GIL-Zaragoza at SemEval-2026 Task 11: Comparing Classification, Autoformalization, and Ontologies for Formal Reasoning Capabilities

Francisco F. López-Ponce[1], **Lucía Pitarch**[2], **J. Iván Saavedra-Martínez**[1], **Ignacio Huitzil**[2], **Sergio-Luis Ojeda-Trueba**[1], **Fernando Bobillo**[2], **Gemma Bel-Enguix**[1]
[1] Universidad Nacional Autónoma de México
{francisco.lopez.ponce,ivan.saavedra}@ciencias.unam.mx
{gbele,sojedat}@iingen.unam.mx
[2] University of Zaragoza, Zaragoza, Spain
Aragon Institute of Engineering Research (I3A), Zaragoza, Spain
{lpitarch,ihuitzil,fbobillo}@unizar.es

Abstract

This paper describes our participation in Task 11 of SemEval-2026, which evaluates the ability of models to determine logical validity of syllogisms independent of real-world content. We develop and compare three approaches for Subtask 1: (1) an encoder-based classification baseline using both classical ML methods and fine-tuned BERT with debiasing strategies; (2) an autoformalization pipeline combining DPO-aligned models with first order logic translation and formal inference via Prover9; and (3) a hybrid neuro-symbolic approach using GPT to generate OWL 2 ontologies evaluated with the HermiT reasoner. Our best result was achieved by the encoder-based classifier, obtaining a 72.25% accuracy and a combined score of 20.37, placing 40th out of 45 participating teams. Analysis shows that classification methods exhibit lower content bias, autoformalization approaches suffer from translation inconsistencies and syntax incompatibilities, and ontology-based reasoning is hindered by prompt design limitations and verbose serialization formats. All our code can be found in the paper's repository[1].

1 Introduction

As Large Language Models (LLMs) continue to increase in size and capabilities (DeepSeek-AI et al., 2025; Singh et al., 2025), reasoning has become a highly debated research topic (Yue et al., 2025). This has led to the development of shared tasks such as Task 11 of SemEval-2026: Disentangling Content and Formal Reasoning in Language Models (Valentino et al., 2026). The objective of this task lies in testing LLMs in content-independent reasoning in order to verify whether a model is capable of maintaining a consistent form of reasoning regardless of an argument's real-world validity. This task gives an LLM a set of syllogisms (either aligned or unaligned with real-world knowledge) from which the model should be able to determine only the logical validity of the syllogisms. Task results are validated using a weighted coefficient between Accuracy and Total Content Effect (model's susceptibility to content bias).

The task is comprised of 4 different subtasks, each with underlying modifications regarding language and amount of irrelevant syllogisms added. For this report, we focus on the results obtained in subtask 1, where all syllogisms are in English, and there are no irrelevant premises added. Initially we generate a baseline using an encoder-based classification pipeline, followed by two autoformalization models based on LLMs (one based on first order logic translation and formal inference, while the other one based on using the LLMs to translate into formal logic language and a later reasoner for inference). The best performing solution obtained a 72.25 Accuracy score and a 11.77 TCE, leading to a final score of 20.37 and placing 40th out of 45 participants.

Our performance, while underwhelming, sheds light into the inner mechanics of autoformalization and abstract reasoning using LLMs and external solvers.

2 Related Work

LLM reasoning has been a highly researched topic since the inception of these models. Prompting strategies such as Chain-of-Thought (Wei et al., 2022) or Tree-of-Thought (Yao et al., 2023) increase a model's performance in general reasoning tasks such as mini crosswords and creative writing. Similarly, mathematical reasoning has been a popular evaluation setting for LLMs in recent times. Math-centered benchmarks such as MATH

[1] https://github.com/Kurocaguama/SemEval-2026

Proceedings of the The 20th International Workshop on Semantic Evaluation (2026), pages 2438–2446

(Hendrycks et al., 2021), AIME (Balunović et al., 2025), and GSM8K (Cobbe et al., 2021), are often a test bed for state-of-the-art LLMs with models such as DEEPSEEK-R1 (DeepSeek-AI, 2025), GPT-OSS (OpenAI, 2025), and PHI-4 (Abdin et al., 2024) reaching near perfect scores.

