UFAL-CUNI at SemEval-2026 Task 11: An Efficient Modular Neuro-symbolic Method for Syllogistic Reasoning

Ivan Kartáč **Kristýna Onderková** **Jan Bronec**
Zdeněk Kasner **Mateusz Lango** **Ondřej Dušek**
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics, Charles University
{kartac,onderkova,bronec,kasner,lango,odusek}@ufal.mff.cuni.cz

Abstract

This paper describes our system submitted to SemEval-2026 Task 11: Disentangling Content and Formal Reasoning in Large Language Models. We present an efficient modular neuro-symbolic approach, combining a symbolic prover with small reasoning LLMs (4B parameters). The system consists of an LLM-based parser that translates natural language syllogisms to a first-order logic (FOL) representation, an automated theorem prover, and two optional modules: machine translation for multilingual inputs and a symbolic retrieval component for the identification of relevant premises. The system achieves competitive accuracy and relatively low content effect on most subtasks. Our ablations show that this approach outperforms LLM-based zero-shot baselines in this parameter size range, but also reveal limited multilingual capabilities of small LLMs. Finally, we include a discussion of the task's main ranking metric and analyze its limitations.[1]

1 Introduction

Large Language Models (LLMs) have demonstrated strong performance on a number of reasoning tasks, including syllogistic reasoning. However, LLMs suffer from various reasoning biases (Eisape et al., 2024; Lampinen et al., 2024), particularly the *content effect* (Evans et al., 1983), where the decision about the validity of a given argument is influenced by its compatibility with world knowledge. The SemEval-2026 Task 11 (Valentino et al., 2026b) on multilingual syllogistic reasoning addresses this issue, with the goals of building systems with content-independent reasoning and advancing our understanding of the content effect.

In our SemEval-2026 Task 11 submission, we explore how small LLMs can be used for syllogistic reasoning and first-order logic (FOL) formalization and made robust to content effects. We propose an efficient neuro-symbolic approach, combining small reasoning LLMs (4B parameters) with a symbolic prover. Specifically, we first translate each natural language proposition into a FOL representation and then use an automated theorem prover to determine whether the conclusion is valid. Optionally, we use an additional LLM as a machine translation component for multilingual inputs, while the identification of relevant premises is addressed symbolically by the FOL prover.

Unlike existing neuro-symbolic approaches, we utilize a logic representation well-represented in the training data of the LLMs, rather than instructing them to provide formalizations directly in the target syntax. Specifically, we instruct the LLMs to first formalize the input in LaTeX notation, followed by a rule-based translation to the target syntax. We show that using this intermediate format leads to more reliable, higher-quality formal representations.

We apply our system to all four subtasks of SemEval-2026 Task 11, achieving an accuracy of around 95% for most of them while keeping the content effect relatively low. Based on our ablation experiments, we find that our method is able to significantly reduce the content effect for LLMs of this size, in contrast to instructing them to directly reason about the validity of the syllogisms. To better understand the failure modes of the system, we present a detailed error analysis.

Finally, we present an empirical and theoretical analysis of the main ranking metric and show that it suffers from limited robustness.

2 Related Work

Content effects in reasoning Biases in human reasoning have been extensively studied in psychology (Tversky and Kahneman, 1974), including content effects in syllogistic reasoning (Evans

[1]The code and data are available at https://github.com/ivankartac/SemEval-2026_task11

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3363–3376

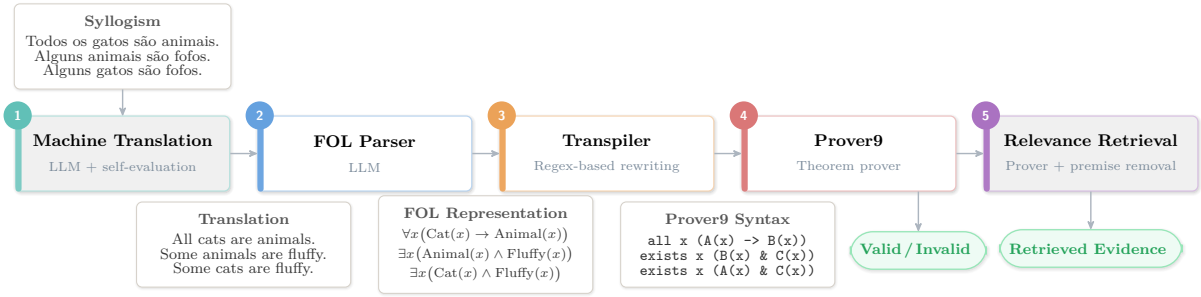

Figure 1: Overview of our system: 1 *Machine Translation* translates multilingual inputs to English (for subtasks 3 and 4); 2 *FOL Parser* translates natural language propositions into FOL formulas in LaTeX format; 3 *Transpiler* rewrites LaTeX formulas to the target Prover9 syntax; 4 *Prover9* determines the validity of the syllogism; 5 *Relevance Retrieval* removes irrelevant premises by using the prover to check which premises do not influence validity (for subtasks 2 and 4).

et al., 1983; Klauer et al., 2000). Recently, reasoning biases similar to those found in humans have been demonstrated for LLMs (Saparov and He, 2023; Eisape et al., 2024; Lampinen et al., 2024). Kim et al. (2025) use circuit discovery to show that while LLMs learn content-independent reasoning mechanisms, these are entangled with world knowledge and therefore prone to content effects. This issue has been addressed through various approaches, such as activation steering (Valentino et al., 2026a), prompting (Xu et al., 2024), supervised fine-tuning (Bertolazzi et al., 2024; Zhou et al., 2025), or hybrid neuro-symbolic methods discussed below.

Neuro-symbolic approaches to logical reasoning Recently, systems combining LLMs and symbolic provers have been proposed to address deductive reasoning. LINC (Olausson et al., 2023) applies an LLM to translate natural language premises into FOL, followed by a symbolic prover to determine the validity of a conclusion. Logic-LM (Pan et al., 2023) and Logic-LM++ (Kirtania et al., 2024) translate the input into different symbolic formulations depending on the problem and apply a corresponding inference tool, such as FOL prover or SAT solver, followed by a refinement step in case the solver fails. Quan et al. (2024) combine LLMs with a theorem prover to automatically verify and improve natural language explanations used to evaluate models in natural language inference (NLI). All these methods translate premises directly into the prover syntax, which can lead to suboptimal results, as we show in Section 6.2.

