HHU-SyLo at SemEval-2026 Task 11: Logic in the Loop – Hybridizing LLMs and Theorem Provers for Robust Formal Reasoning

Wiebke Petersen **Cherine Jaziri** **Diem-Xuan Tran**
Heinrich-Heine-Universität Düsseldorf, Germany
{wiebke.petersen, cherine.jaziri, diem.tran}@hhu.de

Abstract

We present our system for SemEval-2026 Task 11 on reasoning disentanglement, separating syllogistic validity from semantic plausibility. We compare direct neural inference against two neuro-symbolic pipelines: translation to first-order logic and to syllogistic triples. By offloading inference to symbolic theorem provers, these hybrid models effectively mitigate content bias and improve logical fidelity.

1 Introduction

SemEval-2026 Task 11 (Valentino et al., 2026) evaluates syllogistic reasoning by disentangling logical validity from semantic plausibility. LLMs frequently succumb to the content effect, where world knowledge biases them toward labeling plausible conclusions as valid. Subtask 1 focuses on English validity checking, while Subtask 2 adds noise through irrelevant premise identification. Subtasks 3 and 4 extend these challenges to a multilingual setting (11 languages), further combining validity classification with premise retrieval.

To address this, we compare a neural baseline against two translation pipelines mapping text to first order logic (FOL) or Aristotelian Syllogistic Triples (ASTs). By offloading inference to symbolic solvers, we aim at decoupling formal logic from semantic bias. In Subtask 1, our best model reached 95.29% accuracy, ranking 19th among 45. For Subtasks 2, 3, and 4, it achieved 95.79% (9th/16), 88.02% (12th/15), and 77.08% (11th/15), respectively.[1]

2 Related Work

Belief bias, the cognitive tendency to favor semantic plausibility over logical validity, is a well-documented phenomenon in humans (Evans et al., 1983) that persists in both early and contemporary large language models (Ando et al., 2023; Lampinen et al., 2024; Kim et al., 2025). Mechanistic analyses of models like Llama 3 and Qwen reveal that internal reasoning circuits are often 'contaminated' by attention heads dedicated to commonsense knowledge (Kim et al., 2025). To resolve this, our Translation-Based Inference via First-Order Logic (TBI-FOL) explicitly bypasses these biased circuits. By offloading logical deduction to the Otter theorem prover, we ensure that validity checks remain strictly formal and entirely independent of semantic content.

Recent research has explored various methods for modeling logical validity and syllogistic inference in natural language. Poddar et al. (2025) evaluated 14 different models and found that logical reasoning is not a uniform emergent property, with models often struggling with quantifier interpretation. Furthermore, Hua et al. (2025) observed that reasoning capabilities are often tied to specific linguistic domains rather than generalizable rules. To mitigate these effects, Valentino et al. (2025) proposed fine-grained activation steering to decouple content from logic. While steering shows promise in internal model adjustments, Zong and Lin (2024) argue that the vast variety of syllogistic moods still poses a significant challenge for purely neural architectures. These observations are consistent with our finding that compact models like SmolLM reach peak performance only when utilized as specialized structural parsers within a symbolic framework.

To achieve full disentanglement, neuro-symbolic pipelines have emerged as a robust alternative to purely neural methods. Xu et al. (2024) introduced Symbolic Chain-of-Thought (SymbCoT), proving that translating context into symbolic formats significantly reduces hallucinations. This paradigm is supported by Pan et al. (2023) and the need for faithful reasoning grounded in formal rules (Xu

[1] Our code is available at https://github.com/WiebkePetersen/HHU-SyLo-SemEval2026-Task11.

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 3188–3198

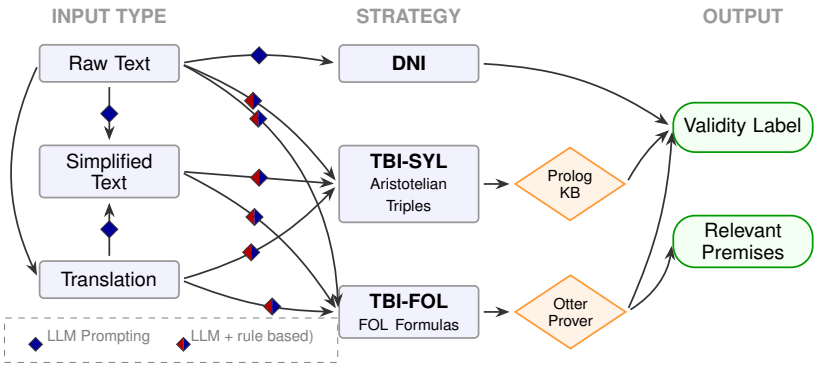

Figure 1: Overview of the modular neuro-symbolic pipeline. Arrows indicate the transition mechanism: purely LLM-based prompting (Blue Diamond) or two alternative transitions via either LLM or rule-based processing (Red and Blue Diamond).

et al., 2024). Our contribution builds on these insights by integrating translations from relatively small LLMs with established symbolic solvers. In contrast to previous work, our pipeline leverages these compact models alongside Otter and Prolog to mitigate content bias. By strictly separating linguistic translation from formal inference, our system seeks to reduce the semantic interference identified in recent literature, providing a scalable solution for multilingual syllogistic reasoning.

3 System Overview

Our system implements a neuro-symbolic pipeline designed to decouple linguistic surface forms from formal logical reasoning. To address the challenge of content bias in syllogistic reasoning, we developed three distinct architectures for assessing argument validity (see Figure 1).

Direct Neural Inference (DNI): A baseline approach where a Large Language Model (LLM) predicts binary validity directly from the natural language prompt without intermediate symbolic steps.