However, these benchmarks do not cover the full range of formal reasoning since they are limited to non-formal reasoning problems, certain mathematical topics (Logic not being one of them), and overlook the validity of a model's internal reasoning steps.

Formal reasoning differs from the aforementioned studies since it grounds a problem to a precise logical framework that enables deterministic inference procedures. This formal representation is often used in graduate level mathematics and sciences, and can be applied to law and general decision making. LLMs are often tested in autoformalization (the task of translating a problem in natural language to a particular logical framework) (Wu et al., 2022; Poiroux et al., 2025), neuro-symbolic reasoning (Pan et al., 2023) and theorem proving (Quan et al., 2024; Ranaldi et al., 2025). Despite the research done in this area, models are susceptible to content bias (Valentino et al., 2025), meaning that they often side with reasoning that is supported by semantic content (Dasgupta et al., 2024).

3 Methodology

Three different approaches were developed and evaluated: a baseline using a supervised fine-tuning of an Encoder, a two step translation and inference model using aligned LLMs and external logic solvers, and an ontology based method.

3.1 Encoder Classification

As a first approach, we built a supervised classification baseline to predict the logical validity of each syllogism. This follows the idea of using an encoder-based pipeline before moving to more complex reasoning methods. The task is evaluated with a Primary Ranking Metric (PRM) that combines Accuracy and Total Content Effect (TCE), where TCE measures how sensitive a model is to content bias (plausible vs. implausible arguments).

Each example contains two premises and one conclusion. To clarify the structural organization of the data, we reformatted the input as follows: `PREMISE1: ... PREMISE2: ... CONCLUSION: ...`. This simple tagging helps the models focus on the reasoning pattern instead of reading the text as one paragraph.

System	ACC	TCE	PRM
Naive Bayes (official-tuned + lemmatization)	67.19	1.89	32.61
Logistic Regression (tuned + lemmatization)	70.83	2.37	31.96
SVM (tuned + lemmatization)	71.35	3.06	29.71
BERT (GroupDRO final)	70.31	8.09	21.92

Table 1: Example results on the development split using the official evaluation metrics.

Classical ML baseline. We trained several traditional classifiers using a TF-IDF representation of the text. We tested linear models such as Logistic Regression and Linear SVM, as well as Naive Bayes and a Random Forest variant. We also explored small changes in preprocessing, including lemmatization, and parameter tuning. Model selection was guided by the official evaluation metric, rather than by standard classification scores alone.

Encoder fine-tuning (BERT). In addition to classical ML, we fine-tuned an encoder model (BERT) as a sentence classification system. The syllogism text (with premise and conclusion tags) is tokenized and passed to the encoder, followed by a classification head that predicts whether the argument is valid or invalid. Since the shared task penalizes content bias, we also evaluated a debiased variant trained with GroupDRO.

Table 1 reports representative results on a development setting, where we used an 80/20 split of the shared-task training data for model comparison. In this table, official-tuned refers to the Naive Bayes configuration optimized with the official shared-task metric, while tuned + lemmatization refers to configurations selected after hyperparameter tuning and lemmatization. On this development split, the strongest PRM values were obtained by the classical ML pipeline, particularly Naive Bayes and the linear models. BERT achieved competitive accuracy, but with higher content sensitivity, which reduced its PRM under the official evaluation.

The official submission to the shared task was Logistic Regression (tuned + lemmatization). Unlike the development results in Table 1, which were obtained from an 80/20 split of the shared-task training data, the official results were obtained by training this configuration on the full shared-task

training data and generating predictions for the official evaluation set.