In contrast to these approaches, LogicGuide (Poesia et al., 2024) integrates a *guide tool* within the LLM's decoding. The guide tool, based on the Peano theorem prover, computes a set of valid continuations in a given step, and the LLM selects one of them through constrained decoding.

3 Task Description

SemEval-2026 Task 11 (Valentino et al., 2026b) investigates how content influences language models when performing formal reasoning. The task evaluates models on the validity of Aristotelian syllogisms (Smith, 2022). Those can align with or oppose common world knowledge, which allows us to measure the content effect on the reasoning (Evans et al., 1983). The task consists of four subtasks: (1) assessing validity of syllogisms, (2) selecting relevant premises for a conclusion, (3) assessing validity in multiple languages, and (4) selecting relevant premises in multiple languages.

Metrics The dataset samples are split into four groups depending on their logical validity and the real-world plausibility of their arguments. The *content effect* (CE) is the average of accuracy differences between valid and invalid samples and between plausible and implausible samples, where each accuracy difference is balanced with respect to the other feature (Valentino et al., 2026b). In addition, standard *validity accuracy* (Acc.) over all samples and the *F1 score on premise relevance* (for subtasks 2 and 4) are measured. Finally, the content effect is used to discount the overall accuracy and obtain the final combined score (CS):

$$\text{CS} = \frac{\text{Acc}}{1 + \ln(1 + \text{CE})} \quad (1)$$

4 System Description

Our system uses a modular neuro-symbolic approach and consists of four main components:

translator (for multilingual subtasks), FOL parser, transpiler, and a FOL prover (see Figure 1).

Translator For multilingual subtasks, we pre-translate the syllogisms from all source languages into English. We prompt an LLM to first provide a translation of a syllogism, followed by a self-evaluation step, and a potential correction step based on the self-evaluation feedback. Prompt templates for each step are shown in Figures 6–8 in the Appendix.

FOL parser We apply a small reasoning LLM to translate natural language syllogisms into FOL representations. To reduce the content effect, individual propositions are formalized one by one, each in a separate inference call. In each subsequent call, the LLM is provided with a mapping from natural language to FOL for the already formalized propositions to ensure consistent predicate names. Since LaTeX notation is very common in training data for similar tasks, we hypothesize that LLMs could provide higher-quality FOL representations when explicitly instructed to generate them in LaTeX format rather than the target prover syntax.

Transpiler To translate the FOL representations to the target prover syntax, we apply a regex-based rewriting with rules to translate the formalized syllogisms from LaTeX to the prover format.

FOL prover To determine the validity of a formalized syllogism, we apply Prover9,[2] an automated theorem prover for first-order logic.

Relevant premise retrieval In subtasks 2 and 4, we apply the following approach to identify relevant premises: We use a greedy algorithm that iterates through premises of a valid syllogism, drops each premise, and applies the prover to check if the proof still holds. If not, the premise is necessary for the conclusion and is labeled as relevant.

5 Experiments

5.1 Data

We use the first 500 training set examples supplied for subtask 1 as validation data and to construct additional synthetic validation sets for the other three sub-tasks. We synthesize multilingual data for subtasks 3 and 4 by translating the original validation set to target languages and sample irrelevant premises for subtasks 2 and 4 from the remaining examples in the training set. Appendix C describes the dataset construction in more detail.

[2] https://www.cs.unm.edu/~mccune/prover9

Subtask	Acc.↑	F1↑	CE↓	CS↑	Rank
1	95.29	–	3.21	39.08	19th of 35
2	97.37	96.84	3.30	39.49	6th of 14
3	93.75	–	6.25	31.45	8th of 13
4	84.90	83.42	1.37	45.20	4th of 15

Table 1: Test set results from the official leaderboard. CS = combined score, Acc. = accuracy, F1 = premise F1, CE = total content effect, Rank = our system's rank in the official leaderboard.

5.2 Models

We use Qwen3 4B Thinking (Yang et al., 2025) as a FOL parser in our main setup and Gemma 3 27B (Team, 2025) as a translator. Since Qwen models occasionally return unparseable output, typically due to output token limits, we implement a retry mechanism with temperature sampling for these cases. Appendix D provides more details on the models and the inference setup.

6 Results

6.1 Overall Scores

The main leaderboard results are presented in Table 1. Despite the small parameter count of the models, our approach achieves competitive accuracy and a reasonably low content effect. The accuracy is preserved even in more complex subtasks, indicating that the approach is robust to irrelevant premises and multilingual variants. We present bootstrapped evaluation results with 95% confidence intervals for both the validation and the test set in Table 4 in the Appendix.

6.2 Ablations

Our ablations are primarily focused on the system components (Table 2), but we also explore different model sizes and variants (Table 6 in the Appendix). Table 5 in the Appendix includes 95% confidence intervals obtained by bootstrap resampling. Prompt templates for all ablations are presented in Figures 9–14 in the Appendix.

Zero-shot end-to-end classification This ablation serves as a simple baseline, where the LLM is instructed to directly predict the validity of a natural language syllogism. For subtasks 2 and 4, we instruct the LLM to also provide indices of relevant premises. Table 2 shows a decrease of around 10

#	Setup	Acc.↑	F1↑	CE↓	CS↑
1	Full	**95.83**	–	2.28	45.61
	- single-step	95.81	–	**2.14**	**46.68**
	- Prover9 format	85.97	–	12.23	24.11
	- LLM prover	79.22	–	20.21	19.58
	End-to-end	84.17	–	14.67	11.25
2	Full	91.17	87.77	5.17	**32.42**
	- single-step	**91.76**	**89.68**	5.60	31.93
	- LLM retrieval	90.40	72.68	**4.76**	30.43
	End-to-end	83.40	64.11	14.92	19.65
3	Full	**94.43**	–	**2.53**	**43.85**
	- MT = Tiny Aya	67.36	–	23.62	8.03
	- MT = Gemma3 4B	87.18	–	5.14	15.96
	- MT = Qwen3 4B	79.75	–	7.09	13.41
	- MT = ∅	90.65	–	5.80	31.84
	End-to-end	83.01	–	14.88	11.06
4	Full	**90.67**	**85.10**	4.83	32.65
	- MT = ∅	85.85	78.14	**3.91**	**33.02**
	End-to-end	81.39	60.72	17.83	18.11

Table 2: Ablations of the pipeline modules (Qwen3 4B Thinking model). # = subtask number. Metrics are explained in Table 1.

points in accuracy, more than 20 points lower F1, and a sharp increase in content effect across all subtasks.

