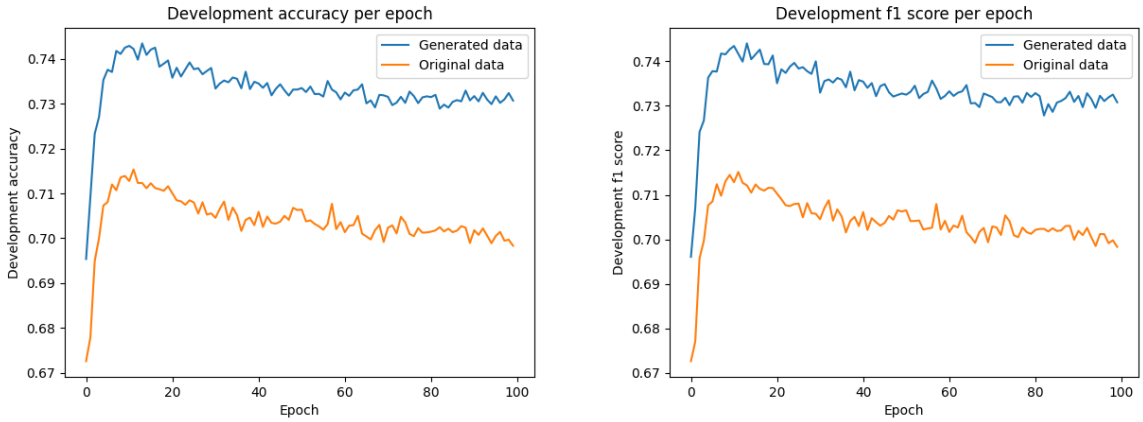
(a) (b)

Figure 7: Average precision and recall curves with one standard deviation.



(a) (b)

Figure 8: Performance on the training set during training.



(a) (b)

Figure 9: Performance on the development set after each epoch.

Implementing a Logical Inference System for Japanese Comparatives

Yosuke Mikami[1,2] **Daiki Matsuoka**[1,2] **Hitomi Yanaka**[1,2]
[1]The University of Tokyo
[2]Riken
{ymikami, daiki.matsuoka, hyanaka}@is.s.u-tokyo.ac.jp

Abstract

Natural Language Inference (NLI) involving comparatives is challenging because it requires understanding quantities and comparative relations expressed by sentences. While some approaches leverage Large Language Models (LLMs), we focus on logic-based approaches grounded in compositional semantics, which are promising for robust handling of numerical and logical expressions. Previous studies along these lines have proposed logical inference systems for English comparatives. However, it has been pointed out that there are several morphological and semantic differences between Japanese and English comparatives. These differences make it difficult to apply such systems directly to Japanese comparatives. To address this gap, this study proposes ccg-jcomp, a logical inference system for Japanese comparatives based on compositional semantics. We evaluate the proposed system on a Japanese NLI dataset containing comparative expressions. We demonstrate the effectiveness of our system by comparing its accuracy with that of existing LLMs.

1 Introduction

Natural Language Inference (NLI) (Bowman et al. 2015) is the task of determining the entailment relation between premise and hypothesis sentences. In particular, this paper focuses on inferences involving comparative expressions (e.g., *heavier*, where the comparative morpheme *-er* is attached). In (1), for example, the premise (1a) and (1b) entail the hypothesis (1c).

(1) a. John is heavier than Bob.
b. Bob is heavier than 70 kg.
c. John is heavier than 70 kg.
(entailment)

Inferences involving comparatives like (1) are challenging to an NLI system because the system needs to correctly understand the meaning of the quantity expression "70 kg" and the comparative relation between John's and Bob's weights.

There are two main approaches to NLI. One is a deep learning (DL)-based approach. Large Language Models (LLMs), such as GPT-4o,[1] have been performing accurately in various tasks, including NLI. However, recent works (She et al. 2023, Liu et al. 2023, Parmar et al. 2024) have pointed out that even such models have difficulties in handling problems involving logical connectives such as negation and quantification. This fact indicates that DL-based models still have room for improvement.

The other approach to NLI is a logic-based approach (Abzianidze 2015, Mineshima et al. 2015, Bernardy and Chatzikyriakidis 2017, Hu et al. 2020, Bernardy and Chatzikyriakidis 2021), in which mathematical logic is utilized to perform NLI involving various logical expressions robustly. In particular, inference systems based on compositional semantics have achieved high performance on NLI problems composed of lexical, syntactic, and semantic phenomena. As for comparatives, Haruta et al. (2022) proposed a logical inference system for English comparatives based on *Combinatory Categorial Grammar* (CCG, Steedman 2000) and *degree semantics* (Cresswell 1976, Klein 1980). However, we cannot apply the system directly to Japanese comparatives because of morphological and semantic differences between Japanese and English comparatives, which we will describe in detail in Section 4.

In this study, we aim to develop a logical inference system for Japanese comparatives based on CCG and degree semantics. Inspired by the logical inference system for English comparatives proposed by Haruta et al. (2022), our system, named ccg-jcomp, compositionally derives the semantic

[1] https://openai.com/index/gpt-4o-system-card/

Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA), pages 18–32

Sentence	Semantic Representation
John is heavy.	$\mathsf{heavy}(\mathsf{john}, \theta)$
John is heavier than 70 kg.	$\exists d.\,(\mathsf{heavy}(\mathsf{john}, d) \wedge d > 70\mathrm{kg})$
John is heavier than all the student.	$\forall x.(\mathsf{student}(x) \rightarrow \exists d.(\mathsf{heavy}(\mathsf{john}, d) \wedge \neg\mathsf{heavy}(x, d)))$

Table 1: Basic semantic representations for comparatives

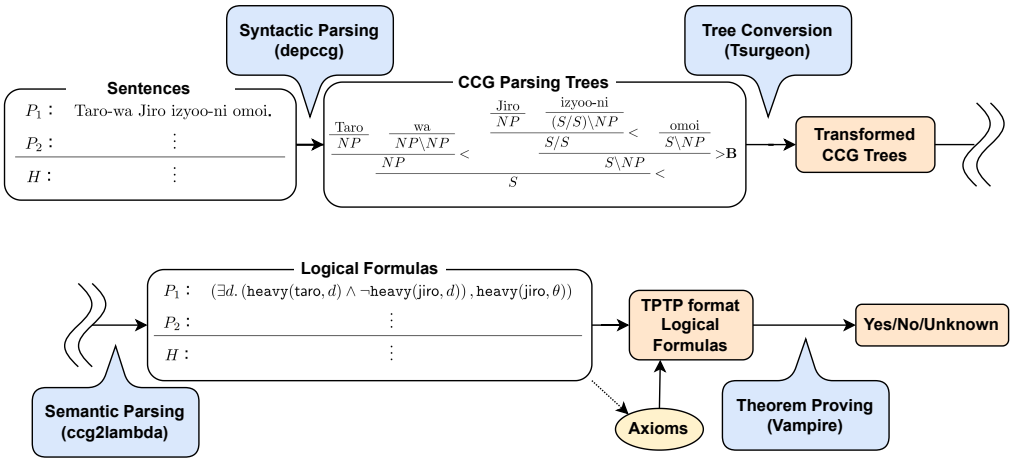


Figure 1: Overview of the proposed system

representations (i.e., the logical formulas representing the sentence meanings) of Japanese sentences through syntactic and semantic parsing and judges the entailment relation using a theorem prover. Further, we implement syntactic and semantic analyses to systematically handle some phenomena specific to Japanese comparatives.

We experiment with JSeM (Kawazoe et al. 2017), a Japanese NLI dataset containing problems involving comparatives. We compare the performance of our system with GPT-4o and some Japanese LLMs. Our experiment shows that our proposed system outperforms all of them in accuracy on the dataset.

Our contributions are as follows:

1. We compositionally derive the semantic representations of Japanese sentences containing some comparative expressions based on CCG and degree semantics.

2. We implement ccg-jcomp, a logical inference system for Japanese comparatives.[2]

3. We demonstrate the effectiveness of our proposed system through experiments on a Japanese NLI dataset involving comparatives.

[2]Our system is available for research use at https://github.com/ynklab/ccg-jcomp

2 Degree Semantics

In our study, we adopt a theoretical framework called degree semantics, which allows us to analyze the meanings of gradable adjectives and comparatives formally. Its basic idea is to treat a gradable adjective as a binary predicate that takes an entity and a degree as arguments. For instance, "John is d feet tall" can be represented as $\mathsf{tall}(\mathsf{john}, d)$ (for simplicity, we omit units such as "feet").

We handle comparatives following the so-called A-not-A analysis (Seuren 1973, Klein 1982) in degree semantics. According to this analysis, (2a) can be represented as (2b), which means that there exists a degree d such that John's weight is more than or equal to d and Bob's weight is not.

(2) a. John is heavier than Bob.
b. $\exists d.\,(\mathsf{heavy}(\mathsf{john}, d) \wedge \neg\mathsf{heavy}(\mathsf{bob}, d))$

Table 1 shows some other examples of basic constructions involving comparatives and their semantic representations.

3 System Overview

Figure 1 shows the overview of ccg-jcomp, our proposed system. The overall system flow follows Haruta et al. (2022): CCG syntactic parsing, tree conversion, semantic parsing, and theorem proving. In what follows, we describe the details of each

Category	Word Type	Semantic Template
NP	common noun	$\lambda E\ N\ F.\exists x.\ (N(E,x) \wedge F(x))$
$S\backslash NP$	positive adjective	$\lambda E\ Q\ N.\ Q(\lambda I.I, \lambda x.N(E, \lambda d.d, \lambda d.d, \lambda t.t, x))$
$S\backslash NP$	negative adjective	$\lambda E\ Q\ N.Q(\lambda I.I, \lambda x.N(E, \lambda d.d, \lambda d.{-}d, \lambda t.\neg t, x))$
$(S/S)\backslash NP$	yori	$\lambda E\ Q\ V.V(\lambda A\ x.Q(\lambda I.I,\ \lambda y.\exists d.(A(x,d) \wedge \neg A(y,d))))$
$(S/S)\backslash NP$	yori (measure phrase)	$\lambda E\ Q\ V.V(\lambda A\ F\ x.Q(\lambda I.I,\ \lambda y.\exists d.(A(x,d) \wedge F(y < d))))$

Table 2: Examples of basic semantic templates

step, deferring the explanation of the specifics of Japanese comparatives until section 4.

3.1 Syntactic Parsing

First, a tokenizer tokenizes the input sentences, and a CCG parser converts them into CCG trees. CCG is a grammar formalism that assigns a syntactic category to each grammatical expression. The set of syntactic categories is defined recursively as follows: (i) atomic categories: NP (noun phrase), S (sentence), etc., (ii) functional categories: X/Y, $X\backslash Y$ (where X and Y are syntactic categories). Both X/Y and $X\backslash Y$ take the category Y as an argument and return the category X. "/" and "\" indicate that the argument is taken from the right and left, respectively.

CCG parsers generally use CCGbank (Hockenmaier and Steedman 2007) or its modified versions for training, which are not necessarily compatible with comparatives. Thus, the output CCG trees are not always the ones we expect at this point. To deal with this issue, we modify the CCG trees if necessary. Another possible way to modify CCG trees is to revise the CCG parser itself. However, this method is costly because it requires re-training or fine-tuning the CCG parser. Thus, we leave this approach for future work.

3.2 Semantic Parsing

In this step, we assign a semantic representation to each lexical item of the CCG tree based on the semantic templates. Then, the semantic representation of the whole sentence is composed according to the CCG rules. To illustrate, we show two rules below. Some other rules are provided in Appendix A.

- Forward functional application rule
$$\frac{X/Y : f \qquad Y : a}{X : f\ a}\ >$$
- Backward functional application rule
$$\frac{Y : a \qquad X\backslash Y : f}{X : f\ a}\ <$$

We set up the semantic templates in order to give semantic representations to the lexical items. Table 2 shows the semantic templates for basic comparative expressions.[3]

Let us proceed to some details of the templates in Table 2, focusing on the function N that appears in the templates for positive/negative adjectives, which we have newly added to handle comparatives.[4] N has five arguments, the first one E being the base form of the adjective, and the fifth one x being the subject of the adjective. Turning to the second argument $\lambda d.d$, it is introduced for the differential comparatives. Consider (3a) for example. In the semantic composition process, this argument becomes $\lambda d.(d + 5)$ as a result of the combination of "5 kg" and the adjective "omoi" (heavy), which leads to the intended semantic representation (3b).

(3) a. Taro-wa Jiro yori 5 kg omoi. (Taro is 5 kg heavier than Jiro.)

b. $\forall d.(\mathsf{heavy}(\mathsf{jiro}, d) \rightarrow \mathsf{heavy}(\mathsf{taro}, d + 5))$

The third argument, $\lambda d.d$ (or $\lambda d.{-}d$), indicates whether the adjective is positive or negative. This allows us to distinguish between (3a) and (4a),

[3] For expository purposes, the semantic templates listed here are simplified from the original ones, which are more complicated in order to handle various expressions.

[4] E (resp. Q) represents the surface form of the word (resp. the generalized quantifier (Barwise and Cooper 1981)).

which contain adjectives of the opposite polarity. For instance, by assuming that "karui" (light) is a negative adjective, we can derive the semantic representation (4b) for (4a), where the argument $\lambda d.-d$ corresponds to -5.

(4) a. Taro-wa Jiro yori 5 kg karui. (Taro is 5 kg lighter than Jiro.)

b. $\forall d.\,(\mathsf{light}(\mathsf{jiro}, d) \rightarrow \mathsf{light}(\mathsf{taro}, d-5))$

Similarly, the fourth argument, $\lambda t.t$ (or $\lambda t.\neg t$), makes a distinction about the polarity of the adjectives in comparatives with measure phrases. Taking (5) and (6) for example, the arguments $\lambda t.t$ and $\lambda t.\neg t$ correspond to $d > 70$ and $\neg(d > 70)$ in the semantic representations, respectively.

(5) a. Taro-wa 70 kg yori omoi. (Taro is heavier than 70 kg.)

b. $\exists d.\,(\mathsf{heavy}(\mathsf{taro}, d) \wedge d > 70)$

(6) a. Taro-wa 70 kg yori karui. (Taro is lighter than 70 kg.)

b. $\exists d.\,(\mathsf{light}(\mathsf{taro}, d) \wedge \neg(d > 70))$

3.3 Theorem Proving

In this step, we input the logical formulas of the premises and hypothesis obtained in the previous step into an automated theorem prover and judge their entailment relation.