Translation-Based Inference via First-Order Logic (TBI-FOL): In this pipeline, natural language syllogisms are converted into First-Order Logic (FOL) formulas using two distinct translation paradigms: (1) *Neural Translation (Neur-TBI-FOL)*, where an LLM is prompted to generate FOL syntax, and (2) *Rule-based Translation (Rule-TBI-FOL)*, where a deterministic mapping is performed via a specialized Python-based regex engine to extract logical structures from the text.

The actual inference is then performed by the automated theorem prover *Otter*.[2] A syllogism is valid if Otter finds a contradiction between premises and the negated conclusion.

For Subtasks 2 and 4, we perform proof analysis by mapping the clause indices used in Otter's refutation proof back to the original premises to identify the minimal set of relevant premises.

text	It is obvious that not all dogs lay eggs.
simplified	some Dog is not LayEgg
TBI-FOL	exists x (Dog(x) & -LayEgg(x))
TBI-SYL	(o, Dog, LayEgg)

Figure 2: Illustrative example of inputs and outputs generated at each translation step.

Translation-Based Inference via Syllogistic Triples (TBI-SYL): In this approach, the LLM maps the text to Aristotelian triples (q, S, P), where $q \in \{a, e, i, o\}$ represents the categorical proposition type (e.g., *"All S are P"* $\rightarrow$ `('a', S, P)`). Similar to TBI-FOL, this uses neural generation or rule-based extraction. Formal validity is then determined by a Prolog-based Knowledge Base encoding the 256 classical syllogistic moods.

The complete modular architecture and the various experimental paths, i.e. from input selection to final inference, are visualized in Figure 1. This design allows us to compare a wide spectrum of reasoning strategies anchored by two fundamental baselines: (I) A **purely neural path (DNI)**, where the model performs the entire task end-to-end, relying solely on its internal representations; and (II) a **purely symbolic path**, where our deterministic rule-based translation engine is applied directly to the raw text, bypassing LLM-mediated reasoning entirely. Figure 2 provides an illustrative example of the concrete outputs generated at each translation step.

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

The core motivation for employing external engines like *Otter* and the *Prolog KB* was the goal of ensuring that the final validity decision is strictly a function of the extracted logical structure. By offloading the inference to these formal solvers, we aimed to render the reasoning process immune to the semantic plausibility of the premises, in an effort to reduce the impact of content bias that often affects direct LLM predictions.

To further mitigate the impact of complex natural language, we implemented an optional *Linguistic Simplification* step. Here, an LLM simplifies raw text into standardized propositions with normalized predicates (e.g., *"Triangles are shapes with three sides"* $\rightarrow$ `all Triangle is Shape_With_Three_Sides`). The rationale behind this was to decouple the identification of conceptual predicates from the construction of a clean logical formula, thereby reducing the cognitive load on the subsequent translator.

By employing the three distinct target representations (FOL, syllogistic triples, and simplified English) exemplified in Figure 2, we aimed to disentangle logical validity from semantic plausibility by varying how strongly surface linguistic cues are preserved. We expected that the strict formal syntax of FOL would minimize reliance on plausibility heuristics and push the model toward structurally valid reasoning, albeit at the cost of higher susceptibility to syntactic errors that cannot always be repaired by postprocessing. Syllogistic triples (q, S, P) were intended to strike a middle ground: by breaking the natural ordering of quantifier, subject, and predicate, they partially disrupt surface-level plausibility and encourage abstraction to logical roles. As syllogistic triples rarely occur in the training data of LLMs, this representation was expected to further force the model to focus on structural regularities. In contrast, simplified English remained close to natural language and the LLM's training distribution, preserving fluency and plausibility while only lightly constraining structure. Through this spectrum, we aimed to test whether increasing formal rigidity helps isolate logical reasoning from linguistic bias. As shown in Section 5, the FOL-based approach performed best overall, although the differences remained moderate.

To address the multilingual challenges (**Subtasks 3 and 4**), we compared two distinct strategies: (I) a direct application of the pipelines developed for Subtasks 1 and 2 to the native multilingual inputs, and (II) a two-stage approach where the syllogisms were first translated into English using Google Translator via `pypi deep-translator`[3] before applying our reasoning pipelines.

4 Experimental Setup

To evaluate the robustness of our neuro-symbolic pipelines against content bias, we conducted extensive experiments across different tasks, model sizes, and prompting configurations.

4.1 Prompting Strategies

The neuro-symbolic pipeline (*TBI-FOL* and *TBI-SYL*) faces two primary challenges: ensuring *terminological consistency* (mapping semantically equivalent concepts to identical predicates) and adhering to the *restrictive syntax* of the Otter prover and Prolog KB.

To address these, we developed specialized prompt categories for logic translation, syllogistic mapping, and linguistic simplification. These range from minimalistic, example-driven prompts to complex instructions with multilingual trigger word mappings for Subtasks 3 and 4. A baseline *Direct Neural Inference* (DNI) prompt was also used for direct one-step binary validity judgments.

The prompts are supported by curated few-shot example sets, which are balanced across belief-bias conditions (validity vs. plausibility) to guide the models through the logical transformations. Detailed descriptions of all prompt variants and the composition of the few-shot sets are provided in Appendix A.1.

4.2 Models and Parameters

To investigate scaling effects and reasoning capabilities within the category of compact Large Language Models (LLMs), we evaluated our pipeline across a diverse selection of models ranging from 135M to 4B parameters. The selection includes the **SmolLM** series, chosen as a fully open model family (providing transparency regarding training data mixtures and recipes) while being specifically optimized for reasoning tasks at a small scale, the **Qwen** family, which was selected for its established strength in formal logic and its native multilingual capabilities (crucial for Subtasks 3 and 4). Additionally, **Llama** and **Gemma** models serve

[3] pypi deep-translator: https://pypi.org/project/deep-translator/

as industry-standard benchmarks for instruction-following and state-of-the-art reasoning in the 2B to 4B parameter range.

