lakshadvani at SemEval-2026 Task 11: A Neuro-Symbolic Approach to Content-Independent Syllogistic Reasoning

Laksh Advani
Independent Researcher
laksh.advani@gmail.com

Abstract

We describe our system for SemEval-2026 Task 11 on disentangling content from formal reasoning. The content effect in syllogistic reasoning, where models judge validity based on conclusion plausibility rather than logical structure, persists even with explicit instructions to ignore real-world knowledge. We find that this bias is better addressed structurally than through prompting: by restricting the LLM to a translation role (mapping natural language to abstract variables) and delegating all deductive reasoning to a deterministic checker over the 24 valid Aristotelian forms, we eliminate content bias entirely on Subtask 1 (100.0 combined, TCE=0.0, 4th place). Our Subtask 2 system, which lacks this separation, scores 41.08 (7th place) despite 95.26% accuracy and 99.47% premise retrieval F1, because a TCE of 2.94 incurs a 58% penalty. A three-way ablation on training data using GPT-5 confirms the pattern: a vanilla LLM achieves 78% accuracy with TCE=19; adding Aristotelian rules to the prompt reaches 90%/TCE=5; offloading to the symbolic checker reaches 97%/TCE=3.

1 Introduction

SemEval-2026 Task 11(Valentino et al., 2026) asks participants to determine the formal validity of syllogistic arguments, independent of whether their conclusions happen to be true in the real world. The difficulty lies in the *content effect*: both humans and LLMs tend to accept logically invalid arguments when the conclusion sounds plausible, and reject valid ones when it sounds absurd (Evans et al., 1983; Dasgupta et al., 2024). The task's scoring metric penalizes this behavior through a Total Content Effect (TCE) term that can reduce an otherwise strong system's score dramatically.

We approached this by separating translation from reasoning. Rather than prompting the LLM to judge validity directly (which we found consistently triggered content-biased responses), we ask it only to extract the logical structure: identify the three terms and rewrite each sentence in standard quantified form using abstract variables. A simple Python script then checks whether the resulting mood-figure combination is one of the 24 classically valid syllogistic forms. Since this checker operates over single-letter variables, it has no access to what the syllogism is about, and therefore cannot exhibit content bias.

On the test set, this approach achieved 100% accuracy with zero bias on Subtask 1 (4th of 5 teams, all at 100). For Subtask 2, where the input includes distractor sentences, we used a different architecture that kept all reasoning within the LLM, and this version scored 41.08 due to a TCE of 2.94, despite 95.26% accuracy and 99.47% premise retrieval F1. The gap between the two systems provides a natural ablation for the value of symbolic validation.

2 Background

2.1 Task Setup

The task provides syllogistic arguments with labels for both *validity* (logical structure) and *plausibility* (real-world truth of the conclusion). The training set contains 960 English syllogisms, roughly balanced across four conditions (Table 1).

	Plausible	**Implausible**	**Total**
Valid	240	240	480
Invalid	234	246	480
Total	474	486	960

Table 1: Training data distribution.

An example of the challenge: *"Not all canines are aquatic creatures known as fish. It is certain that no fish belong to the class of mammals. Therefore, every canine falls under the category of mammals"* is labeled invalid despite having a true con-

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 561–566

clusion. A biased system would mark this as valid because canines really are mammals.

We participated in Subtask 1 (binary classification of validity in English) and Subtask 2 (identify relevant premises from a larger set, then classify validity).

2.2 Evaluation

The ranking metric applies a logarithmic penalty for content bias:

$$\text{Score} = \frac{\text{Perf}}{1 + \ln(1 + \text{TCE})} \tag{1}$$

where Perf is accuracy (Subtask 1) or the average of accuracy and premise retrieval F1 (Subtask 2), and TCE measures the average accuracy disparity across the four validity×plausibility conditions. The penalty is steep: a TCE of 2.94 costs about 58% of the raw score.