Parsing directly to Prover9 syntax To validate parsing through the intermediate LaTeX format, we design an ablation where the LLM is instructed to parse propositions directly to Prover9 syntax. As the results in Table 2 show, the scores obtained with direct parsing are significantly worse, with a more than 10 point decrease in accuracy and a large increase in content effect. Our manual analysis reveals that 18% formulas contain syntax errors, most of them caused by including invalid LaTeX commands in the Prover9 syntax, or adding extra parentheses.[3]

Single-step parsing We compare the multi-step FOL formalization used in our main setup with a single-step formalization, where the entire syllogism is translated into FOL in a single inference call and generated in JSON format (see Figure 13 in the Appendix). As the results show, both approaches lead to comparable performance.

LLM as a prover This ablation tests the extent to which the LLM is able to replace the prover. Given the premises and the conclusion translated to FOL formulas, the LLM is instructed to decide if the conclusion is valid or not. The prompt template is shown in Figure 14 in the Appendix. The results show a significant decrease of more than 15 points in accuracy, suggesting that LLMs of this size perform well in logical translation but not in reasoning, especially given the content effects.

LLM-based premise retrieval The symbolic approach to relevant premise identification used in subtasks 2 and 4 (see Section 4) is compared with a simple LLM-based approach, where a model is instructed to identify all relevant premises in formalized syllogisms. The results show a significantly worse F1 score compared to the symbolic approach.

Direct multilingual parsing This ablation removes the pre-translation module and applies the FOL parser directly to the original untranslated syllogisms. The results show a substantial decrease in accuracy, which could be explained by the limited multilingual capabilities of models in this size range.

MT models We compare different LLMs applied to the pre-translation step. In addition to Gemma3 27B used in our submission, we test three smaller models: Gemma3 4B, Qwen3 4B Instruct, and Tiny Aya Global (Salamanca et al., 2026), a multilingual LLM with 3.35B parameters. The results show that pre-translating with LLMs in this size range leads to even worse results than skipping the pre-translation step.

Model sizes and variants We compare two model sizes of the Qwen3 family: 4B and 30B-A3B (MoE with 3B active parameters) as well as the Instruct and Thinking variants of both LLMs. Interestingly, our results in Table 6 in the Appendix show that the 4B model achieves significantly higher accuracy than the larger variant, while the Instruct and Thinking variants of the 4B category achieve comparable accuracy.

7 Error Analysis

We analyze the errors of our system on subtask 1, both in the validation set and in the test set released after the competition. Our system made 21 and 10 errors in the validation set and test set, respectively. However, we find that most of them are caused by label errors in the dataset or specifics of the Aristotelian logic (Smith, 2022), while only two and

[3] In fact, syntax errors are even more prevalent in the raw outputs for this setup. However, we apply a regex-based cleanup (e.g. removing characters such as ";") for a fair comparison with our main setup.

four (validation + test) are clear errors. In the following, we describe the identified error categories in more detail.

Label errors Ten of the 21 validation errors were caused by logically incorrect ground truth labels. For example, the following example was marked in the dataset as invalid, despite the conclusion following from the premises by simple transitivity:

> P_1: Anything which is an animal is a flightless thing.
> P_2: Every single bird is an animal.
> C: There are no birds that are not flightless.

The organizers seem to have revised these types of mistakes in the test set, as we did not observe this type of error there. However, the following inconsistencies remained.

Ambiguities between logic types We assume the existential promises of Aristotelian logic, as confirmed by the organizers of the shared task. However, aside from the existential import (see Appendix B), we do not implement any specifics of Aristotelian logic to keep the system more general. We explored Aristotle's procedure for assessing validity with figures and moods using LLMs (Smith, 2022, Chapter 5.4), but it did not meaningfully surpass our main FOL system.

Three samples were not valid syllogisms, as they did not follow the exact structure (Appendix B), although they were logically valid in FOL. For example, "Birds" are mentioned in all premises and in the conclusion, making it an invalid syllogism:

> P_1: There are no birds that can be called fish.
> P_2: It is also true that every bird is a type of animal.
> C: This has led to the conclusion that a portion of animals are birds.

Another one of the samples was invalid as a syllogism, because it used "part of" as a predicate:

> P_1: Every single ocean is a body of water.
> P_2: Any area that is a sea is part of an ocean.
> C: This means that all seas are bodies of water.

The remaining 6 of the 9 ambiguity-based errors involved more complex formulations which were parsed directly into FOL instead of being classified into one of four the categories of Aristotelian moods (Appendix B). Five of these errors were made in syllogisms where a premise starts with "It's not true that any:"

> P_1: It is not true that any flower is a plant.
> P_2: All things that are roses are flowers.
> C: Thus, some roses are not plants.

We understand it as "Not for all," or "There are some X for which it does not hold that...". However, in Aristotelian syllogisms this would be interpreted as "For all X it does not hold that..." (universal negative). Similarly, if "something" is used, it should be interpreted as particular ("Some living organisms are mammals") and not universal ("every") as in FOL:

> P_1: Something that is a living organism is a mammal.
> P_2: The entire set of people is composed of living organisms.
> C: It is the case that all people are mammals.

In the test set, 6 of the 10 errors were based on logic type ambiguities.

Clear errors Overall, we consider only 2 of the 21 errors on the validation set and 4 out of 10 errors in the test set to be clear errors caused by our method. These errors were caused by Qwen 3 producing an incorrect FOL parse. For example, in the following sample, the conclusion was parsed into $\exists x(\text{electronicdevice}(x) \wedge \text{noncomputer}(x))$, when the last predicate should have been $\neg\text{computer}(x)$ to be compatible with the previous premises:

> P_1: A computer is never a tablet.
> P_2: All of the tablets are, without exception, electronic devices.
> C: A subset of electronic devices is not composed of computers.

8 Analysis of the Combined Score

We observe a large CS drop in competition submissions with near-perfect accuracy and a large CS variance, which prompted us to analyze the reliability of the CS metric. Our analysis below shows that the CS metric over-amplifies random artifacts stemming from isolated errors.