To ensure deterministic and reproducible outputs, we primarily employed greedy decoding (temperature 0.0), with minor model-specific adjustments to ensure generation stability where required. Detailed model identifiers and the complete hyperparameter configuration are provided in Appendix A.2.

4.3 Data Processing and Symbolic Integration

To bridge the gap between natural language and formal logic, we implemented a processing pipeline that handles output extraction, deterministic rule-based translation, and the interface to symbolic solvers. Detailed extraction steps and prover configurations are provided in Appendix A.3.

Normalization and Rule-based Translation LLM output undergoes aggressive artifact stripping and term normalization (e.g. article removal and singularization), to ensure terminological consistency across premises. For the non-neural path, a deterministic engine maps propositions to FOL via copula identification and hierarchical regex patterns. By offloading the formalization, we guarantee valid FOL syntax, effectively mitigating the risk of malformed outputs often produced by LLMs. This allows the system to remain robust whether the LLM is used only for linguistic simplification or is bypassed entirely.

Inference and Existential Import To align FOL with the existential import of classical syllogisms, the pipeline automatically injects existence conditions ($\exists x P(x)$) for all detected predicates. Formulas are further standardized to meet the strict syntactic requirements of the *Otter* theorem prover, including alpha conversion, negation normalization and quantifier scoping. For Subtasks 2 and 4, we utilize the resolution proof history from the prover to identify the minimal sufficient set of premises.

Syllogistic Path (TBI-SYL) The triple-based pipeline maps propositions to (q, S, P) structures, which are validated against a Prolog knowledge base containing the 24 valid Aristotelian moods. Unlike the FOL path, this solver accounts for existential import through its internal encoding.

4.4 Evaluation Metrics

We evaluate our systems using the official metrics defined for the SemEval-2026 Shared Task 11 (Valentino et al., 2026). The evaluation focuses on three dimensions: (1) **Overall Accuracy (Acc)** for validity prediction in Subtask 1, (2) the **Macro-Averaged F1-Score** ($F1_{prem}$) for premise identification in Subtask 2, and (3) the **Total Content Effect (TCE)**, which quantifies belief bias by measuring performance disparities both within the same validity label (plausible vs. implausible cases) and across validity labels. To determine the final ranking, all subtasks compute a **Combined Score** that balances performance with robustness to content bias. In classification settings (Subtasks 1 and 3), accuracy is penalized based on the observed Total Content Effect (TCE). In retrieval-augmented settings (Subtasks 2 and 4), the same penalty is applied to the average of accuracy and premise selection performance ($F1_{prem}$). This ensures that models are rewarded not only for correctness but also for consistency across plausible and implausible cases.

5 Results and Discussion

This section evaluates the performance of our modular architecture across all shared task challenges on the subtasks' test sets provided by the organizers of the shared task. Due to its role as the logical foundation for all subsequent reasoning steps, we first provide an extensive analysis of Subtask 1, followed by an evaluation of the system's performance in premise selection and multilingual settings. For a comprehensive overview of all tested experimental configurations see Appendix B.

5.1 Subtask 1: Logical Validity

The primary objective of Subtask 1 is to establish a robust backbone for validity prediction while minimizing belief bias. We conducted 31 experimental configurations comparing architectures, model families, scaling effects, and prompting strategies. Our analysis identifies **Eng2FOL5** as the superior configuration, utilizing the *SmolLM3-3B* model with an *Ultrashort Prompt* (320 tokens) and eight varied examples. It serves as the benchmark for all subsequent comparisons.

In Table 6 in the appendix, we report the results for all 31 experimental configurations on Subtask 1. Beyond the aggregated metrics discussed in the main text, the table also includes standard classi-

fication outcomes (FP, FN, Errors), as well as a fine-grained breakdown across the four plausibility / validity conditions (Acc_{PV}, Acc_{IV}, Acc_{PI}, Acc_{II}), enabling a detailed analysis of belief bias effects. It further provides full configuration details of the experimental setup, including model choice, prompt design, input format, and properties of the few-shot examples.

Architecture Comparison and Baselines We compare purely neural (DNI) and symbolic (Rule) baselines against various hybrid *Translation-Based Inference* (TBI) approaches. These hybrids differ in their translation mechanism (LLM vs. Rule), the input format (Raw vs. Simplified), and the logical target (FOL vs. SYL). All experiments reported in Table 1 were conducted using the *SmolLM3-3B* model as the primary engine for both direct inference and translation tasks.

Architecture	Acc	TCE	Score
Baselines			
DNI (Direct Neural Inference)	56.54	24.51	13.34
Rule (Rule-TBI-FOL/Syl)	49.74	–	–
Hybrid TBI from raw syllogism			
LLM-TBI-FOL (Eng2FOL)	91.62	4.17	34.68
LLM-TBI-FOL (Eng2FOL5)	**95.29**	**2.15**	**44.37**
LLM-TBI-Syl (Eng2SyllogismProlog)	72.25	18.66	18.16
from simplified syllogism			
Simp(1)-Rule-TBI-FOL	86.39	14.58	23.06
Simp(1)-Rule-TBI-Syl	85.34	18.75	21.43
Simp-LLM-TBI-FOL (SimpEng2FOL1)	85.86	12.50	23.83
Simp-LLM-TBI-Syl (SimpEng2SyllogismProlog)	76.96	14.49	20.58

Table 1: Architecture Comparison: Performance of Neural, Symbolic, and Hybrid Paradigms using *SmolLM3-3B*.