3.2 Autoformalization and Logic Solvers

Beyond classification tasks, we test an autoformalization pipeline using logically-aligned LLMs and an external first order logic (FOL) solver. This pipeline is comprised of the following steps: preference-based alignment focused in autoformalization of Natural Language premises, translation of the task's syllogisms into first order logic, and formal verification using Prover9 (McCune, 2005–2010). The motive behind this pipeline lies in analyzing each syllogism using only a logical representation and thus avoiding content bias from an LLM.

3.2.1 Alignment

We use a set of LLMs modified using preference-based alignment. In particular, we use the Llama3 (Grattafiori et al., 2024) set of models aligned using DPO (Rafailov et al., 2024) based on the dataset FOLIO (Han et al., 2024), as described in López-Ponce and Bel-Enguix (2025). The list of checkpoints can be found in table 2.

The prompt used for autoformalization follows the same structure as the ones used for preference-based alignment, and contains information about the autoformalization problem, FOL syntax, and a single example of behavior. Simpler prompts that omit such information downgrade the quality of the aligned model's answers, particularly with the vanilla checkpoints. Instruct models still produce readable answers, but often generate a longer response that requires automatic filtering that can be avoided by using a prompt such as the one used. The prompt can be found in the paper's repository, as well as in appendix A.1.

Checkpoint
`Kurosawama/Llama-3.1-8B-LogLim`
`Kurosawama/Llama-3.1-8B-Instruct-LogLim`
`Kurosawama/Llama-3.2-3B-LogLim`
`Kurosawama/Llama-3.2-3B-Instruct-LogLim`

Table 2: List of LLM checkpoints used for autoformalization.

3.2.2 Autoformalization

Using the previously described models, we translate each syllogism to a formal logicl representation. Each LLM's response isn't perfectly formatted, meaning post processing and cleaning was carried out using regular expressions. An example of the cleaned translation, carried out by LLAMA-3.1-8B-INSTRUCT-LOGLIM, is shown below.

NL Syllogism:

The entire set of fruits is composed of round objects with a citrus peel. At least one orange is not a round object with a citrus peel. Every single fruit is an orange.

FOL Translation:

- $\forall x(Fruit(x) \rightarrow (Round(x) \wedge Citrus(x)))$
- $\exists x(Orange(x) \wedge \neg(Round(x) \wedge Citrus(x)))$
- $\forall x(Fruit(x) \rightarrow Orange(x))$

Prover9 The translated syllogisms are evaluated using Prover9. Each syllogism is modified so that it complies with the prover's syntax[2], this post processing is carried out using regular expressions. The following example shows the final version passed for evaluation.

Prover9 Syntax

- `all.x(fruit(x) -> (round(x) & citrus(x))`
- `exists.x(orange(x) & -(round(x) & citrus(x))`
- `all.x(fruit(x) -> orange(x))`

Results Evaluation is carried out by having the solver infer the final premise from the first two, Prover9 returns TRUE in the event that the solution can be deducted, FALSE elsewhere. These same tags are then converted into the binary classification tags, since a TRUE label corresponds to values that are logically deductible.

Each of the checkpoints was tasked with translating the task's syllogisms in order to be passed into the rest of the pipeline. Table 3 shows each checkpoint's scores in terms of Task 11's evaluation, whereas Table 4 shows a fine-grained analysis of the pipeline's behavior. As expected, the best performing model in terms of the task's evaluation is also the same one with the least amount of syntactical errors. Additionally, -Instruct models make less mistakes than their vanilla counter-parts.

Surprisingly, even the best performing model doesn't match a random guess classification. Given

[2]The dot is used as a separator for each clause in Prover9's original syntax. However, the implementation used in this pipeline corresponds to an updated version (`https://github.com/ai4reason/Prover9`) that modifies this symbol's behavior and allows parsing such as this.