Empirical analysis To assess the statistical significance of the content effects measured in the competition, we compare the competition results to those which would result from an unbiased model: one which classifies each of the four groups with the same accuracy $a \in [0, 1]$. While such a model should not exhibit any content effect, as the difference of different group accuracies should be zero, we will show that some will be observed nonetheless. To that end, we simulate predictions for the

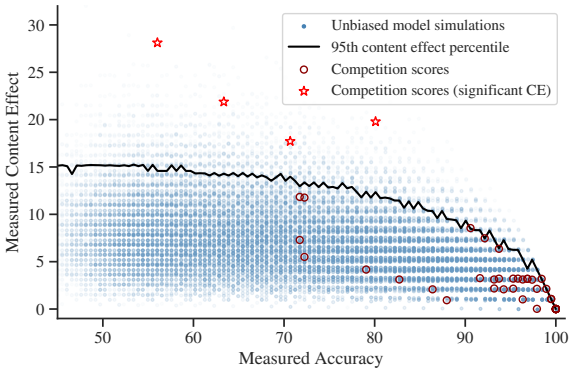

Figure 2: We simulate subtask 1 results of an unbiased model without content effect (blue). We compare these runs against the competition submissions on the same test set (red, dark red). By plotting the empirical 95th percentile of the unbiased model's measured content effect (**black**), we show that only four submissions exhibit a significant content effect (red star).

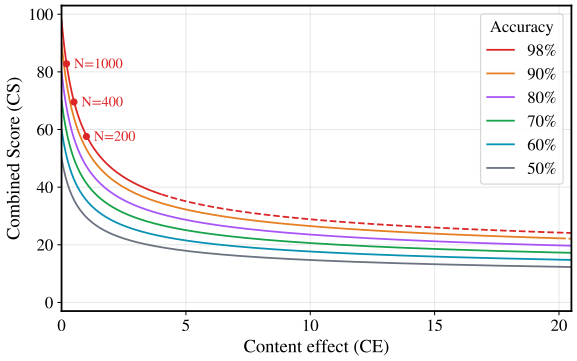

Figure 3: Sensitivity of the combined score (CS) metric with respect to changes in content effect (CE), showing the following properties: (1) CS is most sensitive when CE values are small; (2) systems with higher accuracy are penalized more; (3) flipping a single correct prediction can have a dramatic effect on CS.

subtask 1 test set by choosing a model accuracy $a \sim U(0.5, 1)$ and assigning each sample a correct label with probability a. We plot the measured scores of these simulations in Figure 2. We see that the content effect of the simulated unbiased model not only shows a high variance, but is generally non-zero, with the exception of a perfect model. Since the measured accuracies for each sample group are scaled binomial variables, the high variance in the measured content effects can be simply explained by the small size of the test set (191 samples, roughly 48 per group). We can also see that most subtask 1 submissions match the simulated unbiased results – only four of the submissions show a content effect significantly different from an unbiased model.

Theoretical analysis We derive the expected content effect for an unbiased model, which is non-zero even when the group accuracies are equal:

Proposition 1. *For an unbiased model with accuracy $a \in [0, 1]$, the expected measured content effect on a subtask 1 dataset with N samples in each group is $\mathbb{E}[CE|a] \approx 200\sqrt{\frac{a(1-a)}{\pi N}}$.*

See Appendix A for a proof of Proposition 1. From our experiments, this estimate is already tight for the size of the subtask 1 test set. We should expect a content effect of ≈ 2.3 for a 98%-accurate unbiased model, which already lowers its CS to less than half that of a perfect model. Furthermore, the derivative of the $\mathbb{E}[\text{CE}|a]$ approximation approaches $-\infty$ as accuracy approaches 100%. While this is not accurate for the precise $\mathbb{E}[\text{CE}|a]$ as the Gaussian approximation used in Appendix A is not appropriate for values of a close to 1, it nonetheless provides a reason for the sharp CS drop in task submissions with slightly lower than perfect accuracy.

Figure 3 illustrates the sensitivity of the combined score with respect to the content effect for simulated systems with different accuracies. To demonstrate the concrete impact of this sensitivity, we simulate a high-accuracy system with no content effect and flip a single correct prediction. Even with a sample size $N = 1000$ – substantially larger than the competition's test set – the combined score drops by more than 15 points.

9 Conclusion

This paper presents our system submitted to SemEval-2026 Task 11: Disentangling Content and Formal Reasoning in Large Language Models. We find that while small LLMs (Qwen 3 4B) demonstrate limited capabilities in syllogistic reasoning and high content effect, these models can be useful when integrated with symbolic tools, such as a first-order logic prover. Our experiments show the advantage of parsing syllogisms to LaTeX formulas, a format well-represented in LLMs' training data, rather than using arbitrary parser syntax. We also find that LLMs of this size are not sufficient for multilingual reasoning. Finally, we conduct an analysis of the main ranking metric and show that its sensitivity, combined with a small sample size, can overly penalize systems with high accuracy and low content effect.

Limitations

In our experiments, we did not evaluate the impact of fine-tuning LLMs. As our approach is a pipeline, any errors that occur in earlier stages will propagate through to later stages. The Aristotelian syllogism has a specific structure which may complicate the formal mapping to FOL logic. It is unclear whether the parsing of natural language into FOL creates genuine reasoning with limited expressive power of syllogisms. The generality of the presented approach and its evaluation may be limited: the output of small models may be sensitive to prompt variations, and the data and metrics may not reliably distinguish genuine content effect from statistical noise.

Acknowledgments

This work was was funded by the European Union (ERC, NG-NLG, 101039303). It was additionally supported by the National Recovery Plan funded project MPO 60273/24/21300/21000 CEDMO 2.0 NPO, Charles University Research Centre program No. 24/SSH/009, the project Human-centred AI for a Sustainable and Adaptive Society (reg. no.: CZ.02.01.01/00/23_025/0008691), co-funded by the European Union, and Charles University SVV project number 260 821. It used resources of the LINDAT/CLARIAH-CZ Research Infrastructure (Czech Ministry of Education, Youth, and Sports project No. LM2018101).

References

Leonardo Bertolazzi, Albert Gatt, and Raffaella Bernardi. 2024. A Systematic Analysis of Large Language Models as Soft Reasoners: The Case of Syllogistic Inferences. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL*, pages 13882–13905, USA.

Tiwalayo Eisape, Michael Henry Tessler, Ishita Dasgupta, Fei Sha, Sjoerd van Steenkiste, and Tal Linzen. 2024. A Systematic Comparison of Syllogistic Reasoning in Humans and Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024*, pages 8425–8444, Mexico City, Mexico.

J St BT Evans, Julie L Barston, and Paul Pollard. 1983. On the Conflict Between Logic and Belief in Syllogistic Reasoning. *Memory & cognition*, 11(3):295–306.