Axioms In order to prove entailment relations, we introduce some axioms. To illustrate, we describe one of the axioms, (CP), which is shown below (here, **A** is an adjective). It corresponds to a basic axiom in degree semantics called Consistency Postulate (Klein 1980).

(CP) $\forall x\ y.\,((\exists d.\,(\mathbf{A}(x,d) \wedge \neg\mathbf{A}(y,d)))$
$\rightarrow \forall d.\,(\mathbf{A}(y,d) \rightarrow \mathbf{A}(x,d)))$

Intuitively, this axiom requires that **A** be a predicate such that if the degree of x is greater than the degree of y, then the degree of x is greater than or equal to the degree of y. Using this axiom, we can make inferences such as (1). We give the details of the proof in Appendix B, where we also explain other axioms.

Implementation First, we choose some axioms based on the adjectives in the input sentences and add them as premises. Then, we input the logical formulas of the premises and hypothesis into the automated theorem prover. Given the premises and axioms $P_1, \ldots P_n$ and the hypothesis H, the system output is *yes* (entailment) when $P_1 \wedge \ldots \wedge P_n \rightarrow H$ is proven, *no* (contradiction) when $P_1 \wedge \ldots \wedge P_n \rightarrow \neg H$ is proven, and *unknown* (neutral) when neither is proven.

4 Challenges in Handling Japanese Comparatives

In this section, we explain some linguistic phenomena specific to Japanese comparatives and how we treat them in this study.

4.1 Absence of Overt Comparative Morphemes

English has overt comparative morphemes, such as *more* and *-er*. On the other hand, Japanese has no such morphemes. The examples (7a) and (7b) illustrate that the adjective "omoi" has the same surface form whether it is used for comparison or not.

(7) a. Taro-wa Jiro yori omoi.
Taro-TOP Jiro than heavy
"Taro is heavier than Jiro."

b. Taro-wa omoi.
Taro-TOP heavy
"Taro is heavy."

Although it is possible to give different semantic representations to "omoi" in both sentences, we assign the same semantic representation to simplify the semantic parsing process. Accordingly, we introduce an unpronounced symbol (empty category) to distinguish the semantic representations of the two sentences. Specifically, when there is no comparative expression such as "...yori" and "...izyoo-ni," we insert an empty category *cmp* of category S/S instead. We introduce the aforementioned comparison criterion θ by assigning the following semantic representation (8) to this empty category.

(8) $\lambda S.S(\lambda A\ x.A(x,\theta))$

This inserted operator also plays a role of matching the types of the semantic representations of "Jiro yori omoi" and "*cmp* omoi" (Figures 2 and 3).

4.2 Equatives

English equative sentences such as (9a) are interpreted as indicating "... is at least as heavy as ... ." Thus, Haruta et al. (2022) represented (9a) as (9b).

(9) a. John is as heavy as Bob.

b. $\forall d.\,(\mathsf{heavy}(\mathsf{bob}, d) \rightarrow \mathsf{heavy}(\mathsf{john}, d))$

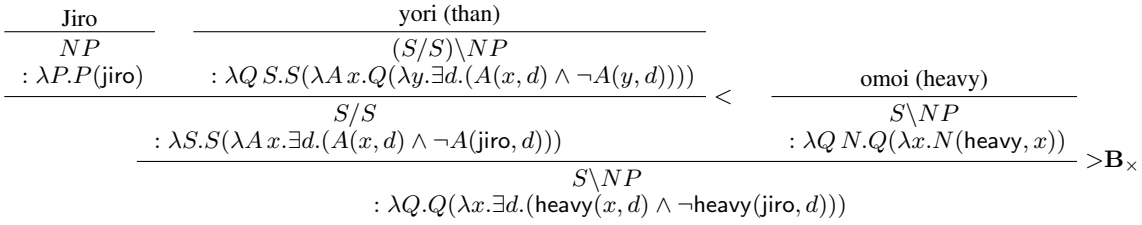


Figure 2: A part of semantic composition of (7a)

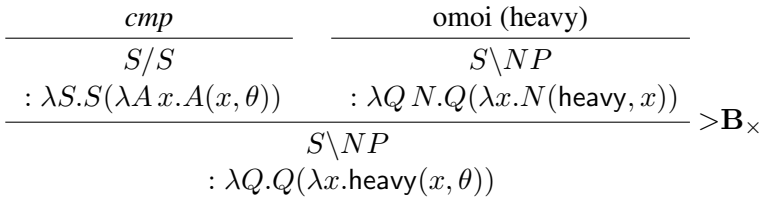


Figure 3: A part of semantic composition of (7b)

On the other hand, Japanese equatives merely express that the degrees are close to each other. For instance, (10a) can be true even when Taro's weight is slightly less than Jiro's.

(10) a. Taro-wa Jiro to onaji kurai-no
Taro-TOP Jiro same as-GEN
omosa-da.
weight-COP
"Taro is as heavy as Jiro."

b. Jiro-wa omoi.
Jiro-TOP heavy
"Jiro is heavy."

c. Taro-wa omoi. (entailment)
Taro-TOP heavy
"Taro is heavy."

To handle the meaning of equatives, we propose the following representation (11) for (10a). This intuitively indicates that the difference in weight between Taro and Jiro is less than the constant δ.

(11) $\forall d_1 d_2.\,((\neg\,(\mathsf{heavy}(\mathsf{taro}, d_1)$
$\leftrightarrow\ \mathsf{heavy}(\mathsf{jiro}, d_1))$
$\wedge\ \neg\,(\mathsf{heavy}(\mathsf{taro}, d_2) \leftrightarrow \mathsf{heavy}(\mathsf{jiro}, d_2)))$
$\rightarrow\ |d_1 - d_2| < \delta)$

We also introduce the following axiom (12), which prescribes the relation between θ and δ. Intuitively, this axiom indicates that δ is so small that the truth value of the predicate heavy does not change within the range of δ from θ.

(12) $\forall x.\,(\mathsf{heavy}(x, \theta - \delta) \leftrightarrow \mathsf{heavy}(x, \theta + \delta))$

We can make inferences such as (10) using this axiom together with (UP) and (DOWN) (see Appendix B for details).

4.3 Clausal Comparatives

Clausal comparatives are comparatives with subordinate clauses. (13a) is an example of a clausal comparative. We also deal with related sentences such as (13b) and (13c).

(13) a. Taro-wa Hanako-ga katta yori
Taro-TOP Hanako-NOM bought than
takai hon-o katta.
expensive book-ACC bought
"Taro bought a more expensive book than Hanako bought."

b. Taro-wa Hanako-ga katta no yori
Taro-TOP Hanako-NOM bought NO than
takai hon-o katta.
expensive book-ACC bought
"Taro bought a more expensive book than what Hanako bought."

c. Taro-wa Hanako yori takai
Taro-TOP Hanako than expensive
hon-o katta.
book-ACC bought
"Taro bought a more expensive book than Hanako."

We assign the same semantic representation (14) to the three sentences in (13).

(14) $\exists d.\,(\exists x.\,(\mathsf{book}(x) \wedge \mathsf{expensive}(x, d)$
$\wedge\,\exists e.\,(\mathsf{bought}(e) \wedge (\mathsf{Nom}(e) = \mathsf{taro})$
$\wedge\ (\mathsf{Acc}(e) = x)))$
$\wedge\,\neg\exists x.\,(\mathsf{book}(x) \wedge \mathsf{expensive}(x, d)$
$\wedge\,\exists e.\,(\mathsf{bought}(e) \wedge (\mathsf{Nom}(e) = \mathsf{hanako})$
$\wedge\ (\mathsf{Acc}(e) = x))))$

Category	Word	Semantic Template
$((NP/NP)/(NP/NP))\backslash(S\backslash NP)$	yori (13a)	$\lambda E\,V\,M.V(\lambda G.\exists d.(M(\lambda A\,x.A(x,d)) \wedge \neg M(\lambda A\,x.(A(x,d) \wedge G(x)))))$
$((NP/NP)/(NP/NP))\backslash NP$	yori (13b)	$\lambda E\,Q\,M.\exists d.(M(\lambda A\,x.A(x,d)) \wedge \neg M(\lambda A\,x.(A(x,d) \wedge Q(\lambda y.(x=y)))))$
$((NP/NP)/(NP/NP))\backslash NP$	yori (13c)	$\lambda E\,Q\,M\,F\,x.Q(\lambda y.$ $(\exists d.M(\lambda A\,z.(A(z,d) \wedge F(x,z)))$ $\wedge \neg M(\lambda A\,z.(A(z,d) \wedge F(y,z)))))$

Table 3: Semantic templates for clausal comparatives

In order to obtain this semantic representation, we assign different semantic representations to "yori" in each sentence, which are listed in Table 3. Note that the template in the second row includes $\lambda y.(x = y)$, which is necessary to consider the fact that the pronominal "no" is identified with "hon" (book) in (13b).

4.4 Presupposition

Some Japanese comparative expressions have a special semantic content called a presupposition (Kubota 2012, Hayashishita 2007). A presupposition is a type of meaning not affected by entailment-canceling operators such as negation and modals (cf. Potts (2015)). The predicate "know" is an example of a presupposition trigger (i.e., an expression or a construction causing presuppositions). In (15a), the presupposition is that Bob ran. This can be confirmed by the fact that the negated sentence (15b) also implies that Bob ran.

(15) a. John knows that Bob ran.
b. John does not know that Bob ran.

We list some Japanese comparative sentences with a presupposition in (16), where the trigger is underlined. Here, the presupposition is that the comparative standard has the property expressed by the predicate. That is, the three sentences in (16) all presuppose that Jiro is heavy.

(16) a. Taro-wa Jiro izyoo-ni omoi.
Taro-TOP Jiro than heavy
"Taro is heavier than Jiro."

b. Taro-wa Jiro to onaji kurai omoi.
Taro-TOP Jiro as same as heavy
"Taro is as heavy as Jiro."

c. Taro-wa Jiro hodo omoku nai.
Taro-TOP Jiro hodo heavy not
"Taro is not as heavy as Jiro."

In formally analyzing presuppositions, it is not adequate to simply conjoin the presupposition with other parts of the sentence. For example, suppose we represent the meaning of (15a) as a conjunction of the semantic representations of "John knows that Bob ran" and "Bob ran," as shown below.

(17) $\mathsf{know}(\mathsf{john}, \mathsf{ran}(\mathsf{bob})) \wedge \mathsf{ran}(\mathsf{bob})$

The negation of this formula, which is shown in (18), does not entail $\mathsf{ran}(\mathsf{bob})$, failing to capture the fact that the presupposition is not subject to the negation (cf. (15b)).

(18) $\neg\,(\mathsf{know}(\mathsf{john}, \mathsf{ran}(\mathsf{bob})) \wedge \mathsf{ran}(\mathsf{bob}))$
$\Leftrightarrow \neg\mathsf{know}(\mathsf{john}, \mathsf{ran}(\mathsf{bob})) \vee \neg\mathsf{ran}(\mathsf{bob})$

In order to correctly handle presuppositions, we use a framework called *multidimensional semantics* (Karttunen and Peters 1979). In this framework, the semantic representation of an entire sentence is represented by a pair of semantic representations. The first element is for the central content conveyed by the sentence (the *at-issue* content), and the second one is for the presupposition. For example, the semantic representation of the sentence (16a) is shown in (19).

(19) $\langle \exists d.\,(\mathsf{heavy}(\mathsf{taro}, d) \wedge \neg\mathsf{heavy}(\mathsf{jiro}, d))\,,$
$\mathsf{heavy}(\mathsf{jiro}, \theta)\rangle$

When the sentence is negated, we only negate the semantic representation of the at-issue content in the semantic composition, and the semantic representation of the entire sentence is (20).

(20) $\langle \neg\exists d.\,(\mathsf{heavy}(\mathsf{taro}, d) \wedge \neg\mathsf{heavy}(\mathsf{jiro}, d))\,,$
$\mathsf{heavy}(\mathsf{jiro}, \theta)\rangle$

In the theorem proving step, we conjoin the semantic representations for the at-issue content and for the presupposition with $\wedge$.

5 Experiment

5.1 Settings

In this section, we describe the implementation settings of the proposed system.

Syntactic Parsing We use a Japanese tokenizer Janome.[5] As a CCG parser, we use depccg (Yoshikawa et al. 2017), the best-performing model provided for Japanese. We use Tsurgeon (Levy and Andrew 2006) to modify CCG parsing trees and insert empty categories. Our modification processes are as follows:

- We add rules to merge some multiword expressions. For instance, "izyoo ni" is converted to "izyoo-ni," "yori mo" to "yori-mo," and "to onaji kurai no" to "to-onaji-kurai-no."
- We insert the empty category *cmp* (cf. Section 4.1).
- We add a new syntactic feature to "yori" in phrasal comparatives related to clausal comparatives[6] in order to distinguish it from "yori" in ordinary phrasal comparatives.
- We add a new syntactic feature to "yori" in comparatives with a measure phrase in order to distinguish it from "yori" in ordinary phrasal comparatives.

In total, we make 60 entries in the Tsurgeon script for these processes.

Semantic Parsing For semantic composition, we use ccg2lambda (Martínez-Gómez et al. 2016), which supports Japanese as well as English. It uses λ-calculus to derive semantic representations. We extend the semantic templates to introduce the semantic representations based on degree semantics. We create two templates, one with multidimensional semantics and one without. The total number of lexical entries in each semantic template file is 222. We newly add 58 entries for words related to comparatives.

Theorem Proving We use Vampire 4.9 (Kovács and Voronkov 2013), a resolution-based automated theorem prover, for theorem proving. Vampire uses the Thousand of Problems for Theorem Provers (TPTP, Sutcliffe 2017) format to describe logical formulas. For this reason, we convert the output of ccg2lambda into first-order predicate logic formulas in the TPTP format. At this point, we add the axioms described in Section 3.3. In this step, we use the CASC mode, the fastest mode in Vampire. We try to prove $P_1 \wedge P_2 \wedge \ldots \wedge P_n \rightarrow H$ and $P_1 \wedge P_2 \wedge \ldots \wedge P_n \rightarrow \neg H$ for up to 20 seconds each to determine the system output.

[5] https://github.com/mocobeta/janome

[6] For example, "Taro-wa Hanako yori takai hon-o katta. (Taro bought a more expensive book than Hanako.)"

5.2 Dataset

We use the comparatives section of the JSeM dataset (Kawazoe et al. 2017) for evaluation of our inference system. This NLI dataset contains Japanese counterparts of the FraCaS test suite (Cooper et al. 1996). It also contains newly added problems that involve phenomena FraCaS does not address or phenomena unique to Japanese.

