Thiyaga6851 at SemEval-2026 Task 11: Disentangling Content and Formal Reasoning in Large Language Models using Neuro-Symbolic Mapping

Thiyagarajaa.P.K
Computer Science Engineering
Sri Sivasubramaniya Nadar
College of Engineering

Durairaj Thenmozhi
Computer Science Engineering
Sri Sivasubramaniya Nadar
College of Engineering

Abstract

This paper presents our system for SemEval-2026 Task 11 Subtask 1, which evaluates the formal validity of English syllogisms independently of semantic plausibility. To reduce content effects, we use a hybrid neuro-symbolic pipeline that separates natural-language abstraction from logical inference. The system maps each syllogism into categorical propositions using template rules and a learned parser, followed by explicit role mapping for the major, minor, and middle terms. If the abstraction is structurally complete, an exact Venn-style satisfiability solver checks validity; otherwise, the instance is routed to a learned fallback classifier. Our official submission achieved 71.73% accuracy, a Total Content Effect of 11.84, and a Combined Score of 20.19. Development analysis shows that symbolic inference is reliable on well-formed abstractions, while most remaining errors arise from segmentation, paraphrase, multiword terms, and unstable term alignment.

1 Introduction

Large Language Models (LLMs) often judge arguments using semantic plausibility rather than formal validity. This content effect can cause models to accept invalid but plausible conclusions and reject valid but counter-intuitive ones. SemEval-2026 Task 11 targets this problem by requiring systems to classify English syllogisms according to logical structure rather than surface content. In Subtask 1, systems perform binary validity classification, and Total Content Effect (TCE) measures sensitivity to plausibility cues.

This task is difficult because high accuracy cannot rely only on lexical or commonsense associations. A system must identify the quantifiers in the premises and conclusion, align the relevant terms, and decide whether the conclusion follows from the premises. In our development analysis, the main source of error was not the symbolic solver itself, but the mapping from natural language into a stable three-term syllogistic form.

Our system follows a hybrid neuro-symbolic design. It first normalizes each syllogism and maps its statements into categorical propositions. It then identifies the major, minor, and middle terms through explicit role mapping. When the extracted structure is complete and consistent, the system applies an exact Venn-style satisfiability solver over the three-term representation. When the structure is incomplete or unstable, the instance is handled by a learned fallback classifier instead of being forced into a symbolic decision.

This design balances interpretability and coverage. The symbolic solver gives deterministic validity judgments for clean abstractions, while the fallback classifier handles cases where parsing or role mapping is uncertain. Our official submission achieved 71.73% accuracy, a Total Content Effect of 11.84, and a Combined Score of 20.19. These results show that symbolic checking can reduce content sensitivity when abstraction is reliable, but also that robust segmentation, paraphrase handling, and term alignment remain the key challenges for content-invariant syllogistic reasoning.

2 Background and Related Work

SemEval-2026 Task 11 evaluates whether NLP systems can perform deductive reasoning when formal validity conflicts with semantic plausibility (Valentino et al., 2026). Subtask 1 focuses on binary validity classification for English syllogisms. Systems must decide whether a conclusion follows from two premises under categorical reasoning, without relying on world knowledge or plausibility cues.

2.1 Task Formalization

Each instance contains two premises and one conclusion. The target label indicates whether the con-

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2187–2192

clusion is formally valid. Unlike standard Natural Language Inference benchmarks, this task is designed to separate validity from factual plausibility. A valid syllogism may contain implausible content, while an invalid syllogism may sound plausible. This makes structural reasoning more important than lexical association.

2.2 Evaluation Metrics

The task uses three main metrics. Validity Accuracy (ACC) measures binary classification performance. Total Content Effect (TCE) measures the performance gap between belief-congruent and belief-incongruent cases. The Combined Score penalizes accuracy when TCE is high, encouraging systems that are both accurate and less sensitive to semantic plausibility.

2.3 Content Effects in Syllogistic Reasoning

Content effects occur when systems judge arguments partly from semantic plausibility rather than formal validity. This issue is especially relevant in syllogistic reasoning, where a conclusion can be logically valid but implausible, or invalid but plausible. Bertolazzi et al. (2024) show that LLMs exhibit content effects and heuristic reasoning patterns in syllogistic inference. Eisape et al. (2024) compare syllogistic reasoning in humans and language models and find systematic biases in both. These findings motivate the SemEval-2026 Task 11 setting, where validity must be separated from plausibility.

2.4 Symbolic and Neuro-Symbolic Reasoning

One line of work addresses reasoning errors by translating natural-language problems into symbolic representations and using external solvers. Jiang et al. (2024) formalize logical reasoning problems in Lean and use theorem proving to improve consistency. Related LLM-symbolic verification work has also explored formal tools for checking and refining natural-language reasoning (Quan et al., 2024). Our approach follows this general direction, but uses a compact categorical-logic solver designed for three-term syllogisms rather than a general theorem prover.