Geonhee Kim, Marco Valentino, and André Freitas. 2025. Reasoning Circuits in Language Models: A Mechanistic Interpretation of Syllogistic Inference. In *Findings of the Association for Computational Linguistics, ACL 2025*, Findings of ACL, pages 10074–10095, Vienna, Austria.

Shashank Kirtania, Priyanshu Gupta, and Arjun Radhakirshna. 2024. LOGIC-LM++: Multi-step Refinement for Symbolic Formulations. *CoRR*, abs/2407.02514.

Karl Christoph Klauer, Jochen Musch, and Birgit Naumer. 2000. On Belief Bias in Syllogistic Reasoning. *Psychological review*, 107(4):852.

Andrew K Lampinen, Ishita Dasgupta, Stephanie CY Chan, Hannah R Sheahan, Antonia Creswell, Dharshan Kumaran, James L McClelland, and Felix Hill. 2024. Language Models, Like Humans, Show Content Effects on Reasoning Tasks. *PNAS nexus*, 3(7):pgae233.

Theo Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-order Logic Provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, pages 5153–5176, Singapore.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang 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*, Findings of ACL, pages 3806–3824, Singapore.

Gabriel Poesia, Kanishk Gandhi, Eric Zelikman, and Noah D. Goodman. 2024. Certified Deductive Reasoning with Language Models. *Trans. Mach. Learn. Res.*, 2024.

Xin Quan, Marco Valentino, Louise A. Dennis, and André Freitas. 2024. Verification and Refinement of Natural Language Explanations through LLM-symbolic Theorem Proving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL*, pages 2933–2958, USA.

Alejandro Salamanca, Diana Abagyan, Daniel D'souza, Ammar Khairi, David Mora, Saurabh Dash, Viraat Aryabumi, Sara Rajaee, Mehrnaz Mofakhami, Ananya Sahu, Thomas Euyang, Brittawnya Prince, Madeline Smith, Hangyu Lin, Acyr Locatelli, Sara Hooker, Tom Kocmi, Aidan N. Gomez, Ivan Zhang, and 7 others. 2026. Tiny Aya: Bridging Scale and Multilingual Depth. *CoRR*, abs/2603.11510.

Abulhair Saparov and He He. 2023. Language Models Are Greedy Reasoners: A Systematic Formal Analysis of Chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023*, Kigali, Rwanda.

Robin Smith. 2022. Aristotle's Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*, Winter 2022 edition. Metaphysics Research Lab, Stanford University.

Gemma Team. 2025. Gemma 3 Technical Report. *CoRR*, abs/2503.19786.

Amos Tversky and Daniel Kahneman. 1974. Judgment under Uncertainty: Heuristics and Biases: Biases in judgments reveal some heuristics of thinking under uncertainty. *science*, 185(4157):1124–1131.

Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2026a. Mitigating Content Effects on Reasoning in Language Models Through Fine-grained Activation Steering. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026*, pages 33314–33322, Singapore.

Marco Valentino, Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, and André Freitas. 2026b. SemEval-2026 Task 11: Disentangling Content and Formal Reasoning in Large Language Models. In *Proceedings of the 20th International Workshop on Semantic Evaluation (SemEval-2026)*.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024. Faithful Logical Reasoning via Symbolic Chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024*, pages 13326–13365, Bangkok, Thailand.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 Technical Report. *Preprint*, arXiv:2505.09388.

Yujun Zhou, Jiayi Ye, Zipeng Ling, Yufei Han, Yue Huang, Haomin Zhuang, Zhenwen Liang, Kehan Guo, Taicheng Guo, Xiangqi Wang, and Xiangliang Zhang. 2025. Dissecting Logical Reasoning in LLMs: A Fine-grained Evaluation and Supervision Study. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17075–17098, Suzhou, China.

A Expected Content Effect Analysis

In this section, we analyze the expected value of the measured content effect for an unbiased model. Let $A_g \in [0, 100]$ denote the classification accuracy for each group g. The content effect score CE is defined as:

$$C_{\text{intra}} = \frac{|A_{\text{v,p}} - A_{\text{v},\neg\text{p}}| + |A_{\neg\text{v,p}} - A_{\neg\text{v},\neg\text{p}}|}{2} \quad (2)$$

$$C_{\text{inter}} = \frac{|A_{\text{v,p}} - A_{\neg\text{v,p}}| + |A_{\text{v},\neg\text{p}} - A_{\neg\text{v},\neg\text{p}}|}{2} \quad (3)$$

$$\text{CE} = \frac{1}{2}(C_{\text{intra}} + C_{\text{inter}}) \quad (4)$$

Now, we focus on Proposition 1, which states that for an unbiased model with accuracy $a \in [0, 1]$, the expected measured content effect on a sub-task 1 dataset with N samples in each group is:

$$\mathbb{E}[\text{CE}|a] \approx 200\sqrt{\frac{a(1-a)}{\pi N}} \quad (5)$$

Proof. The sub-task 1 datasets consist of 4 groups, on which the content effect is measured as shown in Equation 4. For each group g, a correct classification can be modeled as a Bernoulli random variable, so the count of correct classifications is distributed binomially: $C_g \sim B(N, a)$. With enough samples, we can model the measured accuracy A_g for that group as normally distributed:

$$A_g = \frac{100C_g}{N} \xrightarrow{\mathcal{L}} \mathcal{N}\left(100a, \frac{100^2 a(1-a)}{N}\right) \quad (6)$$

Since the difference of the accuracies of the two groups would be roughly normally distributed as well, now with zero mean and doubled variance, the absolute value of that difference is distributed by the folded normal distribution. The mean of a folded normal distribution $\mathcal{F}$ is known:

$$\mu_{\mathcal{F}} = \sigma\sqrt{\frac{2}{\pi}}e^{-\frac{\mu}{2\sigma^2}} + \mu\left(1 - 2\Phi\left(-\frac{\mu}{\sigma}\right)\right) \quad (7)$$

In our case $\mu = 0$, and $\sigma^2 = 2\frac{100^2 a(1-a)}{N}$, so the expected value of the absolute difference of accuracies in different groups amounts to:

$$\mathbb{E}[|A_{g_1} - A_{g_2}|] \approx 200\sqrt{\frac{a(1-a)}{\pi N}} \quad (8)$$

Averaging the four absolute differences to obtain CE as in Equation 4 leaves the expected value unchanged. □

B Assumptions of Aristotelian logic

As the shared task is based on Aristotelian logic, it assumes existential import: a universal proposition implies the corresponding particular proposition

(Smith, 2022, Chapter 5.2). For example, "all A are B" implies "some A is B".