Both baseline approaches performed poorly. The symbolic baseline (*Rule*) achieved an accuracy of only $\approx 50\%$ and failed to find a single valid proof. This stems from the fact that the rule-based grammar was tailored for simplified English; when faced with raw natural language, the extraction of syntactically well-formed formulas fails in 121 out of 191 cases. Similarly, the *Direct Neural Inference* (DNI) baseline, despite a strict prompt emphasizing logical validity over plausibility, achieved only 56% accuracy, with 25 cases being excluded as errors due to non-compliant output formats. The detailed accuracy breakdown in Table 6 further reveals a pronounced plausibility bias in DNI: when distinguishing between plausibility (P vs. I) and validity (V vs. I) conditions, performance drops from $Acc_{PV} = 79.17$ to $Acc_{IV} = 41.67$, while a similar distortion is observed in the invalid cases ($Acc_{II} = 58.33$ vs. $Acc_{PI} = 46.81$).

These results highlight that both purely symbolic and purely neural baselines are strongly affected either by syntactic fragility or by content-driven bias. This suggests that separating linguistic normalization from logical inference may be crucial for robust reasoning. Consistent with this observation, the efficacy of the rule-based system increases dramatically when integrated into a hybrid pipeline. When applied to syllogisms previously simplified by *SmolLM3-3B*, the rule-based translation (*Simp-Rule-TBI*) reaches accuracies of $\approx 86\%$ for both FOL and SYL (Aristotelian Triples) targets.

When delegating the translation entirely to the LLM, a significant performance gap emerges between logical targets. *LLM-TBI-FOL* consistently outperforms *LLM-TBI-Syl*. This advantage likely results from the higher prevalence of FOL-like structures in the LLM's training data. Furthermore, FOL offers higher flexibility, as multiple logically equivalent translations (e.g., varying term orders or conjunctions) can be correctly resolved by the FOL theorem prover, whereas the triple-based SYL format requires an exact match of subject, predicate, and mood. This finding contradicts our initial assumption that disrupting the natural ordering of quantifier, subject, and predicate in syllogistic triples and introducing a less familiar format would help the model abstract away from plausibility biases. Instead, the results suggest that this additional abstraction burden may hinder reliable generation rather than improve logical focus. Due to this robustness and the superior traceability of premises in FOL proofs (crucial for Subtask 2) we focused our subsequent optimization on FOL-based translation.

Against our intuition, the use of a two-stage LLM pipeline (1st simplification, 2nd translation into FOL) did not yield improvements. Comparing *Eng2FOL* (raw) and *SimpEng2Fol1* (simplified) using identical prompts, we observe an accuracy drop and a sharp increase in the *Total Content Effect* (TCE) from 4.17 to 12.5. This suggests that simplification introduces more noise than it removes. Crucially, the 3B-parameter models already produced surprisingly clean FOL syntax from raw text, making an intermediate stage redundant. The increase in TCE is primarily driven by a substan-

tial drop in performance on plausible-valid cases (Acc_{PV}), which decreases from 89.58 to 70.83, as shown in Table 6. Consequently, our best system, *Eng2FOL5*, builds directly on the single-stage-translation *Eng2FOL* approach using optimized prompts and few-shot examples.

Impact of Model Selection and Prompting Strategies Our evaluation across various LLM families and prompting configurations reveals that while smaller models like *SmolLM3-3B* and *Qwen3-4B* can achieve high accuracy ($> 95\%$), the results remain notably unstable. Performance metrics fluctuate significantly depending on the specific few-shot example selection and the alignment between prompt length and model architecture. We observe that no single prompting strategy yields a uniform advantage. Furthermore, a model's peak performance on one subtask or dataset does not consistently translate to others, highlighting a sensitivity to input distributions that complicates the search for a universal "winning" configuration. A comprehensive breakdown of model scaling effects, prompt variations, and full performance tables is provided in Appendix B.

5.2 Subtasks 2–4: Premise Identification and Multilingual Robustness

For **Subtask 2**, premise identification is natively handled by our TBI approach, as the FOL theorem prover (Otter) generates a resolution proof that explicitly lists all utilized premises. The primary challenge lies in the consistent handling of existence clauses introduced during preprocessing.

The best results for Subtask 2 were achieved by *Eng2FOL5Qwen4B*, using the *Qwen3-4B-Instruct* model with the same few-shot configuration as our winning Subtask 1 setup (Acc: 95.79%, $F1_{prem}$: 92.37%, TCE: 3.26, Score": 38.43). This performance underscores the high sensitivity of smaller LLMs to specific input distributions: while *SmolLM3-3B* dominated Subtask 1, it only reached 86.98% accuracy ($F1_{prem}$: 78.38%) on the Subtask 2 test set using the identical configuration (*Eng2FOL5*).

For the multilingual challenges (**Subtasks 3 and 4**), we systematically compared two strategies: (I) direct multilingual processing and (II) automatic translation into English prior to inference.

In **Subtask 3**, strategy (I) performed slightly better: configuration *QwenEng2FOLmultilingual-NewShort* reached 88.02% accuracy, a TCE of 6.38, and an overall score of 29.36. By contrast, strategy (II), using the same underlying *Qwen3-4B-Instruct* model (*QwenEng2FOL6*), delivered comparable performance, with a higher raw accuracy of 90.63% but also an increased TCE of 7.33. A closer examination at the level of individual languages reveals substantial variation in performance: for instance, accuracy ranges from 100% for Dutch to 52% for Swahili in the best-performing configuration. Importantly, these disparities shift depending on the LLM and strategy, indicating substantial model-dependent variation. See Appendix B.3 for details.

In **Subtask 4**, strategy (II), i.e. translation into English combined with the best-performing configuration from Subtask 1 (*Eng2FOL5*), achieved the strongest results (Acc: 77.08%, $F1_{prem}$: 64.15%, TCE: 6.07, Score: 23.89). In contrast, direct multilingual processing performed markedly worse in this setting: the corresponding *Qwen* configuration exhibited a substantially higher content effect (TCE: 14.71). Notably, in 20 out of 192 instances, the LLM failed to generate outputs that could be processed by the theorem prover, indicating robustness limitations in this approach.