2.3 Related Work

Content effects, specifically belief bias, have been documented in humans for decades (Evans et al., 1983) and recently confirmed in LLMs (Dasgupta et al., 2024; Eisape et al., 2024). While some research explores mitigating these effects through activation steering (Valentino et al., 2025) or quasi-symbolic chain-of-thought (Ranaldi et al., 2025), our approach focuses on structural separation. Similar neuro-symbolic strategies like Logic-LM (Pan et al., 2023) and LINC (Olausson et al., 2023) use general-purpose solvers for first-order logic. By contrast, our system targets the finite space of the 24 valid Aristotelian forms, reducing the reasoning stage to a deterministic set-membership check.

3 System Overview

We built two systems: a neuro-symbolic pipeline for Subtask 1, and an LLM-only pipeline for Subtask 2 (Figure 1).

3.1 Subtask 1: Translation + Symbolic Checking

Translation. We prompt the LLM to extract three terms (Subject S, Predicate P, and Middle Term M) and rewrite each sentence using variables A, B, C in one of four standard forms:

Type	Standard Form
A	All X are Y
E	No X are Y
I	Some X are Y
O	Some X are not Y

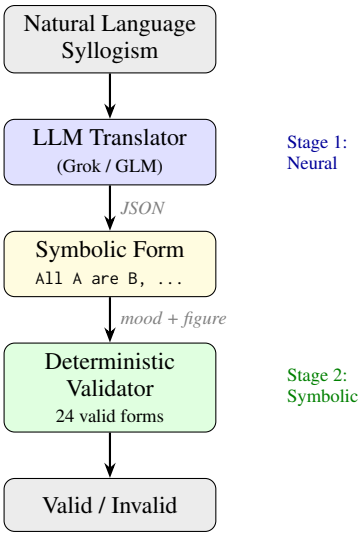

Figure 1: Subtask 1 pipeline. The LLM translates; the checker reasons. Since the checker only sees abstract variables (A, B, C), it cannot be influenced by semantic content.

The prompt includes a few examples of tricky phrasings ("Not a single X is Y" → E-type; "It is not the case that all X are Y" → O-type), since the dataset uses varied paraphrases of these quantifiers. The LLM returns a JSON object with `p1_mapped`, `p2_mapped`, and `conc_mapped`. We used Grok-4.1-fast via OpenRouter for this step.

Symbolic validation. Given the translated output, a Python function does three things:

1. **Mood extraction**: determines A/E/I/O for each statement by checking whether it starts with "all", "no", or "some" (with a "not" check for O vs I).

2. **Figure determination**: identifies which premise is major (contains P) and minor (contains S), then determines the figure (1–4) based on where M appears:

Fig.	Major	Minor
1	M–P	S–M
2	P–M	S–M
3	M–P	M–S
4	P–M	M–S

3. **Lookup**: checks whether the mood-figure string (e.g., "AAA-1") is in the set of 24 valid Aristotelian forms (Appendix A).

If any step fails (unrecognized mood, wrong number of variables, etc.), the system defaults to

Input: *"All mammals are animals. All dogs are animals. Therefore, all dogs are mammals."*

Gold: invalid, plausible

Translation: S=dogs, P=mammals, M=animals
`p1: All C are B  p2: All A are B  conc: All A are C`

Validation: Moods A,A,A. Major=[C,B], Minor=[A,B]. M at position 1 in both → Figure 2. Form: AAA-2. AAA-2 $\notin$ valid forms → **invalid**

A vanilla LLM would likely say "valid" because dogs really are mammals.

Figure 2: Worked example showing how the pipeline catches an Undistributed Middle fallacy despite a plausible conclusion.

"invalid." This is deliberate: false negatives are preferable to false positives, since false positives in the invalid+plausible quadrant directly increase TCE.

Figure 2 shows a worked example.

3.2 Subtask 2: LLM-Only Reasoning

Subtask 2 inputs contain 3–8 sentences, most of which are irrelevant distractors. The system needs to both select relevant premises and judge validity.

We took a different approach here: we pre-indexed sentences numerically and gave GLM-4.7 a system prompt asking it to identify the conclusion, extract terms, select premise indices, determine the Aristotelian form, and judge validity, all in one pass. The prompt explicitly encoded the valid forms and instructed the model to ignore real-world plausibility.