However, existential import is not an assumption of formal logic provers like Prover9, which are based on Boolean logic. We address this by synthesizing the required particular propositions through a regex-based extraction of the necessary predicates from the existing propositions.

Valid sylogisms For a syllogism to be in valid form, it must include exactly two premises and a conclusion. Each premise contains a middle term that is not in conclusion. The other term in a premise is a predicate of a conclusion (major premise) or a subject of a conclusion (minor premise) (Smith, 2022, Chapter 5). Subjects can be individual (e.g. Socrates) or universal (e.g. human) and predicates only universal (Smith, 2022, Chapter 4).

Sentences can take one of four forms for universal subjects: universal affirmative ("Every S is P"), universal negative ("No S is P"), particular affirmative ("Some S is P") or particular negative ("Some S is not P"), and two additional forms for individual subjects: "S is P" and "S is not P" (Smith, 2022, Chapter 4.3).

C Validation Data

In this section, we provide details on the construction of our validation data for all four subtasks.

Subtask 1 We use the first 500 training set examples as our validation data for this subtask. This subset also serves as a basis for constructing the validation sets for the other three subtasks.

Subtask 2 We draw irrelevant premises from a pool of the remaining 460 training examples. For each validation example, we select 3–5 irrelevant premises from topically related pool examples: these are identified by shared nouns that appear in at least two propositions of a syllogism, extracted through POS tagging. This selection respects plausibility labels, drawing irrelevant premises only from plausible pool examples for plausible validation examples, while premises for implausible examples may come from any pool example. The selected irrelevant premises are combined with the original premises, de-duplicated, and shuffled.

Subtask 3 The multilingual data for subtask 3 are synthesized by translating the validation set to the languages included in the shared task: English (en), German (de), Spanish (es), French (fr), Italian (it), Dutch (nl), Portuguese (pt), Russian (ru), Chinese (zh), Swahili (sw), Bengali (bn), Telugu (te). For each validation example, we randomly sample a target language and instruct Claude Opus 4.6 to translate the syllogism to the language.

Subtask 4 Since this subtasks includes both multilingual inputs and irrelevant premises, we combine the approaches for subtasks 2 and 3 to synthesize the corresponding validation set. Specifically, we first sample irrelevant premises and then translate this augmented dataset to the target languages.

D Models and Parameters

We run all our experiments with greedy decoding (the temperature parameter is set to 0). The only exception are retries for the FOL parser, which are triggered by unparseable outputs. In these cases, we repeat the response generation with a temperature of 0.6. The context size is set to 16384 tokens. We use the Ollama[4] platform for inference. Table 3 presents all models used in our experiments and the corresponding Ollama tags.

E Prompt Templates

E.1 Submitted System

Prompt templates for the FOL parser (see Section 4) used in our main setup are presented in Figures 4 and 5. Figures 6–8 show the prompt templates for the translation, self-evaluation, and correction steps of the Translator component.

E.2 Ablations

Figures 9–15 show prompt templates for all ablation experiments described in Section 6.2.

F Detailed Results

F.1 Main Results

The results on both the validation and the test set with 95% confidence intervals are presented in Table 4.

F.2 Ablations

Table 6 shows the comparison of the instruct and thinking variants, as well as different model sizes.

[4] https://ollama.com/

Name	Ollama tag
Qwen3 4B thinking	`qwen3:4b-thinking-2507-fp16`
Qwen3 4B instruct	`qwen3:4b-instruct-2507-fp16`
Qwen3 30B-A3B thinking	`qwen3:30b-a3b-thinking-2507-q8_0`
Qwen3 30B-A3B instruct	`qwen3:30b-a3b-instruct-2507-q8_0`
Tiny Aya Global 3.35B	`hf.co/CohereLabs/tiny-aya-global-GGUF:BF16`
Gemma 3 27B	`gemma3:27b-it-fp16`
Gemma 3 4B	`gemma3:4b-it-fp16`

Table 3: List of all LLMs used in our experiments.

Split	Task	Accuracy ↑	F1 Premises ↑	Content Effect ↓	Combined Score ↑
Validation	1	95.83 [94.20, 97.40]	–	2.28 [0.70, 4.38]	45.61 [35.35, 63.04]
	2	91.17 [88.60, 93.60]	87.77 [83.97, 91.33]	5.17 [2.34, 7.98]	32.42 [27.76, 40.52]
	3	90.65 [88.20, 93.21]	–	5.80 [2.43, 9.56]	31.84 [26.52, 40.85]
	4	85.85 [82.80, 88.80]	78.14 [73.25, 82.74]	3.91 [1.21, 7.34]	33.02 [26.09, 45.46]
Test	1	94.77 [91.62, 97.38]	–	5.81 [2.50, 10.87]	33.45 [26.91, 42.99]
	2	97.35 [94.74, 99.47]	96.91 [93.18, 100.00]	3.93 [1.04, 8.17]	39.98 [29.33, 57.84]
	3	93.71 [90.10, 96.88]	–	7.17 [3.33, 11.76]	30.84 [25.44, 39.28]
	4	84.75 [79.67, 90.10]	83.36 [75.98, 90.46]	6.03 [1.75, 12.02]	29.85 [22.98, 42.09]

Table 4: Detailed results on validation and test sets with 95% confidence intervals obtained with bootstrap resampling.