6 Conclusion

We presented a modular hybrid neuro-symbolic architecture for reasoning disentanglement. By contrasting pure neural and pure symbolic approaches with hybrid ones, we demonstrated that relatively small specialized LLMs like *SmolLM3-3B* can achieve high logical fidelity when the LLM's linguistic flexibility is anchored by the formal rigor of a symbolic solver.

Our analysis reveals that performance remains highly sensitive to input distributions; small variations in few-shot examples or prompt length can lead to significant fluctuations in logical stability. For the multilingual challenges, the results highlight a trade-off between native processing and machine translation. While automated translation provides a robust baseline, it introduces semantic noise that can skew the *Total Content Effect* (TCE).

Future work should address identified limitations, particularly the alignment between prompt instructions and few-shot examples. Refined rule-based post-processing and more rigorous prompt optimization could further stabilize the translation layer, ensuring that hybrid neuro-symbolic pipelines remain robust across diverse linguistic and logical contexts.

Acknowledgements

This work was developed within the framework of a seminar taught by Wiebke Petersen at Heinrich Heine University Düsseldorf during the winter term 2025/2026, in which student groups independently conducted computational linguistics projects. We would like to thank the participants of the other project groups for their valuable feedback and constructive discussions throughout the course, which helped to refine and sharpen our ideas.

Additionally, we would like to thank the anonymous reviewers for their insightful comments, which helped improve the clarity and quality of this paper. We also gratefully acknowledge the organizers of the shared task for their considerable effort, responsiveness, and flexibility in supporting participants.

References

Risako Ando, Takanobu Morishita, Hirohiko Abe, Koji Mineshima, and Mitsuhiro Okada. 2023. Evaluating large language models with NeuBAROCO: Syllogistic reasoning ability and human-like biases. In *Proceedings of the 4th Natural Logic Meets Machine Learning Workshop*, pages 1–11, Nancy, France. Association for Computational Linguistics.

Jonathan 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.

Wenyue Hua, Kaijie Zhu, Lingyao Li, Lizhou Fan, Mingyu Jin, Shuhang Lin, Haochen Xue, Zelong Li, Jindong Wang, and Yongfeng Zhang. 2025. Disentangling logic: The role of context in large language model reasoning capabilities. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19219–19242, Vienna, Austria. Association for Computational Linguistics.

Geonhee Kim, Marco Valentino, and Andre Freitas. 2025. Reasoning circuits in language models: A mechanistic interpretation of syllogistic inference. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10074–10095, Vienna, Austria. Association for Computational Linguistics.

Andrew K Lampinen, Ishita Dasgupta, Stephanie C Y 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.

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.

Aheli Poddar, Saptarshi Sahoo, and Sujata Ghosh. 2025. Understanding syllogistic reasoning in llms from formal and natural language perspectives. *CoRR*, abs/2512.12620.

Marco Valentino, Geonhee Kim, Dhairya Dalal, Zhixue Zhao, and André Freitas. 2025. Mitigating content effects on reasoning in language models through fine-grained activation steering. *arXiv preprint arXiv:2505.12189*.

Marco Valentino, Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, and André Freitas. 2026. 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)*, pages 13326–13365, Bangkok, Thailand. Association for Computational Linguistics.

Shi Zong and Jimmy Lin. 2024. Categorical syllogisms revisited: A review of the logical reasoning abilities of LLMs for analyzing categorical syllogisms. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 230–239, Miami, FL, USA. Association for Computational Linguistics.

A Experimental setup details

A.1 Detailed Prompting Configurations

This section provides the technical specifications of the prompts and few-shot examples used in our experiments.

A.1.1 Prompts

All full prompt templates are available in our repository: `https://github.com/WiebkePetersen/HHU-SyLo-SemEval2026-Task11/tree/main/prompts`.

- **Logic Translation (TBI-FOL):** We compared three prompt variants: `Eng2FOL` (focusing on explicit syntactic rules, 1721 chars), `Eng2FOLimproved` (emphasizing semantic decoupling, abstract treatment of symbols, and consistent PascalCase naming, 1634 chars), and `Eng2FOLultrashort` (a minimalistic, example-driven approach, 320 chars).

Multilingual TBI-FOL: Two dedicated prompts, `MultiLingFOL` (3033 chars) and `MultiLingEng2FOLultrashort` (502 chars), handled the multilingual scenarios (Subtasks 3 and 4). `MultiLingFOL` includes a step-by-step reasoning instruction (in English) and a multilingual trigger word mapping for 11 languages to ensure logical fidelity during the *Native-test* condition.

- **Syllogistic Mapping (TBI-SYL):** The `SyllTrip` prompt (1254 chars) maps natural language to Aristotelian triples (q, S, P), enforcing naming consistency and mapping quantifiers to categorical proposition types (*a, e, i, o*).

- **Linguistic Simplification:** Two prompts, `LogSimp` (1572 chars) and `LogSimpShortPrompt` (1016 chars), were employed to evaluate the impact of text normalization. These prompts instruct the LLM to extract a standardized "Simplified Logic English" (e.g., `all [Subject] is [Predicate]`, see Figure 2), enforcing singular forms capitalization, atomization for multi-word terms, and the elimination of negative atoms (e.g., converting "is not a fish" to a "no" quantifier and not to "not_fish").

Few-Shot Example Sets

Table 2 details the composition of the curated example sets used to guide the models. It provides details regarding the quantity, average length, and the distribution of validity and plausibility across the example sets.

A.2 Model Specifications and Hyperparameters

The following specific LLM checkpoints were utilized in our experiments:

- **SmolLM-Series:** `SmolLM2-135M-Instruct`, `SmolLM2-1.7B-Instruct`, and `SmolLM3-3B`.