In this study, we do not address tense and aspect, so we eliminated problems involving them. We do not address modality as well. With regard to modality, JSeM only has problems involving the property that modals do not affect the presupposition. Thus, we replaced modals with negation on these problems. As a result, the number of problems in the dataset is 71. The distribution of the gold answer labels is (*yes*/*no*/*unknown*) = (42/8/21). Table 4 shows some problems in the dataset.

jsem-569, Gold answer: yes	
P1	PC-6082-wa ITEL-XZ yori hayai. (PC-6082 is faster than ITEL-XZ.)
P2	ITEL-XZ-wa hayai. (ITEl-XZ is fast.)
H	PC-6082-wa hayai. (PC-6082 is fast.)
jsem-576, Gold answer: no	
P1	PC-6082-wa ITEL-XZ to onaji kurai-no hayasa-da. (PC-6082 is as fast as ITEL-XZ.)
P2	PC-6082-wa osoi. (PC-6082 is slow.)
H	ITEL-XZ-wa hayai. (ITEL-XZ is fast.)

Table 4: Examples of the problems in JSeM. P and H stand for "premise" and "hypothesis," respectively.

5.3 Evaluation Method

We use accuracy as an evaluation metric. When an error occurs in the proposed system, we treat it as an incorrect answer. As a baseline, we adopt

GPT-4o and Swallow 8B[7]/70B[8] (S-8B/S-70B), the latter being competitive open Japanese LLMs. We conduct experiments using six different prompts for these models and calculate the accuracy as the average across these prompts.[9] The details of the prompts are shown in Appendix D.

6 Results and Discussion

6.1 Results

Table 5 shows the accuracy on the JSeM dataset. The table shows that our system outperformed all baseline models in terms of accuracy. The detailed results are shown in Appendix E.

Majority	GPT-4o	S-8B	S-70B	Ours
.592	.774	.549	.712	**.845**

Table 5: Accuracy on the JSeM dataset. "Majority" indicates the accuracy achieved when "yes," the most common label in the dataset, is answered for all the problems.

jsem-570, Gold answer: unknown GPT-4o: yes, Ours: unknown	
P	PC-6082-wa ITEL-XZ yori hayai. (PC-6082 is faster than ITEL-XZ.)
H	PC-6082-wa hayai. (PC-6082 is fast.)
jsem-620, Gold answer: yes GPT-4o: unknown, Ours: yes	
P	Taro-wa Hanako izyoo-ni hayaoki-da. (Taro is an earlier riser than Hanako.)
H	Hanako-wa hayaoki-da. (Hanako is an early riser.)

Table 6: Examples of problems that GPT-4o did not answer correctly but ours did

Table 6 shows some examples of problems that our system could predict correct answers while GPT-4o could not. GPT-4o incorrectly answered some of the relatively simple problems, such as jsem-570. The possible reason is that GPT-4o inferred "X is fast" from "X is faster."
Notably, GPT-4o failed to answer correctly some problems with presupposition triggers, such as jsem-620. In order to perform this inference, it is necessary to infer the presupposition that Hanako is an early riser from the premise. GPT-4o was rarely able to solve such problems. On the other hand, our proposed system correctly predicted the entailment relation, thanks to multidimensional semantics.

6.2 Error Analysis

Table 7 shows two cases where our system failed to obtain correct semantic representations, but GPT-4o gave correct answers. In jsem-589, we can interpret "APCOM-no keiyaku" either as the contracts that APCOM won or as the contracts that ITEL won from APCOM. To handle this kind of ambiguity, we need to (i) add a new semantic representation of "yori," and (ii) implement a system for distinguishing between the two interpretations based on syntactic information.

Jsem-606 is another case where our system failed to make a correct prediction. The verb "magaru" (bend) behaves like an adjective when combined with "te-i-ru." However, our system treats the resultant predicate "magatte-i-ru" as a verb, so its semantic type does not match the one required for the argument of "yori," causing an error in semantic parsing. To handle this error, we need to give an exceptional semantic representation to "te-i-ru" when it forms an adjective-like predicate with certain verbs like "magaru."

jsem-589, Gold answer: yes GPT-4o: yes, Ours: error	
P	ITEL-wa APCOM-no keiyaku yori ooku-no chuumon-o kakutoku-sita. (ITEL won more orders than the APCOM contract.)
H	ITEL-wa APCOM-no chuumon-o kakutoku-shita. (ITEL won the APCOM contract.)
jsem-606, Gold answer: yes GPT-4o: yes, Ours: error	
P	Kono boo-wa ano boo yori magatte-i-ru. (This stick is more bent than that one.)
H	Kono boo-wa magatte-i-ru. (This stick is bent.)

Table 7: Examples of problems our system answered incorrectly

7 Conclusion

In this study, we have proposed ccg-jcomp, a logical inference system for Japanese comparatives

[7] tokyotech-llm/Llama-3.1-Swallow-8B-v0.1
[8] tokyotech-llm/Llama-3.1-Swallow-70B-v0.1
[9] The model inferences were conducted in May 2025.

based on CCG, degree semantics, and some analyses of phenomena unique to Japanese comparatives. In our experiments with the Japanese NLI dataset that involves comparatives, we demonstrated that our proposed system achieved higher accuracy than several LLMs.

In future work, we are considering handling the ambiguity of certain sentences and the behavior of the adjective-like verbs discussed in Section 6.2. Additionally, it would be desirable to address adverbial comparatives, which are not covered in JSeM.

Limitations

Few-shot Learning In this study, we did not compare methods using few-shot learning as a baseline. It may improve the performance of the baseline models. For example, the LLMs may correctly answer jsem-620 in Section 6.1 by looking at some example inferences with a presupposition and learning the inference patterns. However, we do not have a sufficient number of problems involving Japanese comparatives to carry out and evaluate few-shot learning. Therefore, we conducted all experiments in a zero-shot setting for all models.

Scalability In addition to comparatives, JSeM has sections on other linguistic phenomena, such as anaphora. However, since our proposed system focuses only on Japanese comparatives, it cannot be used as is to handle these phenomena. To address them, we need to introduce the mechanism employed by some specific frameworks (e.g., *dynamic semantics* (Groenendijk and Stokhof 1991) for anaphora) in a manner consistent with degree semantics, which is not trivial. Hence, we leave for future work the development of a unified system that can handle these phenomena together with comparatives.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments, which helped us improve this paper. This work was supported by the Institute for AI and Beyond of the University of Tokyo and JSPS KAKENHI Grant Number JP24H00809, Japan.

References

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

Jon Barwise and Robin Cooper. 1981. Generalized quantifiers and natural language. In *Philosophy, language, and artificial intelligence: Resources for processing natural language*, pages 241–301. Springer.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2017. A type-theoretical system for the FraCaS test suite: Grammatical framework meets coq. In *Proceedings of the 12th International Conference on Computational Semantics (IWCS) — Long papers*.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2021. Applied temporal analysis: A complete run of the FraCaS test suite. In *Proceedings of the 14th International Conference on Computational Semantics (IWCS)*, pages 11–20, Groningen, The Netherlands (online). Association for Computational Linguistics.

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.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Max J Cresswell. 1976. The semantics of degree. In *Montague grammar*, pages 261–292. Elsevier.

Jeroen Groenendijk and Martin Stokhof. 1991. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100.

Izumi Haruta, Koji Mineshima, and Daisuke Bekki. 2022. Implementing natural language inference for comparatives. *Journal of Language Modelling*, 10(1):139–191.

J-R Hayashishita. 2007. Izyoo (ni)-and gurai-comparatives: Comparisons of deviation in japanese. *GENGO KENKYU (Journal of the Linguistic Society of Japan)*, 132:77–109.

Julia Hockenmaier and Mark Steedman. 2007. Ccgbank: a corpus of ccg derivations and dependency structures extracted from the penn treebank. *Computational Linguistics*, 33(3):355–396.

Hai Hu, Qi Chen, Kyle Richardson, Atreyee Mukherjee, Lawrence S. Moss, and Sandra Kuebler. 2020. MonaLog: a lightweight system for natural language inference based on monotonicity. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 334–344, New York, New York. Association for Computational Linguistics.

Lauri Karttunen and Stanley Peters. 1979. Conventional Implicature. In *Presupposition*, pages 1–56. Brill.

Ai Kawazoe, Ribeka Tanaka, Koji Mineshima, and Daisuke Bekki. 2017. An inference problem set for evaluating semantic theories and semantic processing systems for japanese. In *New Frontiers in Artificial Intelligence*, pages 58–65, Cham. Springer International Publishing.

Ewan Klein. 1980. A semantics for positive and comparative adjectives. *Linguistics and Philosophy*, 4:1–45.

Ewan Klein. 1982. The interpretation of adjectival comparatives1. *Journal of Linguistics*, 18(1):113–136.

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

Yusuke Kubota. 2012. The presuppositional nature of izyoo (-ni) and gurai comparatives: A note on hayashishita (2007). *GENGO KENKYU (Journal of the Linguistic Society of Japan)*, 141:33–47.

Roger Levy and Galen Andrew. 2006. Tregex and Tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of ChatGPT and GPT-4. *Preprint*, arXiv:2304.03439.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL-2016 System Demonstrations*, pages 85–90.

Koji Mineshima, Pascual Martínez-Gómez, Yusuke Miyao, and Daisuke Bekki. 2015. Higher-order logical inference with compositional semantics. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2055–2061.

Mihir Parmar, Nisarg Patel, Neeraj Varshney, Mutsumi Nakamura, Man Luo, Santosh Mashetty, Arindam Mitra, and Chitta Baral. 2024. LogicBench: Towards systematic evaluation of logical reasoning ability of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13679–13707, Bangkok, Thailand. Association for Computational Linguistics.

Christopher Potts. 2015. Presupposition and implicature. *The Handbook of Contemporary Semantic Theory*, pages 168–202.

Pieter A. M. Seuren. 1973. *The Comparative*, pages 528–564. Springer Netherlands, Dordrecht.

Jingyuan S. She, Christopher Potts, Samuel R. Bowman, and Atticus Geiger. 2023. ScoNe: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1803–1821, Toronto, Canada. Association for Computational Linguistics.

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

Geoff Sutcliffe. 2017. The TPTP problem library and associated infrastructure: from CNF to TH0, TPTP v6. 4.0. *Journal of Automated Reasoning*, 59(4):483–502.

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

A Combinatory Rules of CCG

We show some combinatory rules of CCG below (see Steedman (2000) for details).

- Forward functional application rule
$$\frac{X/Y : f \qquad Y : a}{X : f\,a} >$$
- Backward functional application rule
$$\frac{Y : a \qquad X\backslash Y : f}{X : f\,a} <$$
- Forward functional composition rule
$$\frac{X/Y : f \qquad Y/Z : g}{X/Z : \lambda x.f(g\,x)} >\mathbf{B}$$
- Backward functional composition rule
$$\frac{Y\backslash Z : g \qquad X\backslash Y : f}{X\backslash Z : \lambda x.f(g\,x)} <\mathbf{B}$$
- Forward functional crossed composition rule
$$\frac{X/Y : f \qquad Y\backslash Z : g}{X\backslash Z : \lambda x.f(g\,x)} >\mathbf{B}_{\times}$$
- Backward functional crossed composition rule
$$\frac{Y/Z : g \qquad X\backslash Y : f}{X/Z : \lambda x.f(g\,x)} >\mathbf{B}_{\times}$$

B Details of Axioms

Table 8 shows the axioms employed in our system. (CP) is the axiom we already introduced in Section 3.3. We can make the following inferences using this axiom. (21a) and (21b) are the premises, and (21c) is the hypothesis.

Name	Logical Formula
CP	$\forall x\ y.\ ((\exists d.\ (\mathbf{A}(x,d) \wedge \neg\mathbf{A}(y,d))) \rightarrow \forall d.\ (\mathbf{A}(y,d) \rightarrow \mathbf{A}(x,d)))$
ANT	$\forall x\ d.\ (\mathbf{P}(x,d) \leftrightarrow \neg\mathbf{N}(x,d))$
UP	$\forall x\ d.\ (\mathbf{P}(x,d) \rightarrow \forall d'.\ (d' \leq d \rightarrow \mathbf{P}(x,d')))$
DOWN	$\forall x\ d.\ (\mathbf{N}(x,d) \rightarrow \forall d'.\ (d' \geq d \rightarrow \mathbf{N}(x,d')))$
DELTA	$\forall x.\ (\mathbf{A}(x,\theta-\delta) \leftrightarrow \mathbf{A}(x,\theta+\delta))$

Table 8: Axioms for Japanese comparatives. **A** denotes adjectives, **P** denotes positive adjectives such as *long*, and **N** denotes negative adjectives such as *short*.

(21) a. Taro-wa Jiro yori omoi.
Taro-TOP Jiro than heavy
"Taro is heavier than Jiro."

b. Jiro-wa 70 kg yori omoi.
Jiro-TOP 70 kg than heavy
"Jiro is heavier than 70 kg."

c. Taro-wa 70 kg yori omoi.
Taro-TOP 70 kg than heavy
"Taro is heavier than 70 kg."

(entailment)

Concretely, from (CP) and (21a), we obtain $\mathsf{heavy}(\mathsf{jiro}, 70) \rightarrow \mathsf{heavy}(\mathsf{taro}, 70)$. Then, from this formula and (21b), we can derive $\mathsf{heavy}(\mathsf{taro}, 70)$.

(ANT) indicates the antonymy relation between positive and negative adjectives. The following is an example of an inference using this axiom. (22a) is the premise and (22b) is the hypothesis.

(22) a. Taro-wa Jiro yori omoi.
Taro-TOP Jiro than heavy
"Taro is heavier than Jiro."

b. Taro-wa Jiro yori karui.
Taro-TOP Jiro than light
"Taro is lighter than Jiro."

(contradiction)

(UP) and (DOWN) are axioms that indicate the monotonicity of positive and negative adjectives, respectively. (DELTA) is an axiom about equatives. Using these axioms, we can prove the entailment relation in (10) as follows. (23a), (23b), and (23c) are the semantic representations of (10a), (10b), and (10c), respectively.