2.5 Abstraction-Based Content-Invariant Reasoning

Another line of work reduces content effects by abstracting away lexical semantics. Ranaldi et al. (2025) improve chain-of-thought reasoning through quasi-symbolic abstractions. Maraia et al. (2026) propose abstract activation spaces for content-invariant syllogistic reasoning inside LLMs. Kim et al. (2025) analyze reasoning circuits involved in syllogistic inference. Our system shares the goal of separating structure from content, but performs this separation externally through categorical proposition extraction, explicit role mapping, and symbolic satisfiability checking.

2.6 Methodological Challenges

The main challenge in this task is not the complexity of the logical solver, but the reliability of the abstraction step. A system must identify quantifiers, extract subject and predicate terms, and align the major, minor, and middle terms across the premises and conclusion. Errors in segmentation, paraphrase handling, or term normalization can produce incorrect formal structures even when the symbolic inference module is correct. This motivates our hybrid design, which combines learned parsing, explicit role mapping, symbolic checking, and fallback routing.

3 System Design

Our system follows a hybrid neuro-symbolic design that separates natural-language abstraction from validity checking. The goal is not to solve syllogisms directly from raw text, but to first convert each instance into a stable categorical form. Symbolic inference is applied only when this abstraction is complete and consistent. Otherwise, the instance is routed to a learned fallback classifier.

3.1 Structural Normalization and Preprocessing

The first stage reduces surface variation while preserving the logical content of each statement. We segment the syllogism into two premises and one conclusion using discourse markers such as *Hence* and *It follows that*. We also remove common lead-in phrases such as *It is true that*, while preserving negation scope in cases such as *It is not true that all*. Finally, we canonicalize copular expressions such as *belong to*, *classified as*, and *are* to improve statement-level parsing.

3.2 Hybrid Template and T5 Semantic Parsing

We implement semantic parsing using two complementary parsers. The first is a deterministic tem-

plate parser that captures high-confidence categorical forms, including universal affirmative, universal negative, particular affirmative, and particular negative statements. Since many official examples contain paraphrases outside these templates, we also train a T5-small sequence-to-sequence parser to map each normalized statement into a categorical DSL of the form:

$$(Q, S, P)$$

At inference time, the template parser is preferred when it returns a valid parse. Otherwise, the T5 parser is used as a rescue parser with self-consistency sampling. Parser confidence is computed as the proportion of sampled T5 outputs that agree on the selected quantifier.

The T5 component uses the `t5-small` model with approximately 60.5M parameters. It is trained for 3 epochs on 56,587 statement-level training pairs. The input is a normalized single statement, and the output is the corresponding categorical DSL string.

3.3 Categorical Representation

Each parsed statement is represented as a triple (q, S, P), where $q \in \{\text{ALL, NO, SOME, SOME\_NOT}\}$. This representation removes lexical meaning from the reasoning step and keeps only the quantifier, subject term, and predicate term.

- **A:** Universal affirmative, $\forall x(S(x) \rightarrow P(x))$
- **E:** Universal negative, $\forall x(S(x) \rightarrow \neg P(x))$
- **I:** Particular affirmative, $\exists x(S(x) \wedge P(x))$
- **O:** Particular negative, $\exists x(S(x) \wedge \neg P(x))$

3.4 Syllogistic Role Mapping

After statement-level parsing, the system aligns the terms into syllogistic roles. The minor term and major term are taken from the conclusion, while the middle term is identified from the overlap between the two premises:

$$M = (\{s_1, p_1\} \cap \{s_2, p_2\}) \setminus \{s_3, p_3\}$$

The mapping must produce exactly three stable terms. If the terms cannot be aligned cleanly, the instance is not sent to the symbolic solver. This prevents incorrect formal structures from being treated as valid logical inputs.

3.5 Symbolic Satisfiability Engine

The symbolic solver is intentionally simple because Subtask 1 involves three-term categorical syllogisms. We use a transparent Venn-region satisfiability checker over the eight possible intersections of the three terms. The solver is not intended as a contribution in theorem proving. Its role is to provide a deterministic validity check once the natural-language input has been mapped into categorical form.

Validity is tested through unsatisfiability. A conclusion is valid if the premises together with the negation of the conclusion have no satisfying assignment. Universal statements impose emptiness constraints, while particular statements require at least one compatible region to be non-empty. Since there are only 2^8 possible region assignments, exhaustive enumeration is sufficient and fully deterministic.

3.6 Existential Import Semantics

We include existential import as a semantic variant. Under modern logic, universal statements can be true even when the subject class is empty. Under Aristotelian interpretation, universal statements often imply that the subject class exists. Our system evaluates this setting by adding non-emptiness constraints for the subject of universal premises. This allows the solver to match the task's implicit logical conventions when existential assumptions are required.