- **Qwen-Series:** `Qwen3-0.6B`, `Qwen3-1.7B`, and `Qwen3-4B-Instruct-2507`.

- **Gemma-Series:** `google/gemma-2-2b-it` and `google/gemma-3-4b-it`.

- **Llama-Series:** `meta-llama/Llama-3.2-3B-Instruct`.

Example Set ID	#	IV	PV	II	PI	∅
LogSimpEx	5	1	4	0	0	3.4
SyllTripEx	4	1	3	0	0	3.8
SimpSyllTripEx	4	1	3	0	0	3.8
Eng2FOLex	3	0	0	1	2	3.0
Eng2FOLexImproved	5	0	0	2	3	3.2
Eng2FOLexImprovedX	8	0	3	2	3	3.1
Eng2FOLexImproved2	8	2	2	2	2	3.6
Eng2FOLexVaried	8	0	5	1	2	3.0
Eng2FOLexLong	4	0	2	2	0	7.5
MultiLingexImproved	22	2	9	1	10	3.0
MultiLingexShort	6	1	2	0	3	3.0

Table 2: Composition of few-shot example sets. # denotes the total number of examples; ∅ represents the average number of logical statements (premises and conclusion) per syllogism. Categories reflect belief-bias conditions: Implausible/Valid (IV), Plausible/Valid (PV), Implausible/Invalid (II), and Plausible/Invalid (PI).

To ensure reproducibility, we set the temperature to 0.0 for the majority of models. For the *Qwen* models, a specific configuration was used to ensure generation stability: `do_sample=True`, `temperature=0.1`, `top_p=0.9`, and a `repetition_penalty` of 1.1. Inference was performed with batch sizes between 8 and 16, depending on the respective model size.[4]

A.3 Implementation Details: Processing and Translation

Extraction and Normalization

Since LLMs often embed formal outputs within conversational text, we employ a multi-step extraction layer involving recursive JSON and literal parsing to isolate logical lists. Artifact stripping (e.g., removing trailing explanations) is applied to ensure clean inputs for the solvers. During normalization, phrases like *"the birds"* and *"a bird"* are both mapped to the unified predicate `Bird`. All terms are capitalized, singularized, and atomized.

Rule-based Translation Engine

The deterministic engine first attempts a heuristic split based on copula identification (*"is"*, *"are not"*). It specifically handles complex linguistic cases like double negations (e.g., *"no X is not Y"*). If a direct split fails, a hierarchical regex safety net identifies the categorical proposition type.

[4] A comprehensive list of all hyperparameters and model-specific settings is provided in the `experiments.yaml` configuration file in our repository.

Symbolic Standardization

Before inference, formulas undergo refinement for *Otter* compatibility:

- **Alpha-conversion:** Ensures unique variable naming across the proof.
- **Syntactic Cleaning:** Parentheses balancing and negation standardization.
- **Syllogistic Existential Closure Assumption:** Injection of existence conditions ($\exists x P(x)$) for all detected predicates.
- **Negated Conclusion:** The conclusion is identified and provided to the prover in its negated form to facilitate a proof by contradiction.

For successful refutations, Otter's proof history is parsed to map used clauses back to their original premises.

B Detailed Results: Models and Prompting

Table 6 shows the results of all our experiments run on the test data of Subtask 1.

B.1 Model Scaling and Family Performance

To evaluate the impact of model capacity under controlled conditions, we compared various LLMs using the prompting configuration (*Ultrashort Prompt*, 8 examples) from our winning experiment (Eng2FOL5). As shown in Table 3, *SmolLM3-3B* achieves the highest combined score, marginally outperforming the *Qwen* models due to its superior bias mitigation (lowest TCE). It also significantly outperforms both *Llama* and *Gemma* in this task. While we observe a general scaling trend where larger models yield better results, the *Qwen3-1.7B* model represents a notable exception, demonstrating remarkably high accuracy for its size. Notably, *Qwen3-4B* achieves perfect accuracy (100%) in both implausible conditions (valid and invalid), indicating a particularly strong ability to focus on logical structure when surface plausibility cues are misleading, as detailed in Table 6.

While Table 3 compares LLMs under identical conditions, Table 4 reports the peak performance for each LLM across all tested prompt configurations. While most architectures benefit from the *Ultrashort* setup, individual tuning can further reduce the TCE for specific families like *Qwen* (e.g., *QwenEng2FOL1*).

Model (Experiment)	Comb	Acc	TCE
SmolLM3-3B (Eng2FOL5)	**44.37**	95.29	**2.15**
Qwen3-4B-Inst. (Eng2FOL5Qwen4B)	36.23	**96.34**	4.26
Qwen3-1.7B (Eng2FOL5Qwen1.7B)	36.03	95.81	4.26
Llama-3.2-3B-Inst. (Eng2FOL3)	24.89	85.34	10.35
Gemma-2-2B-it (Eng2FOL5Gemma2B)	20.08	73.82	13.54
SmolLM2-1.7B-Inst. (Eng2FOL5small)	17.61	65.97	14.58
Qwen3-0.6B (Eng2FOL5Qwen0.6B)	17.53	65.45	14.38
SmolLM2-135M-Inst. (Eng2FOL5tiny)	11.70	50.26	26.04

Table 3: Impact of model size and family on TBI-FOL performance using a fixed configuration (`Eng2FOLUltrashort` prompt, `Eng2FOLexImprovedX` examples; details in Appendix A.1).