(23) a. $\forall d_1\ d_2.\ ((\neg\ (\mathsf{heavy}(\mathsf{taro}, d_1)$
$\leftrightarrow\ \mathsf{heavy}(\mathsf{jiro}, d_1))$
$\wedge \neg\ (\mathsf{heavy}(\mathsf{taro}, d_2)$
$\leftrightarrow\ \mathsf{heavy}(\mathsf{jiro}, d_2)))$
$\rightarrow\ |d_1 - d_2| < \delta)$

b. $\mathsf{heavy}(\mathsf{jiro}, \theta)$

c. $\mathsf{heavy}(\mathsf{taro}, \theta)$

First, from (23b), (UP), and (DELTA), we can derive $\mathsf{heavy}(\mathsf{jiro}, \theta)$ and $\mathsf{heavy}(\mathsf{jiro}, \theta+\delta)$. Then, by replacing d_1 (resp. d_2) in (27a) with $\theta+\delta$ (resp. δ), and by contraposition, we obtain either $\mathsf{heavy}(\mathsf{taro}, \theta+\delta) \leftrightarrow \mathsf{heavy}(\mathsf{jiro}, \theta+\delta)$ or $\mathsf{heavy}(\mathsf{taro}, \theta) \leftrightarrow \mathsf{heavy}(\mathsf{jiro}, \theta)$. In both cases, $\mathsf{heavy}(\mathsf{taro}, \theta)$ is true since we have $\mathsf{heavy}(\mathsf{jiro}, \theta+\delta)$ and $\mathsf{heavy}(\mathsf{jiro}, \theta)$.

In the implementation, (CP) and (DELTA) are added for all gradable adjectives. (ANT) is added for adjectives that have an antonym. (UP) and (DOWN) are added for positive adjectives and negative adjectives, respectively.

C Problem Replacement

Table 9 shows an example of the problems in JSeM to which we applied the replacement we discussed in section 5.2. The original problem uses the property that the presupposition "Hanako is an early riser" is not affected by the modal "kamo-sire-nai." We did not implement the semantic representation of modals, so we replaced them with a negation "to-iu-wake-de-wa-nai." Since presuppositions are unaffected by negation (as well as by modals), this replacement does not alter the purpose of the problem—namely, to test whether the model understands that presuppositions are not influenced by entailment-canceling operators.

D Prompts for the Baseline Models

Table 10 and Table 11 show examples of prompts for GPT-4o and Swallow, respectively.

jsem-621 (**original**), Gold answer: yes	
Premise	Taro-wa Hanako izyoo-ni hayaoki kamo-sire-nai. (Taro may be an earlier riser than Hanako.)
Hypothesis	Hanako-wa Hayaoki-da. (Hanako is an early riser.)
jsem-621 (**replaced**), Gold answer: yes	
Premise	Taro-wa Hanako izyoo-ni hayaoki toiu-wake-de-wa-nai. (Taro is not an earlier riser than Hanako.)
Hypothesis	Hanako-wa Hayaoki-da. (Hanako is an early riser.)

Table 9: An example of the problems in which we replaced a modal with a negation

system	前提文と仮説文が与えられます。 前提文が仮説文を含意しているか答えてください。 「含意」、「矛盾」、「中立」のいずれかで答えてください。 (You are given premises and a hypothesis. Answer whether the premises entail the hypothesis. Answer with “entailment”, “contradiction”, or “neutral.”)
user	前提１：PC-6082はITEL-XZより速い。 前提２：ITEL-XZは速い。 仮説：PC-6082は速い。 (Premise 1: PC-6082 is faster than ITEL-XZ. Premise 2: ITEL-XZ is fast. Hypothesis: PC-6082 is fast.)

Table 10: Example of the prompt for GPT-4o

前提文と仮説文が与えられます。
前提文が仮説文を含意しているか答えてください。
「含意」、「矛盾」、「中立」のいずれかで答えてください。
前提１：PC-6082はITEL-XZより速い。
前提２：ITEL-XZは速い。
仮説：PC-6082は速い。
回答：

Table 11: Example of the prompt for Swallow

E Detailed Results

Table 12, Table 13, and Table 14 show the detailed evaluation results of each baseline model. E, C, and N represent entailment, contradiction, and neutral, respectively. "Prompt Type" indicates the order of the words 含意 (entailment), 矛盾 (contradiction), and 中立 (neutral) as they appear in each prompt. For example, Table 10 and Table 11 show prompts of the E-C-N (含意-矛盾-中立) type.

Swallow 8B tended to output *yes* when 含意 or 中立 appeared first in the prompt, resulting in substantially lower F1 scores for contradiction and neutral compared to entailment. Conversely, when 矛盾 was presented first, the number of *no* responses increased.

In contrast, Swallow 70B and GPT-4o produced more balanced outputs, achieving higher F1 scores than Swallow 8B.

Prompt Type	Accuracy	Gold Label	Precision	Recall	F1 Score
E-C-N	0.619	E	0.62	1.00	0.76
		C	0.00	0.00	0.00
		N	0.67	0.10	0.17
E-N-C	0.605	E	0.61	1.00	0.76
		C	0.50	0.12	0.20
		N	0.00	0.00	0.00
C-E-N	0.225	E	0.80	0.19	0.31
		C	0.13	1.00	0.23
		N	0.00	0.00	0.00
C-N-E	0.591	E	0.77	0.81	0.79
		C	0.30	1.00	0.46
		N	0.00	0.00	0.00
N-E-C	0.605	E	0.60	1.00	0.75
		C	1.00	0.12	0.22
		N	0.00	0.00	0.00
N-C-E	0.647	E	0.64	1.00	0.78
		C	0.75	0.38	0.50
		N	1.00	0.05	0.09

Table 12: Evaluation results of Swallow 8B on each prompt

Prompt Type	Accuracy	Gold Label	Precision	Recall	F1 Score
E-C-N	0.647	E	0.80	0.86	0.83
		C	0.36	1.00	0.53
		N	0.50	0.10	0.16
E-N-C	0.690	E	0.81	0.83	0.82
		C	0.55	0.75	0.63
		N	0.47	0.38	0.42
C-E-N	0.676	E	0.80	0.86	0.83
		C	0.40	1.00	0.57
		N	0.67	0.19	0.30
C-N-E	0.661	E	0.80	0.86	0.83
		C	0.42	1.00	0.59
		N	0.43	0.14	0.21
N-E-C	0.760	E	0.88	0.83	0.85
		C	0.62	1.00	0.76
		N	0.61	0.52	0.56
N-C-E	0.788	E	0.88	0.86	0.87
		C	0.62	1.00	0.76
		N	0.71	0.57	0.63

Table 13: Evaluation results of Swallow 70B on each prompt

Prompt Type	Accuracy	Gold Label	Precision	Recall	F1 Score
E-C-N	0.746	E	0.83	0.81	0.82
		C	0.75	0.75	0.75
		N	0.59	0.62	0.60
E-N-C	0.774	E	0.84	0.86	0.85
		C	0.75	0.75	0.75
		N	0.65	0.62	0.63
C-E-N	0.774	E	0.85	0.83	0.84
		C	0.78	0.88	0.82
		N	0.62	0.62	0.62
C-N-E	0.760	E	0.85	0.81	0.83
		C	0.78	0.88	0.82
		N	0.59	0.62	0.60
N-E-C	0.788	E	0.86	0.86	0.86
		C	0.75	0.75	0.75
		N	0.67	0.67	0.67
N-C-E	0.788	E	0.86	0.86	0.86
		C	0.78	0.88	0.82
		N	0.65	0.62	0.63

Table 14: Evaluation results of GPT-4o on each prompt

In the Mood for Inference: Logic-Based Natural Language Inference with Large Language Models

Bill Noble[†] and **Rasmus Blanck**[†] and **Gijs Wijnholds**[‡]
[†]Dept. of Philosophy, Linguistics and Theory of Science, University of Gothenburg
[‡]Leiden Institute of Advanced Computer Science, Leiden University
bill.noble@gu.se, rasmus.blanck@gu.se, g.j.wijnholds@liacs.leidenuniv.nl

Abstract

In this paper we explore a hybrid approach to challenging Natural Language Inference datasets that combines Large Language Models (LLMs) and logical theorem proving. We report on an experiment which combines an LLM meta-prompting strategy, eliciting logical representations, and Prover9, a first-order logic theorem prover. In addition, we experiment with the inclusion of (logical) world knowledge. Our findings suggest that (i) requesting first-order logic formalizations of sentences usually improves model performance, even when those formulas are not explicitly used, (ii) determining the inference relation from the generated formulas nevertheless performs worse, and (iii) priming the model to generate relative world knowledge is sometimes effective. We argue that these results explicate the weaknesses of both approaches. As such, we consider this study a source of inspiration for future work in the field of neuro-symbolic reasoning.

1 Introduction

Natural Language Inference (NLI) is a core task in Natural Language Processing (NLP) and is often presented as a proxy measure of the reasoning capabilities of NLP models. Briefly, a model is presented with a premise sentence and a hypothesis sentence and must decide whether the hypothesis is entailed by the premise (E), contradicts it (C), or is neutral with respect to it (N).

Although many NLI datasets have been developed, starting with the SICK dataset of Marelli et al. (2014) and the SNLI dataset of Bowman et al. (2015), more attention has recently been put on using NLI to measure specific linguistic phenomena. The MED dataset, for example, tests for monotonic reasoning (Yanaka et al., 2019a; Richardson et al., 2020). The CURRICULUM benchmark (Chen and Gao, 2022) is a notable aggregation of NLU tasks (including things like question answering) that have been uniformly formulated as NLI tasks.[1]

One often heard criticism of NLI as a task is that datasets often contain biases and annotation artifacts, and that models trained on them exhibit poor generalization capabilities. It is shown by Yanaka et al. (2019a) for English, and corroborated by Wijnholds (2023) for Dutch, that models have a tendency to overtune, and that they fail to properly address negation. That models seem to exploit relatively shallow heuristics such as lexical overlap and sentence length, is confirmed in prior work (Naik et al., 2018; McCoy et al., 2019), and an effort to repair an existing dataset is done by Kalouli et al. (2023). Finally, NLI models don't necessarily transfer gracefully to other NLI datasets (Talman and Chatzikyriakidis, 2019; Bhargava et al., 2021).

Many of the mentioned datasets and results were achieved with encoder-only models like BERT, which can narrowly generalize through finetuning; gradually, this has been replaced by decoder-only Large Language Models (GPT-3 onward), allowing for the NLI task to be stated as a text-to-text problem. Though this avoids some of the above-mentioned pitfalls, the results of McKenna et al. (2023) show that generative language models still suffer from bias and additionally are a source of *hallucinations*, an issue that is persistent for models that are effectively next-word predictors.

In order to control the output of model prompting, one method is to specifically *constrain* model output as a part of the decoding scheme; additionally this has the benefit of guaranteeing syntactic correctness over prompting results, leading to more

[1]A historical note: Prior to the advent of neural (language) models, Recognizing Textual Entailment (RTE) was the more common terminology for inference datasets. These datasets, such as the FraCaS suite (Cooper et al., 1996), typically framed the task as a two-way classification (entailment vs. non-entailment) with canonical examples for different linguistic phenomena, but there is no hard distinction between NLI and RTE. In the continuation, we use the NLI/RTE terminology primarily to distinguish between three- and two-label datasets.

Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA), pages 33–47

effective prompting strategies. Constrained decoding approaches can work either through vocabulary filters (e.g. only 'E', 'N' and 'C' are valid prompt continuations), or through more sophisticated strategies like generating vocabulary filters determined by finite state automata (aka regular expressions) (Willard and Louf, 2023) or context-free grammars (Beurer-Kellner et al., 2024). While these approaches provide some control over model output, they are nevertheless limited to *syntactic* correctness, meaning they will not fully avoid hallucinations. In this work we use vocabulary filters.

Mixing LLM prompting with logic-based approaches is an emerging field with a number of precedents in NLP. A recent example is the study of Pan et al. (2023), which combines LLM prompting with theorem proving for logical reasoning, with the downside of returning incorrect representations back to the LLM to repeat the prompting procedure. Another work suggests constrained decoding for a variety of (structured) NLP tasks (Geng et al., 2023), but unfortunately doesn't provide a concrete implementation for most examples.

While there is some recent work attempting to bring logical representations in the loop in order to formalize the (chain-of-thought) prompting process (Ranaldi et al., 2025), logic-based approaches to NLI are rare, and were mostly performed in the era before LLMs, typically in a multimodal setting or following a pipeline where sentences are first encoded using a syntactic and semantic parser, after which a classification is made (Abzianidze, 2020; Abzianidze and Kogkalidis, 2021; Chen et al., 2021; Suzuki et al., 2019; Tomihari and Yanaka, 2023).

In this work we set out to provide a pilot study mixing the above approaches to tackle complex NLI test sets with a variety of strategies, including prompting LLMs for first-order logical representations. [2]

2 Logic-Based NLI with an LLM

Concretely, our pipeline works as follows: We prompt a model to generate logical representations for a given premise–hypothesis pair, after which we re-prompt the model to generate a (constrained) answer on the (non-)entailment between the premise and hypothesis. Whenever relevant, we feed the generated formulas to a theorem prover[3] to assess which path is more performant. As a baseline we consider a *label only* strategy where the model is not prompted for any logical representations but must directly generate the NLI label.

Datasets We evaluate our approach on six different sections of the CURRICULUM benchmark (Chen and Gao, 2022). The **comparative**, **conditional**, **negation**, and **quantifiers** datasets are drawn from Richardson et al. (2020) and follow the NLI format. The **lexical entailment** section has test set items drawn from Schmitt and Schütze (2021) and Glockner et al. (2018), and the **monotonicity** section has test set items drawn from Yanaka et al. (2019b) and Richardson et al. (2020). These later two sections use the RTE format.

Prompt setup Our prompting approach is an example of meta-prompting, i.e., the outcome of the first prompt is included in a second prompt. We distinguish three alternatives: Firstly, the *label only* prompt asks the model to directly generate an NLI label, constrained to the three possibilities (E, N, C). Second, the *formula* prompt asks the model to first generate formulas in first-order logic (in a model-friendly format) for the premise and hypothesis and then is asked to generate a label, hence incorporating both textual and logical representations of the NLI instance. Finally, there is the *formulas and world knowledge* setting where the intermediate generation prompt provides formulas and relevant world knowledge in logical form.

Model choice Given that we strive for full transparency in our experiments, we set four desiderata (in order of importance) and choose such that the model (1) has freely available architecture code; (2) has fully specified training data; (3) has a reasonable performance baseline; and (4) is as small as possible modulo the preceding points. Given these constraints, we work with Zephyr[4], which strikes a balance between performance and model size, and is fully transparent in terms of architecture code and training data.

3 Results

Table 1 displays the overall results for several NLI datasets. Additionally, results on the two RTE tasks are given in Table 2.