3.7 Fallback Classifier

If segmentation, parsing, role mapping, confidence gating, or existential-import agreement fails, the system does not force an invalid label. Instead, it routes the instance to a fallback classifier trained on abstracted syllogistic inputs. This fallback uses a DistilRoBERTa-base sequence classifier and is trained on official training examples together with synthetic and abstracted syllogistic forms. The fallback avoids treating parser failure as logical invalidity, which could otherwise distort the evaluation of reasoning performance.

3.8 Routing and Gating

The final prediction is selected using a routing policy. Symbolic inference is used only when the parsed structure passes all consistency checks. Otherwise, the fallback classifier is used.

Condition	Action
Segmentation fails	Fallback classifier
Unknown quantifier	Fallback classifier
Role mapping fails	Fallback classifier
Parser confidence below threshold	Fallback classifier
Modern and EI solvers disagree	Fallback classifier
All checks pass	Symbolic solver

Table 1: Routing policy used to select between symbolic inference and fallback classification.

4 Experimental Setup

This section describes the data, implementation details, model settings, and evaluation metrics used in our system.

4.1 Data and Evaluation Splits

Subtask 1 provides English syllogisms annotated with a validity label and a plausibility attribute. The validity label is the main prediction target, while plausibility is used only for bias analysis and TCE computation.

The official training set contains 960 examples, and the blind test set contains 191 examples. For development, we created a stratified internal split from the training data, producing 816 training examples and 144 development examples. The valid and invalid classes were balanced in the development split, with a valid ratio of 0.500 and an invalid ratio of 0.500.

The development split was used to tune existential import, parser confidence thresholds, routing decisions, and fallback behavior. The neural components were trained using the available training data before official test submission, while the symbolic solver remained deterministic.

4.2 Implementation and Environment

The system was implemented in Python[1]. The pipeline contains normalization, statement parsing, role mapping, symbolic inference, and fallback classification modules.

The symbolic solver is a Venn-style satisfiability checker over the eight possible regions of a three-term universe. Since only $2^8 = 256$ region assignments are possible, exhaustive enumeration is sufficient and deterministic. The solver is CPU-based and computationally small. Neural components were trained and evaluated on a single NVIDIA T4 GPU. A complete development cycle, including parser evaluation, fallback training, and configuration testing, required approximately 75 minutes.

[1]Python 3.10.12, PyTorch 2.1.0

4.3 Parser and Fallback Model Settings

The semantic parser combines a deterministic template parser with a learned T5 parser. The T5 component uses `t5-small`, which has approximately 60.5M parameters. It was trained for 3 epochs on 56,587 statement-level pairs. The input is a normalized single statement, and the output is a categorical DSL string of the form:

$$(Q, S, P)$$

The fallback classifier uses `distilroberta-base` with a binary classification head. Inputs are either the lowercased syllogism or an abstracted skeleton such as:

```
ALL(T1,T2) || SOME(T2,T3) -> SOME(T1,T3)
```

It was trained for 3 epochs with learning rate $2e^{-5}$, batch size 16, evaluation batch size 32, maximum sequence length 256, and seed 42. The fallback training set contained 10,150 examples: 150 derived from official training data and 10,000 derived from synthetic syllogisms. A stratified 85/15 split produced 8,627 fallback training examples and 1,523 fallback development examples.

4.4 Configuration and Metrics

The main configuration choices were existential import, parser confidence, normalization depth, and routing policy. Existential import controls whether universal premises imply non-empty subject classes. Parser confidence controls whether a parsed structure is trusted for symbolic inference. Normalization depth controls lead-in removal, copula canonicalization, and surface-form cleanup.

System performance is measured using Validity Accuracy (ACC), Total Content Effect (TCE), and the Combined Score:

$$S = \frac{ACC}{1 + \ln(1 + TCE)} \tag{1}$$

We report ACC as a percentage and TCE in percentage points. Internally, metric computation uses fractions. For example, a development TCE of 0.0818 corresponds to 8.18 percentage points. On the internal development split, the final configuration obtained 0.6736 ACC, 0.0818 TCE, and a rank score of 0.6245.

5 Results and Discussion

This section reports the official test results and summarizes the main development findings on routing,

parser coverage, role mapping, and representation errors.

5.1 Official Evaluation Results

Our system, submitted under participant ID `thiyaga6851`, was evaluated on the blind test set for SemEval-2026 Task 11, Subtask 1. The official performance metrics are summarized in Table 2.

Metric	Score
Validity Accuracy (ACC)	71.73%
Total Content Effect (TCE)	11.84
Combined Score	20.19
Rank	**41st**

Table 2: Official evaluation results for Subtask 1.

The results show that accuracy alone is not sufficient for this task. Although the system reached 71.73% accuracy, the final score was reduced by a TCE of 11.84, confirming that robustness to plausibility effects is central to the evaluation.