Model (Best Experiment)	Comb	Acc	TCE
SmolLM3-3B (Eng2FOL5)	**44.37**	95.29	**2.15**
Qwen3-4B-Instruct-2507 (QwenEng2FOL1)	38.44	93.72	3.21
Qwen3-1.7B (Eng2FOL5Qwen1.7B)	36.03	**95.81**	4.26
Llama-3.2-3B-Instruct (Eng2FOL3)	24.89	85.34	10.35
gemma-2-2b-it (Eng2FOL5Gemma2B)	20.08	73.82	13.54
SmolLM2-1.7B-Instruct (Eng2FOL5small)	17.61	65.97	14.58
Qwen3-0.6B (Eng2FOL5Qwen0.6B)	17.53	65.45	14.38
SmolLM2-135M-Instruct (Eng2FOL5tiny)	11.70	50.26	26.04
gemma-3-4b-it (Eng2FOL11)	10.41	43.46	22.92

Table 4: Peak performance per model family across all experimental configurations. Each entry represents the best-performing individual setup for that specific architecture.

B.2 Prompting Strategies and Few-Shot Effects

The transition between prompt strategies yields no uniform advantage across all architectures. For *SmolLM3-3B*, we observe a slight preference for shorter prompts (see Appendix A.1 for prompt details): comparing `Eng2FOLUltrashort` (Eng2FOL5) to the longer `Eng2FOL` (Eng2FOL7), accuracy marginally increases from 95.29% to 96.34%, yet this is offset by the *Total Content Effect* (TCE) doubling from 2.15 to 4.26 (see Table 6 for full details). Conversely, *Qwen* models exhibit an opposite tendency, showing improved robustness with longer, more descriptive prompts (e.g., `QwenEng2FOL4` vs. `QwenEng2FOL2`).

Variations in the few-shot example sets similarly provide an ambiguous picture (see Table 2 for overview of used few-shot example sets). Analysis of experiments `Eng2FOL5`, `6`, `8`, and `9`—which differ solely in example selection—indicates a subtle trend where a higher quantity of examples improves performance. However, explicitly balancing the plausibility-validity distribution within the examples did not yield the expected gains in bias mitigation.

B.3 Detailed Multilingual Performance Results

The evaluation of the multilingual subtasks revealed a significant variance in performance across different languages and model architectures. While Western European languages like Dutch (nl) consistently achieved high accuracy (up to 100%), results for other languages were markedly inconsistent. Swahili (sw), for instance, proved highly sensitive to model choice, with accuracy fluctuating between 52.94% (best Qwen-configuration) and a mere 23.53% (best SmolLM-configuration).

Table 5 provides a detailed breakdown of the 11 languages, comparing the best-performing configurations for Qwen-4B, SmolLM-3B, and the English-translation pipeline. A notable discrepancy occurs in Chinese (zh-CN): the translation strategy dropped to 77.78% accuracy, whereas direct multilingual processing via SmolLM and Qwen achieved 94.44%. These results suggest divergent training data distributions; while SmolLM remains robust for European datasets, its performance drops sharply for languages like Swahili or Telugu.

Language	**Qwen**	**transl.**	**SmolLM**
sw (Swahili)	52.94	88.24	23.53
pt (Portuguese)	94.12	94.12	88.24
zh-CN (Chinese)	94.44	77.78	94.44
fr (French)	94.44	100.0	100.0
nl (Dutch)	100.0	100.0	94.12
bn (Bengali)	76.47	76.47	64.71
it (Italian)	83.33	94.44	94.44
es (Spanish)	88.89	88.89	100.0
te (Telugu)	88.24	94.12	47.06
de (German)	100.0	94.44	72.22
ru (Russian)	82.35	88.24	64.71
Total Acc.	**88.02**	**90.63**	**79.69**
Errors	**4**	**0**	**9**

Table 5: Accuracies for the best-performing configurations across multilingual and translation-based strategies, broken down by 11 languages: direct multilingual processing with Qwen-4B (*QwenEng2FOLmultilingualNewShort*) and SmolLM-3B (*Eng2FOL5multilingual*), as well as the English-translation pipeline (*QwenEng2FOL6*) using Google Translate.