[2] The code is available online: https://github.com/GU-CLASP/logic-based-NLI-with-LLMs/

[3] Prover9, (McCune, 2005–2010)

[4] https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

	Label	**E**		**C**		**N**	
		DA	P9	DA	P9	DA	P9
Dataset	*Prompt*						
comparative	label-only	77.6	–	73.7	–	6.8	–
	forms	71.4	8.2	**94.9**	0.0	**17.5**	99.0
	forms+wk	**98.0**	100.0	56.6	0.0	9.7	0.0
conditional	label-only	48.4	–	0.0	–	50.9	–
	forms	**74.7**	59.8	**54.7**	50.0	**43.6**	100.0
	forms+wk	50.5	37.0	51.6	48.4	0.9	100.0
negation	label-only	0.0	–	15.6	–	36.5	–
	forms	0.0	0.0	60.0	98.9	**90.4**	100.0
	forms+wk	**2.2**	0.0	**62.2**	98.8	9.6	98.2
quantifier	label-only	70.0	–	3.5	–	**96.9**	–
	forms	72.2	59.8	**82.5**	27.2	29.2	100.0
	forms+wk	**98.9**	64.0	1.8	23.5	5.2	100.0

Table 1: Mean accuracy (recall) by label for Zephyr on NLI datasets, using different prompt schemes. Where relevant, accuracy is shown for both the LLM's direct answer (**DA**) and the label inferred from the generated formulas and the Prover9 theorem prover (**P9**). For each dataset and label, the best DA result is **bolded** and the P9 result in underlined when it exceeds the DA result. The P9 column excludes items that resulted in a Prover9 error (see Table 3 for the unfiltered results).

	Label	**E**		**N/C**	
		DA	P9	DA	P9
Dataset	*Prompt*				
lexical	label-only	6.7	–	94.6	–
	forms	**25.0**	1.5	**97.3**	100.0
	forms+wk	**25.0**	65.3	96.6	34.8
monotonicity	label-only	**58.3**	–	34.8	–
	forms	46.8	16.7	**47.8**	90.9
	forms+wk	47.5	50.0	47.2	54.6

Table 2: As Table 1 but for the RTE datasets. See Table 4 for the unfiltered Prover9 results.

In the mood for logic We firstly note that the *direct answer* performance is generally higher in the setup in which the model is asked to generate a logic formula (*formulas*) and subsequentially generate the NLI label. In other words: When the model is *self-primed* to think in terms of logic, it appears to label in terms of logic, generally improving performance. A notable exception to this rule is the case of neutral-labeled items in the **quantifier** dataset, where the *label-only* setup performed significantly better (96.9%) than either of the two prompt schemes involving formulas (29.9% for *formulas* and 5.2 for *formulas + world knowledge*). For this dataset, it seems that generating formulas causes the model to predict a logical relationship between sentences where in fact none exists.

Priming the model to generate formulas for relevant world knowledge helps in some cases, but the effect is much more mixed. For example, it is beneficial for entailments in the **comparative** and **quantifier** datasets, but detrimental for the other two labels in the same datasets. It is possible that in these cases, asking the model to generate world knowledge biases it more towards finding an entailment in general.

Theorem proving We secondly note that the labels inferred from the generated formulas are not always an improvement over the model's direct answer. For the neutral (or non-entailment) columns, we see that the Prover9-inferred label accuracy is typically higher than the direct answer (often much higher), but recall that we infer a neutral label whenever Prover9 cannot find a proof of the hypothesis (or its negation) from the premise and any relevant world-knowledge formulas, so these apparently good results for neutral items may just be evidence that the generated formulas don't fully capture the logical relationships that would be required to draw an inference if there were one. This can happen when the model produces formulas that are unrelated to each other for the wrong reasons (e.g., inconsistently translated predicates).

There are, however, several other cases where the inferred label accuracy shows a notable performance improvement over the direct answer. Prover9 accuracy is significantly better for contradiction-labeled items in the **negation** dataset, and it is somewhat better for entailments of the **lexical** dataset in the prompt scheme that includes world-knowledge formulas.

Overall, while there are some cases in which the label inferred from the generated formulas outperforms the direct-answer label, there are even more cases where generating the formulas improves the direct answer but the formulas themselves cannot be used to infer the correct label. This suggests that the utility of prompting the model for formulas is mostly in priming it to attend to the logical relationships between the natural language sentences. Upon closer inspection, we suspect that the logical representations for the premise and hypothesis are often not linked together logically, pushing the theorem prover towards Neutral.

4 Conclusion

This work investigates the use of constrained decoding and LLM prompting for Natural Language Inference. We specifically test three setups: (1) An LLM is prompted to solve the task directly; (2) the LLM first is prompted to generate logic formulas and subsequently re-prompted to use those formulas to provide an answer; and (3) the model is also prompted to generate formulas capturing any relevant lexical knowledge before answering. Generated formulas are also fed to a theorem prover. We observe that while the theorem prover may help in cases of entailment and non-entailment, the overall performance is highest for the two-step prompting approach of letting the model decide based on its own generated formulas.

Limitations We consider this work a pilot study investigating the applicability of constrained decoding to support NLI systems based on LLM prompting. This leads to a number of lessons of this work: (1) Priming an LLM by asking for logical representations increases performance on challenging NLI test sets; (2) Generating logical representations with only prompt examples as gold standard is too primitive to use in combination with theorem proving; (3) Adding world knowledge can mitigate the gold standard issue; (4) With the adequate combination of representation format, language model, and prompt setup one may push the limits of NLI; (5) Constrained decoding can play a role in controlling LLM output and overall performance.

Future Work The findings in this paper warrant a lot of future work; for example, the lack of gold standard data in the test sets we used makes it difficult for the model to tune its generated logical representations, so ideally a gold standard is required.

Another underrepresented concept is a change in representation format, where predicate logic formulas could be encoded in formats more represented in LLM training data, such as Python code, or Z3 statements. Natural logic may also be a promising output format for LLMs due to its closeness to natural language (Lakoff, 1970).

Acknowledgments

This work was supported by grant 2014-39 from the Swedish Research Council (VR) for the establishment of the Centre for Linguistic Theory and Studies in Probability (CLASP) at the University of Gothenburg.

References

Lasha Abzianidze. 2020. Learning as abduction: Trainable natural logic theorem prover for natural language inference. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 20–31, Barcelona, Spain (Online). Association for Computational Linguistics.

Lasha Abzianidze and Konstantinos Kogkalidis. 2021. A logic-based framework for natural language inference in dutch. *arXiv preprint arXiv:2110.03323*.

Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2024. Guiding LLMs the right way: Fast, non-invasive constrained generation. In *Forty-first International Conference on Machine Learning*.

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in NLI: Ways (not) to go beyond simple heuristics. In *Proceedings of the Second Workshop on Insights from Negative Results in NLP*, pages 125–135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Zeming Chen and Qiyue Gao. 2022. Curriculum: A broad-coverage benchmark for linguistic phenomena in natural language understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3204–3219, Seattle, United States. Association for Computational Linguistics.

Zeming Chen, Qiyue Gao, and Lawrence S. Moss. 2021. NeuralLog: Natural language inference with joint neural and logical reasoning. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pages 78–88, Online. Association for Computational Linguistics.

Robin Cooper, Dick Crouch, Jan Van Eijck, Chris Fox, Johan Van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, et al. 1996. Using the framework. Technical report, Technical Report LRE 62-051 D-16, The FraCaS Consortium.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10932–10952, Singapore. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Aikaterini-Lida Kalouli, Hai Hu, Alexander F. Webb, Lawrence S. Moss, and Valeria de Paiva. 2023. Curing the SICK and other NLI maladies. *Computational Linguistics*, 49(1):199–243.

George Lakoff. 1970. Linguistics and natural logic. *Synthese*, 22(1):151–271.

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).

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.

W. McCune. 2005–2010. Prover9 and Mace4. `http://www.cs.unm.edu/~mccune/prover9/`.

Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. Sources of hallucination by large language models on inference tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.

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.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Leonardo Ranaldi, Marco Valentino, Alexander Polonsky, and Andrè Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. *arXiv preprint arXiv:2502.12616*.

Kyle Richardson, Hai Hu, Lawrence Moss, and Ashish Sabharwal. 2020. Probing natural language inference models through semantic fragments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8713–8721.

Martin Schmitt and Hinrich Schütze. 2021. Language Models for Lexical Inference in Context. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1267–1280, Online. Association for Computational Linguistics.

Riko Suzuki, Hitomi Yanaka, Masashi Yoshikawa, Koji Mineshima, and Daisuke Bekki. 2019. Multimodal logical inference system for visual-textual entailment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 386–392, Florence, Italy. Association for Computational Linguistics.

Aarne Talman and Stergios Chatzikyriakidis. 2019. Testing the generalization power of neural network models across NLI benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 85–94, Florence, Italy. Association for Computational Linguistics.

Akiyoshi Tomihari and Hitomi Yanaka. 2023. Logic-based inference with phrase abduction using vision-and-language models. *IEEE Access*, 11:45645–45656.

Gijs Wijnholds. 2023. Assessing monotonicity reasoning in Dutch through natural language inference. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1494–1500, Dubrovnik, Croatia. Association for Computational Linguistics.

Brandon T Willard and Rémi Louf. 2023. Efficient guided generation for large language models. *arXiv e-prints*, pages arXiv–2307.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019a. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. 2019b. Can Neural Networks Understand Monotonicity Reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy. Association for Computational Linguistics.

A LLM prompts

Figures 1, 2, and 3 show the three different prompt templates used to elicit NLI classifications from the LLM. Only the `ITEM_PREMISE` and `ITEM_HYPOTHESIS` vary by item, whereas the `DATASET_LABELS` and `EXAMPLE_*` fields vary by dataset. The `LM_*` fields are filled in by the LLM.

The specific few-shot examples used for each dataset can be found in Figures 4–9.

```
Given a pair of sentences, the task is to determine whether Sentence A entails Sentence B by labeling
    ↪  the pair with <DATASET_LABELS>.

Sentence A: <EXAMPLE_1_PREMISE>
Sentence B: <EXAMPLE_1_HYPOTHESIS>
###
The relation between Sentence A and Sentence B is: <EXAMPLE_1_LABEL>
∼ REPEATED FOR EXAMPLES 2 and 3 ∼

Sentence A: <ITEM_PREMISE>
Sentence B: <ITEM_HYPOTHESIS>
###
The relation between Sentence A and Sentence B is: <LM_DIRECT_ANSWER>
```

Figure 1: The prompt asking the model for a direct answer. <DATASET_LABELS> is adaptable, depending on whether the task at hand uses binary (RTE) or ternary (NLI) classification; <LM_DIRECT_ANSWER> is constrained by the relevant label set; the three few-shot examples are specific to the dataset the item comes from (see Figures 4-9).

```
 Given a pair of sentences, the task is to parse each sentence into first-order logic formulas and
     ↪ then determine whether Sentence A entails Sentence B by labeling the pair with
     ↪ <DATASET_LABELS>.
The grammar of first-order logic formulas is defined as follows:
1) logical conjunction of expr1 and expr2: expr1 & expr2
2) logical disjunction of expr1 and expr2: expr1 | expr2
3) logical negation of expr1: -expr1
5) expr1 implies expr2: expr1 -> expr2
6) expr1 if and only if expr2: expr1 <-> expr2
7) logical universal quantification over expr1: forall x. expr1
8) logical existential quantification over expr1: exists x. expr1

Sentence A: <EXAMPLE_1_PREMISE>
Sentence B: <EXAMPLE_1_HYPOTHESIS>
###
Formula A: <EXAMPLE_1_PREMISE> ::: <EXAMPLE_1_PREMISE_FORMULA>
Formula B: <EXAMPLE_1_PREMISE> ::: <EXAMPLE_1_HYPOTHESIS_FORMULA>
###
The relation between Sentence A and Sentence B is: <EXAMPLE_1_LABEL>
∼ REPEATED FOR EXAMPLES 2 and 3 ∼

Sentence A: <ITEM_PREMISE>
Sentence B: <ITEM_HYPOTHESIS>
###
Formula A: <ITEM_PREMISE> ::: <LM_PREMISE_FORMULA>
Formula B: <ITEM_HYPOTHESIS> ::: <LM_HYPOTHESIS_FORMULA>
###
The relation between Sentence A and Sentence B is: <LM_DIRECT_ANSWER>
```

Figure 2: The prompt asking the model for first-order logic formulas for the premise and hypothesis, as well as a direct answer. <DATASET_LABELS> is adaptable, depending on whether the task at hand uses binary (RTE) or ternary (NLI) classification; <LM_DIRECT_ANSWER> is constrained by the relevant label set; the three few-shot examples are specific to the dataset the item comes from (see Figures 4-9).

```
Given a pair of sentences, the task is to parse each sentence into first-order logic formulas, then
    ↪ write first-order logic formulas that capture any relevant lexical knowledge, and finally
    ↪ determine whether Sentence A entails Sentence B by labeling the pair with <DATASET_LABELS>.
1) logical conjunction of expr1 and expr2: expr1 & expr2
2) logical disjunction of expr1 and expr2: expr1 | expr2
3) logical negation of expr1: -expr1
5) expr1 implies expr2: expr1 -> expr2
6) expr1 if and only if expr2: expr1 <-> expr2
7) logical universal quantification over expr1: forall x. expr1
8) logical existential quantification over expr1: exists x. expr1

Sentence A: <EXAMPLE_1_PREMISE>
Sentence B: <EXAMPLE_1_HYPOTHESIS>
###
Formula A: <EXAMPLE_1_PREMISE> ::: <EXAMPLE_1_PREMISE_FORMULA>
Formula B: <EXAMPLE_1_PREMISE> ::: <EXAMPLE_1_HYPOTHESIS_FORMULA>
###
Lexical knowledge:
<EXAMPLE_1_WORLD_KNOWLEDGE_FORMULA_1>
. . .
<EXAMPLE_1_WORLD_KNOWLEDGE_FORMULA_N>
###
The relation between Sentence A and Sentence B is: <EXAMPLE_1_LABEL>
∼ REPEATED FOR EXAMPLES 2 and 3 ∼

Sentence A: <ITEM_PREMISE>
Sentence B: <ITEM_HYPOTHESIS>
###
Formula A: <ITEM_PREMISE> ::: <LM_PREMISE_FORMULA>
Formula B: <ITEM_HYPOTHESIS> ::: <LM_HYPOTHESIS_FORMULA>
###
Lexical knowledge:
<LM_WORLD_KNOWLEDGE_FORMULAS>
###
The relation between Sentence A and Sentence B is: <LM_DIRECT_ANSWER>
```