This design was partly pragmatic. Our Subtask 1 checker assumes exactly two premises and a conclusion, but Subtask 2 inputs have variable length with distractors. We could have had the LLM select premises first and then run the symbolic checker, but did not implement this during the competition. In hindsight, this was a mistake; the lack of a separate validation stage is directly responsible for the higher content bias in our Subtask 2 results.

On failures (API errors, unparseable JSON), the system defaults to `{validity: false, relevant_premises: []}`.

3.3 Development History

We went through three iterations on Subtask 2:

1. **Direct classification**: asked the LLM "is this valid?" Score: 29.20.

2. **Prompt engineering**: added instructions like "if the conclusion is factually true but doesn't follow logically, return false." This helped somewhat but TCE stayed high.

3. **Logic Compiler prompt**: encoded Aristotelian rules explicitly in the system prompt. Score improved to 41.08.

The Subtask 1 system (with the symbolic checker) was developed in parallel and achieved 100.0 on the first submission. The fact that we could not easily port the symbolic checker to Subtask 2 (due to the variable-length distractor problem) was the main practical limitation.

3.4 Model Choices

We used Grok-4.1-fast (via OpenRouter) for Subtask 1 and GLM-4.7 for Subtask 2, both with temperature 0. The choice was driven by practical experimentation rather than systematic comparison; we had limited API budget. The symbolic checker itself is model-agnostic, as we verified this post-hoc using GPT-5 (Section 5.3).

4 Experimental Setup

Our system is entirely zero-shot, using the 960 labeled training examples provided by the organizers only for post-hoc analysis. We used Grok-4.1-fast (via OpenRouter) for Subtask 1 and GLM-4.7 for Subtask 2, both with temperature 0. Models were accessed via API with JSON-mode output, and calls were parallelized using Python's `ThreadPoolExecutor` (10–20 workers) with exponential backoff for rate limits. Total processing time for each subtask was under 30 minutes. For our ablation studies (Section 5.3), we utilized GPT-5 via Azure OpenAI to verify the generalizability of our architectural findings.

5 Results

5.1 Competition Results

Table 2 summarizes our official scores.

Subtask	Rank	Score	Acc	TCE
1 (Classification)	4	100.0	100.0	0.0
2 (Retr.+Class.)	7	41.08	95.26	2.94

Table 2: Official results. Subtask 2 also achieved F1=99.47 for premise retrieval.

All five Subtask 1 teams scored 100/100. On Subtask 2 (Table 3), the picture is more varied. Our premise retrieval is strong (99.47 F1, second-best),

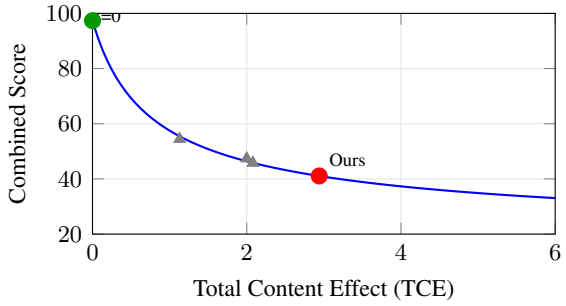

Figure 3: Combined score as a function of TCE, assuming Avg=97.37. Gray triangles: teams 4–6.

but the bias penalty dominates. The top three teams achieved TCE=0.0; teams 4–7, ourselves included, all have TCE > 1 and score below 55.

#	Team	Score	Acc	F1	TCE
1	Habib_TAZ	100.0	100.0	100.0	0.0
2	YNU-HPCC	100.0	100.0	100.0	0.0
3	zhyuxie	99.47	100.0	98.95	0.0
4	junhaofu	54.43	96.84	94.21	1.13
5	joanitolopo	47.40	98.95	100.0	2.0
6	butasrafael	45.80	97.89	96.84	2.08
7	**Ours**	**41.08**	**95.26**	**99.47**	**2.94**

Table 3: Subtask 2 leaderboard.