ExpID	FP	FN	Errors	Acc	TCE	Score	Acc_{PV}	Acc_{IV}	Acc_{PI}	Acc_{II}	Model	Prompt	PromptLen	Input	ExID	#Ex	∅ ExLen
Eng2FOL5	4	5	0	95.29	2.15	44.37	93.75	95.83	93.62	97.92	SmolLM3-3B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
QwenEng2FOL3	5	7	12	93.72	2.08	44.08	91.67	93.75	93.62	95.83	Qwen3-4B	Eng2FOLUltrashort	320	syll	Eng2FOLex	3	3.00
QwenEng2FOL4	3	2	1	97.38	3.19	40.02	95.83	100.00	93.62	100.00	Qwen3-4B	Eng2FOLimproved	1634	syll	Eng2FOLexImproved	5	3.20
Eng2FOL7	4	2	2	96.86	3.19	39.81	95.83	100.00	93.62	97.92	SmolLM3-3B	Eng2FOL	1721	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL5Qwen4B	3	3	1	96.86	3.19	39.81	93.75	100.00	93.62	100.00	Qwen3-4B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL5Qwen1.7B	3	4	1	96.34	3.19	39.59	95.83	95.83	93.62	100.00	Qwen3-1.7B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL6	5	5	1	94.76	3.21	38.86	91.67	97.92	91.49	97.92	SmolLM3-3B	Eng2FOLUltrashort	320	syll	Eng2FOLexImproved	5	3.20
QwenEng2FOL5	3	5	1	95.81	4.17	36.26	91.67	97.92	93.62	100.00	Qwen3-4B	Eng2FOL	1721	syll	Eng2FOLexImproved	5	3.20
Eng2FOL10Qwen4B	3	5	0	95.81	4.17	36.26	91.67	97.92	93.62	100.00	Qwen3-4B	Eng2FOLimproved	1634	syll	Eng2FOLexVaried	8	3.00
QwenEng2FOL2	3	6	5	95.29	4.17	36.06	91.67	95.83	93.62	100.00	Qwen3-4B	Eng2FOLUltrashort	320	syll	Eng2FOLexImproved	5	3.20
QwenEng2FOL1	3	7	3	94.76	4.17	35.87	91.67	93.75	93.62	100.00	Qwen3-4B	Eng2FOLUltrashort	320	syll	Eng2FOLexImproved2	8	3.62
Eng2FOL	5	11	2	91.62	4.17	34.68	89.58	87.50	93.62	95.83	SmolLM3-3B	Eng2FOL	1721	syll	Eng2FOLex	3	3.00
QwenEng2FOL6	3	5	0	95.81	5.21	33.90	89.58	100.00	93.62	100.00	Qwen3-4B	Eng2FOLimproved	1634	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL8	5	12	1	91.10	5.21	32.24	87.50	87.50	91.49	97.92	SmolLM3-3B	Eng2FOLUltrashort	320	syll	Eng2FOLexImproved2	8	3.62
Eng2FOL4	5	11	1	91.62	6.25	30.74	85.42	91.67	91.49	97.92	SmolLM3-3B	Eng2FOLimproved	1634	syll	Eng2FOLexImproved	5	3.20
Eng2FOL9Qwen4B	3	8	12	94.24	7.29	30.25	85.42	97.92	93.62	100.00	Qwen3-4B	Eng2FOLUltrashort	320	syll	Eng2FOLexVaried	8	3.00
Eng2FOL3	9	19	2	85.34	10.35	24.89	72.92	87.50	93.62	87.50	Llama-3.2-3B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL10	4	20	2	87.43	12.50	24.27	72.92	85.42	93.62	97.92	SmolLM3-3B	Eng2FOLimproved	1634	syll	Eng2FOLexVaried	8	3.00
Eng2FOL9	5	21	3	86.39	12.50	23.98	70.83	85.42	93.62	95.83	SmolLM3-3B	Eng2FOLUltrashort	320	syll	Eng2FOLexVaried	8	3.00
SimpEng2FOL1	5	22	2	85.86	12.50	23.83	70.83	83.33	93.62	95.83	SmolLM3-3B	Eng2FOL	1721	Simp1	Eng2FOLex	3	3.00
Simp1-Rule-TBI-FOL	2	24	3	86.39	14.58	23.06	70.83	79.17	95.74	100.00	SmolLM3-3B	LogSimp	1572	syll	LogSimpEx	5	3.40
Simp2-Rule-TBI-FOL	6	24	3	84.29	13.54	22.92	70.83	79.17	89.36	97.92	SmolLM3-3B	LogSimpShortPrompt	1016	syll	LogSimpEx	5	3.40
Eng2FOL2	14	31	5	76.44	9.38	22.89	68.75	66.67	85.11	85.42	Llama-3.2-3B	Eng2FOLimproved	1634	syll	Eng2FOLexImproved	5	3.20
SimpEng2FOL2	6	25	1	83.77	14.58	22.36	68.75	79.17	89.36	97.92	SmolLM3-3B	Eng2FOL	1721	Simp2	Eng2FOLex	3	3.00
Simp1-Rule-TBI-Syl	0	28	0	85.34	18.75	21.43	62.50	79.17	100.00	100.00	SmolLM3-3B	LogSimp	1572	syll	LogSimpEx	5	3.40
SimpEng2SyllogismProlog	9	35	0	76.96	14.49	20.58	64.58	62.50	91.49	89.58	SmolLM3-3B	SyllTrip	1254	Simp1	SimpSyllTripEx	4	3.75
Simp2-Rule-TBI-Syl	4	29	0	82.72	19.79	20.50	58.33	81.25	93.62	97.92	SmolLM3-3B	LogSimpShortPrompt	1016	syll	LogSimpEx	5	3.40
Eng2FOL5Gemma2B	14	33	6	75.39	16.67	19.47	60.42	70.83	76.60	93.75	gemma-2-2b-it	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2SyllogismProlog	13	40	0	72.25	18.66	18.16	54.17	62.50	91.49	81.25	SmolLM3-3B	SyllTrip	1254	syll	SyllTripEx	4	3.75
Eng2FOL5small	23	41	10	66.49	14.58	17.75	62.50	52.08	70.21	81.25	SmolLM2-1.7B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL5Qwen0.6B	17	43	19	68.59	17.58	17.49	52.08	58.33	87.23	77.08	Qwen3-0.6B	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Eng2FOL11	0	69	100	63.87	37.50	13.73	25.00	31.25	100.00	100.00	gemma-3-4b-it	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
DNI	45	38	25	56.54	24.51	13.34	79.17	41.67	46.81	58.33	–	–	–	–	–	–	–
Eng2FOL5tiny	62	29	10	52.36	23.96	12.41	70.83	68.75	46.81	22.92	SmolLM2-135M	Eng2FOLUltrashort	320	syll	Eng2FOLexImprovedX	8	3.12
Rule-TBI-Syl	0	96	126	49.74	50.00	10.09	0.00	0.00	100.00	100.00	–	–	–	–	–	–	–
Rule-TBI-FOL	0	96	121	49.74	50.00	10.09	0.00	0.00	100.00	100.00	–	–	–	–	–	–	–

Table 6: Detailed performance metrics per model and configuration on the test set for Subtask 1. Reported columns include standard classification outcomes (FP, FN, Errors), overall performance (Acc), bias sensitivity (TCE), and the official Combined Score. Subgroup accuracies are shown for the four plausibility/validity conditions (Acc_{PV}, Acc_{IV}, Acc_{PI}, Acc_{II}). The remaining columns describe the experimental setup, including model, prompt configuration, input type, and properties of the few-shot examples used (ExID, number of examples, and average example length).