Figure 3: The prompt asking the model for first-order logic formulas for the premise and hypothesis, as well as a direct answer. <DATASET_LABELS> is adaptable, depending on whether the task at hand uses binary (RTE) or ternary (NLI) classification; <LM_DIRECT_ANSWER> is constrained by the relevant label set; <LM_WORLD_KNOWLEDGE_FORMULAS> is constrained to be a newline-separated list of strings; the three few-shot examples are specific to the dataset the item comes from (see Figures 4-9).

```
EXAMPLE_1 = { # example id 4598
    'PREMISE' = "The purple alien drank soda."
    'HYPOTHESIS' = "The purple alien drank coke."
    'PREMISE_FORMULA' = "exists x. exists y. Purple(x) & Alien(x) & Soda(y) & Drank(x, y)"
    'HYPOTHESIS_FORMULA' = "exists x. exists y. Purple(x) & Alien(x) & Coke(y) & Drank(x, y)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. (Coke(x) -> Soda(x)),
    ]
}

EXAMPLE_2 = { # example id 5075
    'PREMISE' = "Nobody danced."
    'HYPOTHESIS' = "Nobody moved."
    'PREMISE_FORMULA' = "forall x. -Dance(x)"
    'HYPOTHESIS_FORMULA' = "forall x. -Move(x)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. (Dance(x) -> Move(x)),
    ]
}

EXAMPLE_3 = { # example id 54
    'PREMISE' = "All animals like to scratch their ears."
    'HYPOTHESIS' = "All dogs like to scratch their ears."
    'PREMISE_FORMULA' = "forall x. (Animal(x) -> LikesToScratchEars(x, x))"
    'HYPOTHESIS_FORMULA' = "forall x. (Dog(x) -> LikesToScratchEars(x, x))"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. (Dog(x) -> Animal(x)),
    ]
}
```

Figure 4: Few-shot examples for items from the **monotonicity** dataset. CURRICULUM item ids: 4598 5075 54

```
EXAMPLE_1 = { # example id 2675
    'PREMISE' = "Ruben is as tall as Jack , Jack is as tall as Francis , Francis is as tall as Gordon
        ↪  , Gordon is as tall as Bruce , Bruce is as tall as Alan , Alan is as tall as Danny ,
        ↪ Danny is taller than Allen"
    'HYPOTHESIS' = "Keith is taller than Alan"
    'PREMISE_FORMULA' = "AsTallAs(ruben, jack) & AsTallAs(jack, francis) & AsTallAs(francis, gordon)
        ↪ & AsTallAs(gordon, bruce) & AsTallAs(bruce, alan) & AsTallAs(alan, danny) & TallerThan(
        ↪ Danny, alan)"
    'HYPOTHESIS_FORMULA' = "TallerThan(keith, alan)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. forall y. TallerThan(x, y) -> -AsTallAs(y, x),
       forall x. forall y. forall z. (TallerThan(x, y) & TallerThan(y, z)) -> TallerThan(x, z),
    ]
}

EXAMPLE_2 = { # example id 648
    'PREMISE' = "Russell is taller than Oscar, Terrance, Lawrence, Dan, Felix, Todd, Alex, Jose and
        ↪ Harry , Russell is as tall as Clifton"
    'HYPOTHESIS' = "Felix is taller than Clifton"
    'PREMISE_FORMULA' = "TallerThan(russell, oscar) & TallerThan(russell, terrance) & TallerThan(
        ↪ russell, dan) & TallerThan(russell, felix) & TallerThan(russell, todd) & TallerThan(
        ↪ russell, alex) & TallerThan(russell, jose) & TallerThan(russell, harry) & AsTallAs(
        ↪ russell, clifton)"
    'HYPOTHESIS_FORMULA' = "TallerThan(felix, clifton)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. forall y. forall z. (AsTallAs(x, y) & TallerThan(x, z) -> TallerThan(y, z)),
       forall x. forall y. (TallerThan(x, y) -> -TallerThan(y, x)),
    ]
}

EXAMPLE_3 = { # example id 1421
    'PREMISE' = "Jesse is as tall as Paul , Paul is as tall as Terry , Terry is as tall as Sidney ,
        ↪ Sidney is as tall as Luis , Luis is as tall as Andy , Andy is as tall as Freddie ,
        ↪ Freddie is as tall as Adrian , Adrian is taller than James"
    'HYPOTHESIS' = "Luis is taller than James"
    'PREMISE_FORMULA' = "AsTallAs(jesse, paul) & AsTallAs(paul, terry) & AsTallAs(terry, sidney) &
        ↪ AsTallAs(sidney, luis) & AsTallAs(luis, andy) & AsTallAs(andy, freddie) & AsTallAs(
        ↪ freddie, adrian) & TallerThan(adrian, james)"
    'HYPOTHESIS_FORMULA' = "TallerThan(luis, james)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. forall y. forall z. (AsTallAs(x, y) & TallerThan(y, z) -> TallerThan(x, z)),
       forall x. forall y. forall z. (AsTallAs(x, y) & AsTallAs(y, z) -> AsTallAs(x, z)),
    ]
}
```

Figure 5: Few-shot examples for items from the **comparative** dataset. CURRICULUM item ids: 2675 648 1421

```
EXAMPLE_1 = { # example id 2200
    'PREMISE' = "Tony has not visited Beaverton, Johnny has not visited Long Beach, Ken has visited
        ↪ Kingston and if Tony has not visited Beaverton then Fred has not visited Danville"
    'HYPOTHESIS' = "Fred has not visited Danville"
    'PREMISE_FORMULA' = "-Visited(tony, beaverton) & -Visited(johnny, long_beach) & Visited(ken,
        ↪ kingston) & (-Visited(tony, beaverton) -> -Visited(fred, danville))"
    'HYPOTHESIS_FORMULA' = "-Visited(fred, danville)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}

EXAMPLE_2 = { # example id 2699
    'PREMISE' = "Felix has not visited Pampa, William has not visited Bessemer, Eddie has visited
        ↪ Grants Pass and if Felix has visited Pampa then Danny has visited Belmont"
    'HYPOTHESIS' = "Danny has visited Belmont"
    'PREMISE_FORMULA' = "-Visited(felix, pampa) & -Visited(william, bessemer) & Visited(eddie,
        ↪ grants_pass) & (Visited(felix, pampa) -> Visited(danny, belmont))"
    'HYPOTHESIS_FORMULA' = "Visited(danny, belmont)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}

EXAMPLE_3 = { # example id 1384
    'PREMISE' = "Don has not visited Norwich, Alberto has visited Nevada, Wallace has visited Wyoming
        ↪  and if Alberto has visited Nevada then Sam has visited Arcadia"
    'HYPOTHESIS' = "Sam has not visited Arcadia"
    'PREMISE_FORMULA' = "-Visited(don, norwhich) & Visited(alberto, nevada) & Visited(wallace,
        ↪ wyoming) & (Visited(alberto, nevada) -> Visited(sam, arcadia))"
    'HYPOTHESIS_FORMULA' = "-Visited(sam, arcadia)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}
```

Figure 6: Few-shot examples for items from the **conditional** dataset. CURRICULUM item ids: 2200 2699 1384

```
EXAMPLE_1 = { # example id 1933
    'PREMISE' = "A stricken butterfly has Wings."
    'HYPOTHESIS' = "A stricken butterfly wavers on Wings."
    'PREMISE_FORMULA' = "exists x. exists y.(Stricken(x) & Butterfly(x) & Wings(y) & Has(x, y))"
    'HYPOTHESIS_FORMULA' = "exists x. exists y.(Stricken(x) & Butterfly(x) & Wings(y) & WaversOn(x, y
        ↪ ))"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       forall x. forall y. (Has(x, y) -> WaversOn(x, y)),
    ]
}

EXAMPLE_2 = { # example id 3662
    'PREMISE' = "Sadat beat Jimmy Carter."
    'HYPOTHESIS' = "Jimmy Carter secluded Sadat."
    'PREMISE_FORMULA' = "Beat(sadat, jimmy_carter)"
    'HYPOTHESIS_FORMULA' = "Secluded(jimmy_carter, sadat)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       -(forall x. forall y. (Beat(x, y) -> Secluded(y, x))),
    ]
}
```

Figure 7: Few-shot examples for items from the **lexical** dataset. CURRICULUM item ids: 1933 3662

```
EXAMPLE_1 = { # example id 1874
    'PREMISE' = "Harry has only visited Bhutan, Curtis has only visited Ecuador, Roland has only
        ↪ visited Philippines, Tom has only visited Uganda, Darren has only visited Jordan, Byron
        ↪ has only visited Cameroon, Willie has only visited Vanuatu, Brett has only visited North
        ↪ Korea"
    'HYPOTHESIS' = "Curtis didn't visit Belize"
    'PREMISE_FORMULA' = "Visit(harry, bhutan) & forall x. (Visit(harry, x) -> x = bhutan) & Visit(
        ↪ curtis, ecuador) & forall x. (Visit(curtis, x) -> x = ecuador) & Visit(roland,
        ↪ philippines) & forall x. (Visit(roland, x) -> x = philippines) & Visit(tom, uganda) &
        ↪ forall x. (Visit(tom, x) -> x = uganda) & Visit(darren, jordan) & forall x. (Visit(darren,
        ↪  x) -> x = jordan) & Visit(byron, cameroon) & forall x. (Visit(byron, x) -> x = cameroon)
        ↪  & Visit(willie, vanuatu) & forall x. (Visit(willie, x) -> x = vanuatu) & Visit(brett,
        ↪ north_korea) & forall x. (Visit(brett, x) -> x = north_korea)"
    'HYPOTHESIS_FORMULA' = "-Visit(curtis, belize)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       belize != ecuador,
    ]
}

EXAMPLE_2 = { # example id 2526
    'PREMISE' = "Howard has only visited Croatia, Karl has only visited Kosovo"
    'HYPOTHESIS' = "Ross didn't visit Croatia"
    'PREMISE_FORMULA' = "Visit(howard, croatia) & forall x. (Visit(howard, x) -> x = croatia) & Visit
        ↪ (karl, kosovo) & forall x. (Visit(karl, x) -> x = kosovo)"
    'HYPOTHESIS_FORMULA' = "-Visit(ross, croatia)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
       howard != ross,
    ]
}

EXAMPLE_3 = { # example id 581
    'PREMISE' = "Thomas has only visited Romania, Adrian has only visited Tuvalu, Everett has only
        ↪ visited Djibouti, Marc has only visited Dominica, Don has only visited China, Nicholas
        ↪ has only visited Turkmenistan, Lonnie has only visited Iraq, Theodore has only visited
        ↪ North Korea, Andrew has only visited Nepal, Ken has only visited Saint Lucia, Terrence
        ↪ has only visited Liberia"
    'HYPOTHESIS' = "Ken didn't visit Saint Lucia"
    'PREMISE_FORMULA' = "Visit(thomas, romania) & forall x. (Visit(thomas, x) -> x = romania) & Visit
        ↪ (adrian, tuvalu) & forall x. (Visit(adrian, x) -> x = tuvalu) & Visit(everett, djibouti)
        ↪ & forall x. (Visit(everett, x) -> x = djibouti) & Visit(marc, dominica) & forall x. (
        ↪ Visit(marc, x) -> x = dominica) & Visit(don, china) & forall x. (Visit(don, x) -> x =
        ↪ china) & Visit(nicholas, turkmenistan) & forall x. (Visit(nicholas, x) -> x =
        ↪ turkmenistan) & Visit(lonnie, iraq) & forall x. (Visit(lonnie, x) -> x = iraq) & Visit(
        ↪ theodore, korea) & forall x. (Visit(theodore, x) -> x = korea) & Visit(andrew, nepal) &
        ↪ forall x. (Visit(andrew, x) -> x = nepal) & Visit(ken, saint_lucia) & forall x. (Visit(
        ↪ ken, x) -> x = saint_lucia) & Visit(terrence, liberia) & forall x. (Visit(terrence, x) ->
        ↪  x = liberia)"
    'HYPOTHESIS_FORMULA' = "-Visit(ken, saint_lucia)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}
```

Figure 8: Few-shot examples for items from the **negation** dataset. CURRICULUM item ids: 1874 2526 581

```
EXAMPLE_1 = { # example id 2857
    'PREMISE' = "Everyone has visited Lesotho, Botswana, Cambodia, Kyrgyzstan, Lithuania, Tonga,
        ↪ Suriname, Costa Rica, Thailand, Bangladesh, New Zealand, Nigeria, Pakistan, Palau, Libya,
        ↪  Bosnia & Herzegovinia, United Arab Emirates, Chad, Solomon Islands and Ireland"
    'HYPOTHESIS' = "That person there did visit Libya"
    'PREMISE_FORMULA' = "forall x. Visit(x,lesotho) & Visit(x, botswana) & Visit(x, cambodia) & Visit
        ↪ (x, kyrgyzstan) & Visit(x, lithuania) & Visit(x, tonga) & Visit(x, suriname) & Visit(x,
        ↪ costa_rica) & Visit(x, thailand) & Visit(x, bangladesh) & Visit(x, new_zealand) & Visit(x,
        ↪  nigeria) & Visit(x, pakistan) & Visit(x, palau) & Visit(x, libya) & Visit(x,
        ↪ bosnia_herzegovinia) & Visit(x, united_arab_emirates) & Visit(x, chad) & Visit(x,
        ↪ solomon_islands) & Visit(x, ireland)"
    'HYPOTHESIS_FORMULA' = "exists x. Visit(x, libya)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}

EXAMPLE_2 = { # example id 1121
    'PREMISE' = "Everyone has visited Togo, Saudi Arabia, Malta, Bosnia & Herzegovinia, Gabon, Sierra
        ↪  Leone, El Salvador, The Bahamas, Mongolia, Mali and Djibouti"
    'HYPOTHESIS' = "Roland didn't visit Gabon"
    'PREMISE_FORMULA' = "forall x. Visit(x, togo) & Visit(x, saudi_arabia) & Visit(x, malta) & Visit(
        ↪ x, bosnia_herzegovinia) & Visit(x, gabon) & Visit(x, sierra_leone) & Visit(x, el_salvador)
        ↪  & Visit(x, bahamas) & Visit(x, mongolia) & Visit(x, mali) & Visit(x, djibouti)"
    'HYPOTHESIS_FORMULA' = "-Visit(roland, gabon)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}

EXAMPLE_3 = { # example id 2850
    'PREMISE' = "Someone has visited every person and every place"
    'HYPOTHESIS' = "That person there didn't visit United States"
    'PREMISE_FORMULA' = "exists x. forall y. (Person(y) | Place(y)) -> Visit(x, y)"
    'HYPOTHESIS_FORMULA' = "exists x. Person(x) & -Visit(x, united_states)"
    'WORLD_KNOWLEDGE_FORMULAS' = [
    ]
}
```

Figure 9: Few-shot examples for items from the **quantifier** dataset. CURRICULUM item ids: 2857 1121 2850

B Prover9 Errors

The LLM generated largely syntactically correct formulas of first-order logic, especially when provided with few-shot examples. However, there were cases where the generated formulas resulted in an error when fed into Prover9. In Tables 1 and 2 the P9 columns present label-wise accuracy results that have been filtered to exclude items where Prover9 generated an error (i.e., the denominator of the accuracy metric does not include those items). In most cases, the difference is small—note that considering the filtered or un-filtered version never changes whether the Prover9 result is higher than the direct answer). For completeness, this section shows a comparison of the filtered and un-filtered results.