Task	Setup	Accuracy ↑	F1 Premises ↑	Content Effect ↓	Combined Score ↑
1	Full	**95.83** [94.20, 97.40]	–	2.28 [0.70, 4.38]	45.61 [35.35, 63.04]
	- single-step	95.81 [94.00, 97.40]	–	**2.14** [0.57, 4.24]	**46.68** [35.84, 66.39]
	- Prover9 format	85.97 [82.86, 88.72]	–	12.23 [8.99, 16.08]	24.11 [21.77, 26.78]
	- LLM prover	79.22 [75.40, 82.80]	–	20.21 [16.49, 23.87]	19.58 [18.04, 21.27]
	End-to-end	84.17 [80.80, 87.40]	–	14.67 [11.31, 18.31]	11.25 [10.31, 12.35]
2	Full	91.17 [88.60, 93.60]	87.77 [83.97, 91.33]	5.17 [2.34, 7.98]	**32.42** [27.76, 40.52]
	- single-step	**91.76** [89.40, 94.20]	**89.68** [86.19, 92.82]	5.60 [2.84, 8.48]	31.93 [27.75, 39.12]
	- LLM retrieval	90.40 [87.80, 93.00]	72.68 [67.01, 77.89]	**4.76** [1.90, 7.89]	30.43 [25.31, 39.65]
	End-to-end	83.40 [80.00, 86.60]	64.11 [58.40, 69.83]	14.92 [11.45, 18.80]	19.65 [17.54, 22.02]
3	Full	**94.43** [92.40, 96.20]	–	**2.53** [0.67, 5.26]	**43.85** [32.99, 63.61]
	- MT = Tiny Aya Global	67.36 [63.40, 71.21]	–	23.62 [18.90, 28.27]	8.03 [7.35, 8.75]
	- MT = Gemma3 4B	87.18 [84.00, 90.01]	–	5.14 [1.88, 8.72]	15.96 [13.17, 21.34]
	- MT = Qwen3 4B	79.75 [76.35, 83.17]	–	7.09 [2.31, 12.93]	13.41 [10.84, 17.97]
	- MT = ∅	90.65 [88.20, 93.21]	–	5.80 [2.43, 9.56]	31.84 [26.52, 40.85]
	End-to-end	83.01 [79.60, 86.20]	–	14.88 [11.24, 18.81]	11.06 [10.08, 12.19]
4	Full	**90.67** [88.20, 93.00]	**85.10** [80.88, 89.01]	4.83 [2.02, 8.46]	32.65 [27.07, 41.93]
	- MT = ∅	85.85 [82.80, 88.80]	78.14 [73.25, 82.74]	**3.91** [1.21, 7.34]	**33.02** [26.09, 45.46]
	End-to-end	81.39 [78.00, 84.80]	60.72 [54.92, 66.33]	17.83 [13.75, 22.04]	18.11 [16.25, 20.24]

Table 5: Detailed results of the component ablations with including 95% confidence intervals obtained with bootstrap resampling.

Task	Model	Accuracy ↑	F1 Premises ↑	Content Effect ↓	Combined Score ↑
1	Qwen3 4B Thinking	**95.83** [94.2, 97.4]	–	2.28 [0.70, 4.38]	**45.61** [35.35, 63.04]
	Qwen3 4B Instruct	94.83 [92.6, 96.8]	–	**2.27** [0.62, 4.80]	45.60 [34.16, 64.35]
	Qwen3 30B Thinking	91.37 [88.8, 93.8]	–	4.25 [1.56, 7.20]	35.51 [28.92, 46.74]
	Qwen3 30B Instruct	77.79 [74.0, 81.4]	–	21.16 [17.40, 25.41]	19.02 [17.55, 20.58]
2	Qwen3 4B Thinking	**91.17** [88.60, 93.60]	**87.77** [83.97, 91.33]	5.17 [2.34, 7.98]	32.42 [27.76, 40.52]
	Qwen3 4B Instruct	90.65 [88.00, 93.20]	86.37 [82.59, 90.23]	**4.79** [2.09, 8.19]	**32.92** [27.45, 41.89]
3	Qwen3 4B Thinking	**90.65** [88.20, 93.21]	–	**5.80** [2.43, 9.56]	**31.84** [26.52, 40.85]
	Qwen3 4B Instruct	89.02 [86.40, 91.60]	–	6.39 [2.75, 10.65]	30.42 [25.37, 38.38]
4	Qwen3 4B Thinking	**85.85** [82.80, 88.80]	**78.14** [73.25, 82.74]	**3.91** [1.21, 7.34]	**33.02** [26.09, 45.46]
	Qwen3 4B Instruct	84.13 [80.80, 87.20]	72.92 [68.10, 77.92]	4.45 [1.34, 8.24]	30.28 [24.10, 41.77]

Table 6: Comparison of model size and variants with 95% confidence intervals obtained with bootstrap resampling.

```
You are given a proposition in natural language. Your task is to convert the proposition to
first-order logic. For logical operators, use latex symbols: \forall, \exists, \land, \lor,
\neg, \rightarrow. If possible, parse the proposition to a formula consisting of two atomic
formulas, each of them a unary predicate. Each unique predicate should be represented by a
lowercase (or camel-case) word. For a context, you are also given previous propositions
already translated to first-order logic. Make sure you use predicate mapping from previous
responses. Generate the final formula in the boxed format.

Previous propositions:
{previous_propositions}

Proposition: {proposition}
```

Figure 4: Prompt template for the FOL parser. For each natural language proposition, the LLM is instructed to translate it to FOL representation, and is given the mapping from natural language to FOL representation for all previously translated propositions in the syllogism.

```
You are given a proposition in natural language. Your task is to convert the proposition to
first-order logic. For logical operators, use latex symbols: \forall, \exists, \land, \lor,
\neg, \rightarrow. If possible, parse the proposition to a formula consisting of two atomic
formulas, each of them a unary predicate. Each unique predicate should be represented by a
lowercase (or camel-case) word. Generate the final formula in the boxed format.

Proposition: {proposition}
```

Figure 5: Prompt template for the FOL parser used to parse the initial proposition of a syllogism.

```
Translate the following syllogism to English. Translated syllogism will be used as an input
to a FOL parser. Make sure that your translation is unambiguous and easy to parse and
understand. Some propositions may be nonsensical, but you should still preserve their
meaning. Output ONLY the translation, nothing else.

Text to translate:
{syllogism}

English translation:
```

Figure 6: Prompt template for translation used in subtasks 3 and 4.

```
You are a translation quality evaluator. Your task is to verify if the following translation
of a syllogism is correct.

Original text:
{formatted_original}

Translation:
{translation}

Determine whether the translation preserves the meaning of each proposition, and whether
there any mistranslations or omissions.

Provide your verdict as a JSON object with exactly these fields:
- “feedback”: explanation of errors if incorrect, or confirmation that translation is correct
- “correct”: true if the translation is acceptable, false otherwise

Your response MUST end with the JSON object on its own line, formatted as:
{“feedback”: “<your feedback>”, “correct”: <true or false>}
```

Figure 7: Prompt template for translation self-evaluation.

```
Translate the following syllogism to English. Translated syllogism will be used as an input
to a FOL parser. Make sure that your translation is unambiguous and easy to parse and
understand. Some propositions may be nonsensical, but you should still preserve their
meaning. Output ONLY the translation, nothing else.

Text to translate:
{syllogism}

A previous translation attempt was incorrect. Here is the feedback:
{feedback}

Please provide a corrected translation.

English translation:
```