5.2 Development Experiments

We evaluated several configurations on the internal development split. Table 3 reports ACC and TCE, with TCE shown in percentage points.

Configuration	ACC	TCE
Template-only symbolic	68.2	15.1
Template + EI symbolic	69.5	14.8
Full router with fallback	67.36	8.18

Table 3: Development results on the internal split.

The full router reduced TCE compared with the template-only symbolic variants, although its accuracy was lower on the development split. This reflects the main tradeoff in our system: routing uncertain cases to the fallback classifier can reduce content sensitivity, but it does not guarantee higher raw accuracy. The fallback classifier is not used as an independent baseline; it is part of the final pipeline and is called only when parsing, role mapping, confidence checks, or EI agreement fail.

5.3 Parser Coverage and Routing

On the 144-example internal development split, the deterministic template parser alone covered only 14.58% of examples. The T5 parser increased coverage to 97.92%, and the combined parser returned a parse for all examples. However, successful statement-level parsing did not always lead to successful syllogistic reasoning. Only 34 examples were routed to the symbolic solver, while 110 examples were routed to the fallback classifier, mainly because role mapping did not produce a clean three-term structure.

Parser or router outcome	Value
Template parser coverage	14.58%
T5 parser coverage	97.92%
Combined parser coverage	100.00%
Symbolic route	34
Fallback route	110
Parser failure	0
Mapping failure	110

Table 4: Parser coverage and routing statistics on the internal development split.

These results show that the main bottleneck is not local quantifier parsing alone, but global alignment of subject, predicate, and middle terms across the full syllogism. Confidence analysis also showed that several incorrect predictions occurred at parser confidence values near 0.9 to 1.0, meaning local parser confidence was not sufficient for final correctness.

5.4 Effect of Existential Import and Ablations

Existential import improved some cases involving universal premises and existential conclusions, but its effect was not uniform. For this reason, agreement between modern and existential-import interpretations was used as one routing signal. If the two interpretations disagreed, the instance was routed to the fallback classifier.

Ablation results showed that role mapping was the most important component. Removing syllogistic role mapping reduced accuracy by 3.5 points, while removing lead-in stripping, copula normalization, and existential import reduced accuracy by 1.2, 0.8, and 0.5 points respectively. These results confirm that global term alignment contributes more than the symbolic solver itself to final performance.

5.5 Error Analysis and Observations

Manual inspection shows that most errors are representation failures rather than failures of the Venn solver. Frequent issues include hypernym mismatches such as *vehicle* versus *type of vehicle*, multiword term mismatches such as *knowledgeable person* versus *person*, plural normalization errors such as *ferraries* versus *ferrari*, and non-canonical expressions such as *mutually exclusive categories*.

These observations support the view that the main bottleneck is global syllogistic abstraction. The symbolic solver is reliable when quantifiers

and terms are correctly mapped, but high parser confidence alone does not guarantee correct final reasoning.

6 Conclusion and Future Work

This paper presented a hybrid neuro-symbolic system for SemEval-2026 Task 11 Subtask 1. The system separates natural-language abstraction from validity checking by combining template and T5 parsing, explicit syllogistic role mapping, Venn-style satisfiability checking, and a DistilRoBERTa fallback classifier.

Our official submission achieved 71.73% accuracy, a Total Content Effect of 11.84, and a Combined Score of 20.19. These results show that symbolic checking is useful when the input is reduced to a stable three-term categorical form. However, development analysis shows that the main difficulty is robust abstraction, especially segmentation, paraphrase handling, term normalization, and global role alignment.

The main limitation of the system is that local statement parsing is not sufficient for reliable syllogistic reasoning. High parser confidence can still lead to incorrect predictions when the extracted terms do not align across the premises and conclusion. The symbolic solver is therefore reliable only when the upstream representation is correct.

In future work, we plan to improve term canonicalization, add agreement-based role validation, and explore constrained neural parsers for more robust content-invariant syllogistic reasoning. We also plan to extend the approach to multilingual syllogistic reasoning, where surface variation and term alignment may be even more challenging.

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*, pages 13882–13905.

Tiwalayo Eisape, Michael 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*, pages 8425–8444.

Dongwei Jiang, Marcio Fonseca, and Shay B. Cohen. 2024. Leanreasoner: Boosting complex logical reasoning with lean. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 7497–7510.

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.

Gabriele Maraia, Marco Valentino, Fabio Massimo Zanzotto, and Leonardo Ranaldi. 2026. Abstract activation spaces for content-invariant reasoning in large language models. *Preprint*, arXiv:2602.02462.

Xin Quan, Marco Valentino, Louise A. Dennis, and Andre 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*, pages 2933–2958.

Leonardo Ranaldi, Marco Valentino, and Andre Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17222–17240.

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)*.