A total of 310 errors were encountered over 3 600 generated formula sets (300 for each of 6 datasets and 2 prompt schemes). The breakdown of errors encountered can be found in Table 5.

	Label	E		C		N	
		⛛	▽	⛛	▽	⛛	▽
Dataset	*Prompt*						
comparative	forms	8.2	8.2	0.0	0.0	99.0	99.0
	forms+wk	100.0	100.0	0.0	0.0	0.0	0.0
conditional	forms	57.9	59.8	48.4	50.0	100.0	100.0
	forms+wk	35.8	37.0	47.4	48.4	100.0	100.0
negation	forms	0.0	0.0	97.8	98.9	96.5	100.0
	forms+wk	0.0	0.0	93.3	98.8	95.7	98.2
quantifier	forms	57.8	59.8	24.6	27.2	97.9	100.0
	forms+wk	61.1	64.0	21.1	23.5	96.9	100.0

Table 3: Unfiltered (⛛) and filtered (▽) mean label-wise accuracy for NLI classification when using the LLM-generated formulas to infer the label with Prover9.

	Label	E		N/C	
		⛛	▽	⛛	▽
Dataset	*Prompt*				
lexical	forms	1.3	1.5	89.9	100.0
	forms+wk	52.0	65.3	27.2	34.8
monotonicity	forms	13.7	16.7	74.5	90.9
	forms+wk	36.0	50.0	36.6	54.6

Table 4: Unfiltered (⛛) and filtered (▽) mean label-wise accuracy for RTE classification when using the LLM-generated formulas to infer the label with Prover9.

Prover9 Error	forms	forms+wk
parsing error (unexpected symbol)	79	92
symbol used with multiple arities	25	65
symbol used as both relation and function	14	35

Table 5: Counts of error types encountered by Prover9 when given the LLM-generated formulas. A total of 1 800 sets of formulas were generated for each prompt scheme (*forms* and *forms+wk*).

Building a Compact Math Corpus

Andrea Ferreira
Independent Researcher
andreafer.uni@gmail.com

Abstract

This paper introduces the Compact Math Corpus (CMC), a preliminary resource for natural language processing in the mathematics domain. We process three open-access undergraduate textbooks from distinct mathematical areas and annotate them in the CoNLL-U format using a lightweight pipeline based on the spaCy Small model. The structured output enables the extraction of syntactic bigrams and TF-IDF scores, supporting a syntactic-semantic analysis of mathematical sentences.

From the annotated data, we construct a classification dataset comprising bigrams potentially representing mathematical concepts, along with representative example sentences. We combine CMC with the conversational corpus UD English EWT and train a logistic regression model with K-fold cross-validation, achieving a minimum macro-F1 score of 0.989. These results indicate the feasibility of automatic concept identification in mathematical texts.

The study is designed for easy replication in low-resource settings and to promote sustainable research practices. Our approach offers a viable path to tasks such as parser adaptation, terminology extraction, multiword expression modeling, and improved analysis of mathematical language structures.

1 Introduction

Mathematical textbooks, though precise and structured, present unique challenges to standard Natural Language Processing (NLP) tools. Their language differs significantly from general-domain English, incorporating symbolic notation, diagrams, and domain-specific terminology. Consequently, models trained primarily on non-technical corpora often underperform on this type of texts.

Recent benchmarks such as MATHVISTA (Lu et al., 2024) illustrate these challenges. Even advanced vision-language models, for example: GPT-4V, achieve accuracy in the range 50% when tasked with understanding mathematical content (see Figure 3, Appendix A). Meanwhile, datasets including GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) focus on mathematical problem solving, providing question–answer pairs, but lack the linguistic annotations necessary for syntactic or semantic analysis.

Other resources have been introduced for the processing of the mathematical language. An example is NaturalProofs (Welleck et al., 2021), which focuses on theorem proving and alignment of formal and informal proofs, but does not address the broader expository writing found in instructional or pedagogical texts. This scarcity limits the development and evaluation of NLP tools tailored for mathematical language. The **Compact Math Corpus (CMC)** aims to help bridge this gap by offering a preliminary, automatically annotated resource, not a gold standard, but a practical starting point for linguistic processing in this domain.

The CMC is built from three open-access undergraduate textbooks: *Abstract Algebra: Theory and Applications* (Judson, 2022), *Linear Algebra* (Hefferon, 2022), and *Discrete Mathematics: An Open Introduction* (Levin, 2024), all sourced from the Open Math Textbook Initiative[1]. Using a computationally lean NLP pipeline based on the spaCy Small model (Honnibal et al., 2020), we automatically annotate these texts in the CoNLL-U format[2], capturing both morphological and syntactic features.

Designed for low-resource and sustainable settings (Luccioni et al., 2023), our pipeline prioritizes accessibility and replicability. To assess its utility, we extract syntactic bigrams from CMC, combine them with the conversational UD English EWT corpus (Silveira et al., 2014), and pair each bigram with a representative sentence from its source cor-

[1] https://textbooks.aimath.org/
[2] https://universaldependencies.org/

Proceedings of the 5th Workshop on Natural Logic Meets Machine Learning (NALOMA), pages 48–55

pus. We then train a logistic regression model using scikit-learn's `LogisticRegression` with K-fold cross-validation. The model achieves a minimum macro-F1 score of 0.989, indicating that cost-effective methods can effectively support the detection of mathematical concepts.

We release the annotated corpus and supporting materials on GitHub[3].

2 Corpus Preprocessing and Annotation

The Compact Math Corpus (CMC) was constructed from the three undergraduate-level textbooks introduced in the previous section, each covering a distinct area of mathematics. All texts are openly licensed and available in PDF format.

Although LaTeX is generally a more suitable format for mathematical texts due to its richer structural markup (Collard et al., 2024; de Paiva et al., 2023), most educational materials are distributed in PDF format, which aligns better with our goal of scalability.

To better understand the trade-offs involved, we processed a matched section from the *Linear Algebra* textbook in both LaTeX and PDF formats. The source LaTeX was converted to JSON using `pylatexenc`[4], and the PDF using `PyMuPDF`[5]. Both outputs were then passed through the same annotation pipeline, and we extracted compounds[6] from the resulting CoNLL-U files.

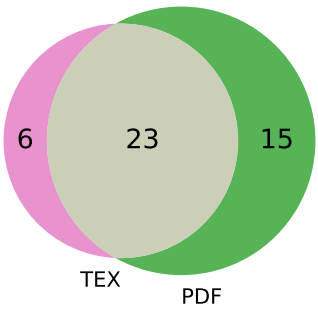


Figure 1: Overlap of Compounds Between LaTeX and PDF.

Of the total compounds detected, 23 appeared in both formats, 6 were exclusive to LaTeX, and 15 were exclusive to PDF (see Figure 1). These results indicate that, despite some discrepancies, PDF-based processing yields comparable compound extraction quality, an encouraging outcome given the prevalence and accessibility of PDF materials in educational settings.

2.1 Preprocessing

Following best practices in corpus development (Collard et al., 2024), PDF textbooks were converted to structured JSON to support consistent downstream analysis.

Linguistic content was extracted using the `PyMuPDF` library, which provides raw text and layout information. Since PDFs prioritize visual formatting over semantic structure, extraction introduced common issues, including token merging, incorrect sentence segmentation, hyphenation artifacts, and irregular or misplaced line breaks.

To address these concerns, we applied a preprocessing pipeline with anomaly detection and cleaning steps, including sentence filtering, non-ASCII character removal, and format normalization. The cleaned text was stored in JSON and parsed using `spaCy Small` to generate part-of-speech and dependency annotations in `CoNLL-U` format[7].

This setup enables us to evaluate the performance of general-purpose NLP tools on math-heavy texts and points to challenges such as symbolic content, domain-specific terminology, and structural noise, areas where parser adaptation or multimodal integration may improve outcomes.

2.2 Linguistic Annotation

To assess the performance–efficiency balance of our method, we compared `spaCy`'s small model (`en_core_web_sm`) with its transformer-based counterpart (`en_core_web_trf`) on a section of the *Linear Algebra* textbook.

As shown in Figure 2, both models produced nearly identical counts of tokens, lemmas, and unique words. The main difference was in sentence segmentation, where the transformer model generated more sentence boundaries. However, this did not affect the performance of our downstream classification task, supporting the adequacy of the compact model given its streamlined computational requirements.

[3] https://github.com/andreafer-uni/Compact-Math-Corpus

[4] https://pylatexenc.readthedocs.io/

[5] https://pymupdf.readthedocs.io/

[6] In UD, a *compound* refers to a noun–noun construction where one noun modifies another.

[7] See (Nivre et al., 2016) for a complete overview of the UD framework and `CoNLL-U` structure.

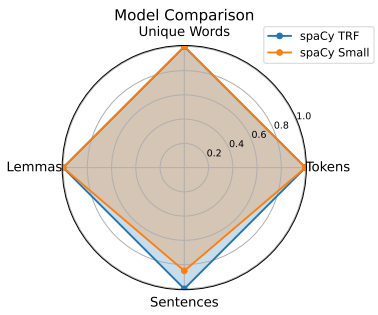


Figure 2: Sentence segmentation differences between transformer-based and small models.

To enhance the linguistic value of the annotation process, the use of the CoNLL-U format provides substantial linguistic benefits. In contrast to unstructured text, syntactically annotated corpora enable the systematic recognition of domain-specific constructions such as compounds and multiword expressions (MWEs), which act as indicators of domain-specific language (Collard et al., 2022).

As highlighted in Table 1, the integration of syntactic information, specifically the dependency relations captured by Universal Dependencies (UD), increased the prominence of mathematically relevant bigrams in the Compact Math Corpus (CMC). For example, the terms *vector space*, *linear combination*, and *closed formula* not only received high TF-IDF scores after annotation, but also aligned well with the core mathematical concepts.

Before CoNLL-U		**After CoNLL-U**	
Bigram	**TF-IDF**	**Bigram**	**TF-IDF**
vector space	1328.53	vector space	1957.41
closed formula	834.34	closed formula	1105.03
linear combination	726.46	linear combination	1008.42
recurrence relation	668.48	recurrence relation	831.47
bit string	585.78	bit string	804.58

Table 1: Top 5 TF-IDF bigrams in the Compact Math Corpus before and after CoNLL-U annotation.

The syntactic layer, though often optional in modern NLP pipelines, remains crucial for applications that prioritize interpretability and robustness. As noted by Sag et al. (Sag et al., 2002), the combination of symbolic and statistical methods produces a more complete view of language, particularly in detecting MWEs that define mathematical vocabulary.

Although the CMC is not manually annotated, its syntactic enrichment via spaCy and CoNLL-U enables a hybrid pipeline in which frequency-based methods such as TF-IDF are grounded in structural patterns. This becomes evident when we compare the results before and after annotation. After syntactic parsing, high-ranking terms became more semantically coherent, whereas noisy entries (e.g. *many way*) were relatively de-emphasized.

Together, the CoNLL-U annotation process enhances the linguistic utility of the CMC by facilitating the extraction of interpretable, conceptually grounded MWEs, despite being derived from non-pretrained model. This makes the resource not only computationally efficient but also linguistically rich enough to support downstream tasks including classification and terminology extraction.

3 Dataset Construction

To investigate whether syntactic bigrams can help distinguish mathematical language from general English, we constructed a binary classification dataset combining the Compact Math Corpus (CMC) with the UD English Web Treebank (EWT), leveraging their shared CoNLL-U format.

3.1 The UD-EWT

The UD English Web Treebank (EWT) is a gold-standard corpus that provides syntactic and morphological annotations for English as part of the Universal Dependencies (UD) project. It contains informal web-based texts from blogs, emails, forums, product reviews, and Q&A websites such as Yahoo! Answers.

Due to its conversational and general-domain nature, EWT serves as a useful baseline for comparison with domain-specific corpora such as CMC. Since both CMC and UD-EWT follow the same annotation format, we used this compatibility to extract and compare syntactic features between corpora.

3.2 Bigrams and Labeling

To assess the potential of syntactic bigrams for the recognition of mathematical concepts, we first extracted bigrams from UD-EWT using the same TF-IDF methodology applied to CMC. We then compared the resulting lists and removed all overlapping bigrams, retaining only the unique ones from each corpus.

This comparison produced two distinct sets of bigrams: one containing candidates for mathemat-

ical concepts (unique to CMC) and another reflecting general English patterns (unique to EWT). For each bigram, we retrieved representative sentences from their respective corpora in which the bigrams occurred. These sentences were labeled as `True` (mathematical concept) or `False` (non-mathematical), resulting in a dataset of 2,796 sentences evenly balanced between mathematical and general-domain content.

The resulting dataset forms the basis for the classification task described in the next section, where we evaluate the feasibility of concept recognition using a logistic regression model.

4 Identifying Mathematical Concepts

To evaluate whether syntactic bigrams can effectively signal mathematical content, we framed a binary classification task: Given a sentence containing a candidate bigram, predict whether it expresses a mathematical concept. The classifier operates solely on text-based features, without access to labels or metadata from the source corpora.

4.1 Model and Setup

We trained a logistic regression model using TF-IDF features over unigrams and bigrams (max features = 5,000), implemented with `scikit-learn` (Pedregosa et al., 2011). We opted for logistic regression due to its efficiency, interpretability, and reliable performance in sparse feature spaces, well-suited to our minimal NLP setup.

4.2 Performance and Evaluation

When we evaluated the logistic regression model using 10-fold cross-validation over the dataset, the classifier achieved a macro F1-score of 0.996 ± 0.003, indicating consistent performance across all folds. Table 2 reports precision, recall, and F1-scores per class.