Figure 8: Prompt template for translation with feedback provided by the self-evaluation step.

```
You are a logic expert specializing in formal reasoning and categorical syllogisms. Assess
the logical validity of a syllogism regardless of its real-world plausibility. Carefully
examine the premises and the conclusion to determine whether the conclusion necessarily
follows from the premises using formal logical structure. Identify any fallacies, such as
undistributed middle, invalid contraposition, or missing conclusions. If the conclusion is
logically entailed by the premises, output 'true'; otherwise, output 'false'.

Generate your output as a JSON object with the following fields.
{
  "reasoning": "...",
  "valid": "..." # note: the value you produce must be 'true' or 'false'"
}

[[ ## syllogism ## ]]
{syllogism}

Respond with a JSON object in the following order of fields: 'reasoning', then 'valid' (must
be formatted as a valid JSON bool).
```

Figure 9: Prompt template for the zero-shot end-to-end baseline.

```
You are a logic expert specializing in formal reasoning and categorical syllogisms. Assess
the logical validity of a syllogism regardless of its real-world plausibility. Carefully
examine the premises and the conclusion to determine whether the conclusion necessarily
follows from the premises using formal logical structure. Identify any fallacies, such as
undistributed middle, invalid contraposition, or missing conclusions. If the conclusion is
logically entailed by the premises, output 'true'; otherwise, output 'false'.

If the syllogism is valid, also identify which premises are relevant (i.e., necessary) for
the conclusion to follow. List them as a 0-indexed array of premise indices.

Generate your output as a JSON object with the following fields.
{
  "reasoning": "...",
  "valid": "..." # note: the value you produce must be 'true' or 'false'"
  "relevant_premises": [0, 1, ...] # only if valid is 'true'; 0-indexed premise indices
}

[[ ## syllogism ## ]]
{syllogism}

Respond with a JSON object in the following order of fields: 'reasoning', then 'valid' (must
be formatted as a valid JSON bool), then 'relevant_premises' (only if valid is 'true').
```

Figure 10: Prompt template for the zero-shot end-to-end baseline with relevant premise identification.

```
You are given a proposition in natural language. Your task is to convert the proposition to
first-order logic. Express the statement in Prover9 syntax (an automated theorem prover
library). Below you have a list of the most common logical operations, their symbol in
Prover9 and an example of a formula.

 Operation                  Prover9   Example
 negation                   -         (-p)
 disjunction                |         (p | q | r)
 conjunction                &         (p & q & r)
 implication                ->        (p -> q)
 backward implication       <-        (p <- q)
 equivalence                <->       (p <-> q)
 universal quantification   all       (all x all y P(x,y))
 existential quantification exists    (exists x exists y P(x,y))
 combinations of the above  ...       (exists x (P(x)) & exists x (P(x) & Q(x)))

If possible, parse the proposition to a formula consisting of two atomic formulas, each of
them a unary predicate. Each predicate should be represented by a single uppercase
alphabetic symbol. Note that single-letter lowercase names: x y z u v w p q r must be
individual variables in Prover9. So, "p(x)" is incorrect because "p" is an illegal predicate
name, but "P(x)" is correct. For a context, you are also given previous propositions already
translated to first-order logic. Make sure you use predicate mapping from previous responses.
Generate the final formula in the boxed format.

Previous propositions:

{previous_propositions}

Proposition: {proposition}
```

Figure 11: Prompt template for the ablation where the FOL parser parses propositions directly to the Prover9 syntax.

```
You are given a proposition in natural language. Your task is to convert the proposition to
first-order logic. Express the statement in Prover9 syntax (an automated theorem prover
library). Below you have a list of the most common logical operations, their symbol in
Prover9 and an example of a formula.

 Operation                  Prover9   Example
 negation                   -         (-p)
 disjunction                |         (p | q | r)
 conjunction                &         (p & q & r)
 implication                ->        (p -> q)
 backward implication       <-        (p <- q)
 equivalence                <->       (p <-> q)
 universal quantification   all       (all x all y P(x,y))
 existential quantification exists    (exists x exists y P(x,y))
 combinations of the above  ...       (exists x (P(x)) & exists x (P(x) & Q(x)))

If possible, parse the proposition to a formula consisting of two atomic formulas, each of
them a unary predicate. Each predicate should be represented by a single uppercase
alphabetic symbol. Note that single-letter lowercase names: x y z u v w p q r must be
individual variables in Prover9. So, "p(x)" is incorrect because "p" is an illegal predicate
name, but "P(x)" is correct. Generate the final formula in the boxed format.

Proposition: {proposition}
```

Figure 12: Prompt template for the initial step in the ablation where the FOL parser parses propositions directly to the Prover9 syntax.

```
You are given a syllogism in natural language, consisting of {num_premises } premises and a
conclusion. You task is to convert premises and conclusion of the syllogism to first-order
logic. For logical operators, use LaTeX symbols: \forall, \exists, \land, \lor, \neg,
\rightarrow. Make sure to use consistent mapping from natural language to FOL predicates. If
possible, parse each proposition to a formula consisting of two atomic formulas, each of them
a unary predicate. Each unique predicate should be represented by a lowercase (or camel-case)
word. Do not focus on validity of the syllogism, only parse it to FOL. Generate the output
with natural language propositions and first-order logic formulas in the JSON format:
[{"proposition": "...", "fol_formula": "..."}, ...]

Syllogism:
{syllogism}
```

Figure 13: Prompt template for the FOL parser used for single-step parsing. Unlike the multi-step setup, the LLM is instructed to translate the entire syllogism to a FOL representation and generate the output in JSON format.

```
You are given an argument consisting of two premises and a conclusion, expressed in
first-order logic. Your task is to analyze the formulas and decide whether the conclusion
logically follows from the premises.

Do not focus on notational details of the first-order logic representation or on the specific
names of predicates. Focus only on the logical structure and validity of the argument.

Premises:

{premises}

Conclusion:

{conclusion}

Perform analysis and give the final answer in boxed format:
- \boxed{true} if the conclusion is logically valid
- \boxed{false} if it is not
```

Figure 14: Prompt template for the LLM prover.

```
You are given a syllogism that has been determined to be valid. Your task is to identify
which premises are actually relevant (necessary) for the conclusion to follow.

Premises:
{premises}

Conclusion: {conclusion}
Return ONLY a JSON array of 0-based indices of the relevant premises. For example: [0, 2]
```

Figure 15: Prompt template for the LLM-based relevant premise retrieval.