5.2 The TCE Penalty

Figure 3 shows the severity of the penalty. With our raw Avg(Acc, F1) of 97.37, a TCE of 0 would yield a score of 97.37; instead, TCE=2.94 drops it to 41.08. The gap between teams 3 and 4 on the leaderboard is almost entirely explained by TCE.

5.3 Three-Way Ablation

To understand how much each component contributes, we ran three configurations on the 960-item training set using GPT-5 (Table 4). The competition itself used Grok and GLM, but the architectural comparison holds across models.

Configuration	Acc	TCE	Comb.
Vanilla ("is this valid?")	78.1%	19.14	66.48
+ Logic Compiler prompt	89.9%	4.95	85.76
+ Symbolic checker	96.6%	2.63	94.12

Table 4: Three-way ablation on training data using GPT-5.

The vanilla baseline gets 78% accuracy with massive content bias (TCE=19). Adding the Logic Compiler prompt (which encodes Aristotelian rules but still lets the LLM do all the reasoning) brings accuracy up to 90% and cuts TCE to 5. Offloading the validity check to the symbolic engine adds another 7 points of accuracy and halves TCE again. On the competition test set with Grok, the neuro-symbolic configuration reached 100%/0.0.

5.4 Per-Quadrant Breakdown

Table 5 shows where the accuracy differences come from.

	V+P	V+I	Inv+P	Inv+I
GPT-5 on training data:				
Vanilla	59.6	59.6	97.9	95.5
Logic Compiler	86.7	85.8	95.7	91.5
Neuro-Symbolic	98.8	97.5	96.6	93.5
Competition (test data):				
Neuro-Sym. (Grok)	100.0	100.0	100.0	100.0
LLM-Only (GLM)	95.8	97.9	95.6	92.0

Table 5: Accuracy (%) by condition. V=Valid, Inv=Invalid, P=Plausible, I=Implausible.

The vanilla LLM gets only about 60% on valid syllogisms but 96% on invalid ones; it defaults to "invalid" when uncertain. This is correct for half the data but produces high TCE. The Logic Compiler narrows this gap but doesn't close it. The symbolic checker makes performance nearly uniform.

An unexpected finding in Table 5: the worst quadrant for the LLM-only competition system is invalid+implausible (92.0%), not invalid+plausible as classical belief bias would predict (Evans et al., 1983). In the human literature, belief bias primarily manifests as accepting invalid arguments with believable conclusions. Our LLM system instead struggles most with invalid arguments whose conclusions are *unbelievable*. One possible explanation is that semantically incoherent premises (e.g., "every bird is a rock") disrupt the model's ability to parse the logical structure, independent of any plausibility judgment on the conclusion. This may reflect a different mechanism than human belief bias, though the aggregate effect on TCE is similar.

5.5 Error Analysis

Our Subtask 2 system made 9 errors on 190 test items (Table 6).

	Pred Valid	Pred Invalid
True Valid	92	3
True Invalid	6	89

Table 6: Confusion matrix, Subtask 2 test set (n=190).

We inspected all 9 errors. Two were content bias errors where invalid+plausible syllogisms were accepted as valid. One example:

> *"It is true that all consultants are professionals. Every teacher is a professional. [. . .] Everything that is a lawyer is a professional. There exist lawyers who are not doctors."*
> **Gold:** invalid, plausible. **Predicted:** valid.

The conclusion ("some lawyers are not doctors") is factually true, but the premises only establish that various professions are professionals; nothing connects lawyers to doctors. The system appears to have been swayed by the plausible conclusion.

Four errors involved invalid+implausible syllogisms with semantically incoherent premises (e.g., "every bird is a rock"), where the LLM seemed unable to parse the logical structure when the content was nonsensical. One error was a pure reasoning failure: the system identified the correct premises (indices [0, 5], matching ground truth) but still returned an incorrect validity judgment. The remaining two were cases where the system assigned premise indices to invalid syllogisms that have no relevant premises in the ground truth, suggesting it tried to impose structure where none exists.