Although performance is reliable within the labeled dataset, it is important to note that the model relies on sparse, surface-level features and may struggle with ambiguous or out-of-distribution cases. However, it correctly recognized sentences containing previously unseen bigrams, indicating that the model generalizes based on sentence context rather than simple memorization.

Metric	Mean	Std
Accuracy	0.9964	0.0034
Precision_0	0.9934	0.0072
Recall_0	0.9993	0.0023
F1_0	0.9963	0.0035
Precision_1	0.9993	0.0022
Recall_1	0.9938	0.0069
F1_1	0.9965	0.0033
F1_Macro	0.9964	0.0034

Table 2: Cross-validation results (mean ± std) for logistic regression classifier.

4.3 Error Analysis and Interpretability

The confusion matrix (see Appendix B, Figure 4) confirms the model's consistent performance: only three mathematical examples were misclassified as non-mathematical, and no false positives were observed.

Zero-shot evaluations on held-out examples (Appendix C, Table 3) revealed a similar generalization capacity. The model correctly labeled unseen concepts such as *probability distribution* and *integral calculus* with high confidence, while remaining uncertain in borderline cases. An inspection of errors showed that school-related phrases such as *school project* and *homework folder* introduced ambiguity due to their lexical proximity to educational and mathematical contexts.

We further explored the behavior of the models through feature weight analysis (Appendix D, Figure 5). Some high-weight features aligned with intuitive mathematical concepts. However, others reflected corpus-specific statistical artifacts, terms *email* or document identifiers that lacked semantic relevance. This contrast illustrates how statistical models often rely on distributional regularities that may not align with human notions of meaning, underscoring the gap between symbolic interpretability and statistical association.

5 Conclusions and Future Directions

Summary of Findings. This paper presented the Compact Math Corpus (CMC), a syntactically annotated resource built from open-access instructional materials and processed using a lightweight NLP pipeline based on `spaCy Small` and the `CoNLL-U` format. Our goal was to enable structured linguistic analysis of mathematical language in a format that supports downstream applications

such as terminology extraction and concept classification.

Through a comparison of TF-IDF bigrams before and after annotation, we confirmed that syntactic information enhances the identification of multiword expressions aligned with core mathematical concepts. Furthermore, we constructed a balanced dataset by combining CMC with UD-EWT and found that a non-neural approach was able to effectively distinguish mathematical from general-domain content using only text-based features.

These findings indicate that even a non-pretrained, resource-efficient model can accurately detect domain-specific content, provided the input is linguistically enriched and well balanced. The approach is computationally efficient and interpretable, making it a suitable framework for educational or resource-constrained NLP scenarios. Although more powerful models may improve generalization on ambiguous input, our results provide evidence that structured features like those in CoNLL-U can support accurate concept classification in controlled settings.

Future Directions in Mathematical NLP Future work will explore expanding the CMC with additional textbooks across a broader range of mathematical topics, as well as refining the annotation pipeline to improve linguistic coverage.

Furthermore, the classification task can be extended by incorporating richer context (e.g., surrounding sentences or section-level cues) and testing generalization on unseen technical or scientific domains. Developing a small-scale gold-standard evaluation set with human-labeled concept phrases could further support benchmarking and model calibration.

An open direction for future research involves integrating symbolic representations, such as formulas or equation labels, with the current linguistic layer remaining an open direction. Despite being outside the scope of this initial study, such integration would enable a more comprehensive modeling of mathematical discourse, combining syntactic structure with symbolic reasoning.

6 Limitations

The main limitations of this study stem from its resource-constrained setup and reliance on general-purpose tools not specialized for mathematical language. The use of the `spaCy Small` model, while efficient and accessible, introduces trade-offs in parsing accuracy, particularly for domain-specific terminology, symbolic expressions, and non-standard syntactic constructions typical of mathematical discourse.

Another concern involves the quality of the input data. Although the `CoNLL-U` format assumes clean, well-formed text, most of our source material originated from PDFs, which are optimized for visual layout rather than semantic structure. This introduces common issues such as token merging and sentence-boundary errors. Despite preprocessing steps mitigated some of these problems, they did not completely eliminate noise.

A further challenge lies in the lack of a domain-specific, human-annotated gold standard. Without a reliable reference for comparison, it is difficult to measure parsing quality or validate the accuracy of syntactic analyses. This restricts our ability to compare tools rigorously or perform fine-grained error analysis. Creating a gold-standard treebank for mathematical texts within the `CoNLL-U` framework remains an open direction for future work.

Although compact models promote reproducibility and sustainability, they may underperform in complex linguistic contexts where transformer-based alternatives would offer greater accuracy. In our case, the simplicity of the classification task helped offset this topic, but it remains a relevant factor when considering more advanced downstream applications.

Acknowledgments

This work was inspired by Valeria de Paiva's leadership in the Network Mathematics project, which provided both direction and intellectual grounding.

We are also grateful to Jacob Collard for his earlier work on the mathematical topic, which offered valuable perspective throughout this study.

We acknowledge the American Institute of Mathematics for its exemplary initiatives to encourage the adoption of open-source and open-access textbooks in mathematics, an essential step toward equity and reproducibility in mathematical education and research.

Special thanks go to Diego Frassinelli, whose dedication to teaching and thoughtful guidance reflect the values of true academic mentorship.

References

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

Jacob Collard, Valeria de Paiva, Brendan Fong, and Eswaran Subrahmanian. 2022. Extracting mathematical concepts from text. *Preprint*, arXiv:2208.13830.

Jacob Collard, Valeria de Paiva, and Eswaran Subrahmanian. 2024. Mathematical entities: Corpora and benchmarks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11080–11089, Torino, Italia. ELRA and ICCL.

Valeria de Paiva, Qiyue Gao, Pavel Kovalev, and Lawrence S. Moss. 2023. Extracting mathematical concepts with large language models. *Preprint*, arXiv:2309.00642.

Jim Hefferon. 2022. *Linear Algebra*, 4 edition. Orthogonal Publishing L3C. Available online at `https://hefferon.net/linearalgebra/`.

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. *CoRR*, abs/2103.03874.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. Project homepage: `https://spacy.io`.

Thomas W. Judson. 2022. *Abstract Algebra: Theory and Applications*, annual edition 2022 edition. Stephen F. Austin State University. Available online at `http://abstract.pugetsound.edu/download/aata-20220728-sage-9.6.pdf`.

Oscar Levin. 2024. *Discrete Mathematics: An Open Introduction*, 4 edition. Self-published. Available online at `https://discrete.openmathbooks.org/`.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. 2023. Estimating the carbon footprint of bloom, a 176b parameter language model. *Journal of Machine Learning Research*, 24(253):1–15.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

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*, 12:2825–2830.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Conference on Intelligent Text Processing and Computational Linguistics*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D Manning. 2014. A gold standard dependency corpus for english. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC)*, pages 2897–2904. European Language Resources Association (ELRA).

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. *Preprint*, arXiv:2104.01112.

Appendix

This appendix provides supplementary material, including detailed linguistic annotations in the CoNLL-U style, as referenced in the main text.

A MathVista Experiment Results

The following graph comes from the MATHVISTA paper website at https://mathvista.github.io/.

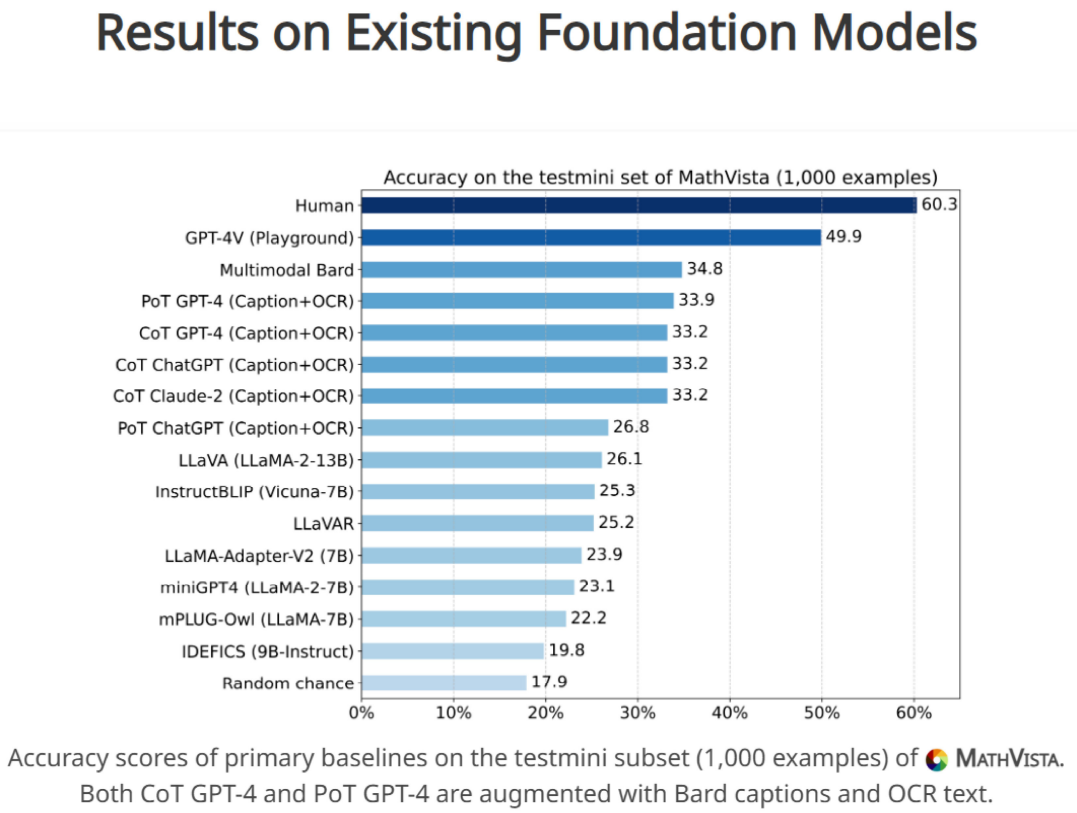


Figure 3: Foundation model performance on MathVista visual reasoning tasks.

B Confusion Matrix

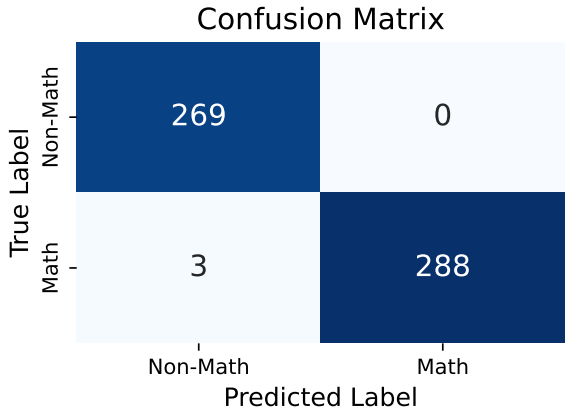


Figure 4: Confusion matrix for the classification task. The model correctly classifies most instances, with only a small number of false negatives, indicating effective performance in distinguishing mathematical from non-mathematical concepts.

C Zero Shot Predictions

Bigram	Sentence	Predicted Class	Probability
linear function	A linear function represents a straight line on a Cartesian plane and has a constant rate of change.	1 (Math)	92.6%
probability distribution	The shape of a probability distribution affects how likely specific outcomes are.	1 (Math)	81.6%
school project	She worked late on her school project about environmental science.	**1 (Math)***	63.1%
integral calculus	Integral calculus deals with accumulation and the calculation of areas under curves.	1 (Math)	76.8%
homework folder	He forgot his homework folder on the bus.	0 (Non-Math)	49.3%

Table 3: Zero-shot predictions on bigrams not seen during training. *Incorrect prediction — *school project* is not a mathematical concept.

D Model Interpetation

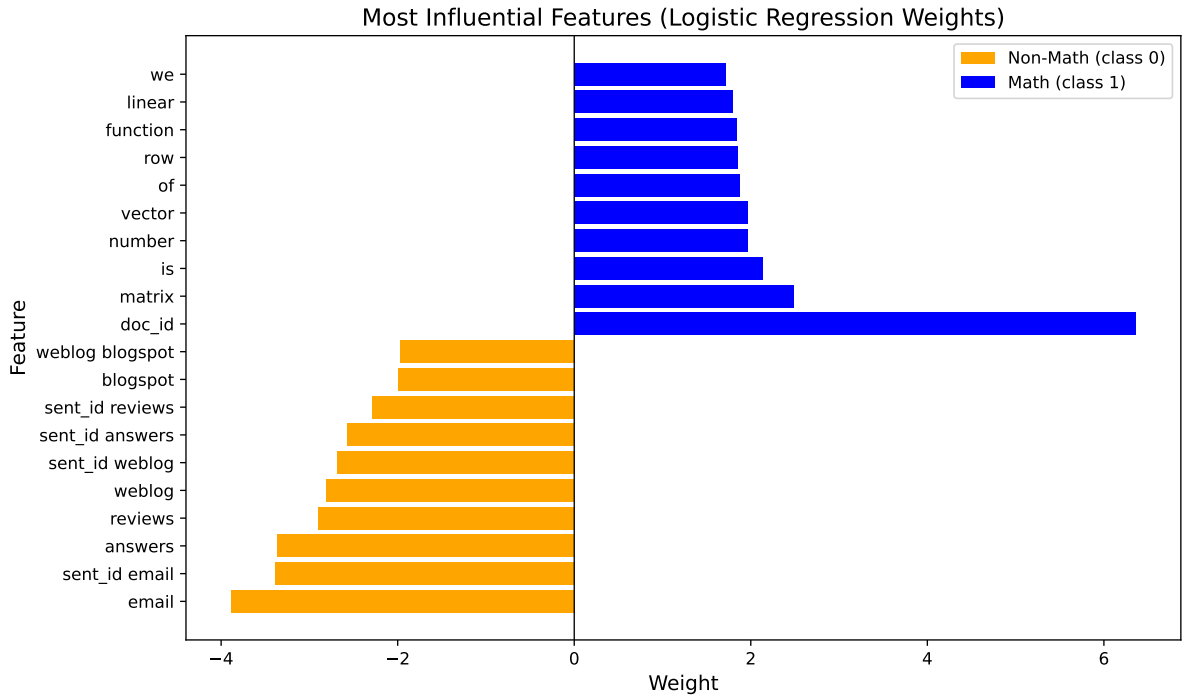


Figure 5: Top weighted features from the logistic regression model. Positive weights (blue) indicate strong association with mathematical concepts, while negative weights (orange) are associated with non-mathematical content. Some features may reflect structural tokens (e.g., *doc_id*, *email*) from the dataset.

Author Index