Of the 9 errors, 8 involve invalid syllogisms (which have empty ground-truth premise sets), and the system predicted premise indices for all 8. This over-interpretation of invalid arguments appears to be a systematic failure mode of the LLM-only approach.

6 Conclusion

Content bias in syllogistic reasoning can be prevented structurally. Prompt-based mitigation ("ignore real-world truth") improved our Subtask 2 score from 29 to 41 but did not eliminate the bias. Removing the LLM from the reasoning step entirely, by delegating validity checking to a deterministic engine over abstract symbols, reduced TCE to zero on Subtask 1. The top three teams on Subtask 2 also achieved TCE=0.0, suggesting that content-independent validation (whether symbolic or otherwise) may be necessary rather than optional for this class of tasks.

The most immediate improvement to our system would be a two-stage Subtask 2 pipeline: use the LLM for premise retrieval (where it already achieves 99.47% F1) followed by the symbolic checker for validation. We did not implement this during the competition due to the variable-length input issue, which in retrospect was a solvable engineering problem rather than a fundamental limitation. Majority voting over multiple translation attempts and extension to the multilingual subtasks (3–4) are also natural next steps; the symbolic validation logic is language-independent by construction.

More broadly, our results suggest that LLMs' difficulty with formal reasoning may stem less from a lack of logical "knowledge" and more from the entanglement of that knowledge with distributional semantics. The same model that fails to reason correctly over natural language premises succeeds when its role is restricted to structured extraction. This points toward a general design principle for tasks requiring formal guarantees: use neural models for what they do well (language understanding), and keep the reasoning in a system that can be verified.

References

Ishita Dasgupta, Andrew K Lampinen, 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.

Tiwalayo Eisape, MH 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 NAACL*.

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.

Theo X 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 EMNLP*.

Liangming Pan, Alon Alber, Wenhu Cai, and Jamie Callan. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of EMNLP*.

Leonardo Ranaldi, Marco Valentino, and André Freitas. 2025. Improving chain-of-thought reasoning via quasi-symbolic abstractions. In *Proceedings of ACL*, pages 17222–17240.

Marco Valentino, Giwon Kim, Dhruv Dalal, Zhenyun 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)*.

A Valid Aristotelian Syllogistic Forms

Figure 1	**Figure 2**	**Figure 3**	**Figure 4**
AAA (Barbara)	EAE (Cesare)	IAI (Disamis)	AEE (Camenes)
EAE (Celarent)	AEE (Camestres)	AII (Datisi)	IAI (Dimaris)
AII (Darii)	EIO (Festino)	OAO (Bocardo)	EIO (Fresison)
EIO (Ferio)	AOO (Baroco)	EIO (Ferison)	AAI (Bamalip)
AAI (Barbari)	AEO (Camestros)	AAI (Darapti)	AEO (Calemos)
EAO (Celaront)	EAO (Cesaro)	EAO (Felapton)	EAO (Fesapo)

Table 7: The 24 valid Aristotelian syllogistic forms.

B Prompt Templates

Subtask 1 (translation).

```
Translate this syllogism to standard form using variables A, B, C. S (Subject of Conclusion) = A, P (Predicate of Conclusion) = C, M (Middle Term) = B. Standard forms: All X are Y / No X are Y / Some X are Y / Some X are not Y.

Examples:
1. “Not a single bird is a cat.” -> “No A are C”
2. “It is not the case that all tigers are lions.” -> “Some A are not C”
3. “Every single car is a vehicle.” -> “All A are C”

Syllogism: "{text}"
Return ONLY JSON: {"p1_mapped": "...", "p2_mapped": "...", "conc_mapped": "..."}
```

Subtask 2 (system prompt).

```
You are a Formal Logic Compiler. PROTOCOL: 1) Find conclusion. 2) Extract S, P, M. 3) Select premise indices. 4) Map to standard form. 5) Determine mood + figure. 6) Check against 24 valid forms. If the conclusion is factually true but doesn't follow logically, return false. OUTPUT: {"validity": bool, "relevant_premises": [int, int], "formal_mood": "AAA-1"}
```