Checkpoint	ACC	TCE	PRM
Llama-3.1-8B	14.71	60.73	21.88
Llama-3.1-8B-Instruct	**15.34**	**63.35**	**21.88**
Llama-3.2-3B	10.66	43.46	21.88
Llama-3.2-3B-Instruct	13.86	51.83	16.67

Table 3: Checkpoints and scores. The checkpoint name is shortened, but all models correspond to the logically aligned versions.

Checkpoint	Bad Auto	Invalid Syntax	XOR
Llama-3.1-8B	14	45	7
Llama-3.1-8B-Instruct	48	32	0
Llama-3.2-3B	74	85	23
Llama-3.2-3B-Instruct	81	71	32

Table 4: Statistical breakdown of checkpoint behavior. **Bad Auto** corresponds to parsed results that don't follow the three syllogism structure, however the parsing might be logically or syntactically valid. **Invalid Syntax** corresponds to syllogisms not accepted by Prover9. **XOR** corresponds to syntactically invalid responses that contain the logical symbol $\oplus$ associated with XOR.

that this pipeline incorporates a deterministic first-order logic solver over a probabilistic classifier, this suggests that the task of autoformalization is a struggling point for LLMs.

3.3 Formal Verification via Ontology Reasoning

As our final approach, we adopt a hybrid neuro-symbolic approach to validate syllogistic reasoning. First, a decoder-based LLM (GPT-4o-mini) translates each natural language syllogism into an OWL 2 ontology serialized in RDF/XML. The generated ontology encodes premises and hypotheses as class axioms. This translation step replaces costly and error-prone manual formalization while keeping the reasoning stage fully symbolic and interpretable.

The resulting ontology is processed in Python V.3.12.12 using Owlready2[3] V.0.49 and evaluated with the HermiT[4] reasoner V1.3.8. Logical validity is assessed through standard OWL consistency checking, since the semantics of OWL is based on Description Logic. An ontology is consistent if it contains no logical contradictions; otherwise, any statement could be inferred, making the reasoning unreliable. The reasoner also identifies unsatisfiable classes (i.e., classes that cannot have instances without introducing inconsistency).

To test whether a hypothesis follows from a set of premises, we apply a *reductio ad absurdum* strategy: we add the negation of the hypothesis to the set of premises and re-run the reasoner. If the ontology becomes inconsistent, then the hypothesis is entailed by the premises. Otherwise, the syllogism is not logically valid. Reasoning is performed under the Open World Assumption (OWA) of OWL, meaning that absence of information does not imply falsity. This ensures that conclusions are derived strictly from explicit logical constraints rather than implicit closed-world assumptions.

This pipeline provides a transparent and verifiable reasoning layer on top of LLM-based semantic parsing.

4 Results

Out of the three previously described methodologies, the best performing one in the test set is the Encoder-based classification. Obtaining a 72.25% Accuracy, 11.77 Content Effect, and a 20.37 Combined Score. This result ranked us 40th out of 45 participants. Table 5 shows the best[5] and worst performing teams, our results and the closest teams to us.

Team Name	Combined Score	Placement
rongchuan	100	1st
abhinandan	23.02	39th
GIL-Zaragoza	**20.37**	**40th**
thiyaga6851	20.19	41st
sujual007	12.81	45th

Table 5: Shortened results list from Codabench.

5 Analysis

5.1 BERT Classification

Our BERT-based classifier reached competitive accuracy, but its final score was lower because its TCE was higher than in the classical ML baselines. On the development split, BERT obtained 70.31 Accuracy, 8.09 TCE, and 21.92 PRM. By contrast, Naive Bayes, Logistic Regression, and SVM obtained lower TCE values and higher PRM scores. This shows that BERT was more affected by whether the arguments sounded plausible or implausible, even when their logical form was similar,

[3] https://owlready2.readthedocs.io/en/v0.49/

[4] http://www.hermit-reasoner.com

[5] 11 Teams achieved 100 Combined Score

and the official metric penalizes this strongly. One possible reason is that BERT is pre-trained on natural language and may rely more on semantic cues and world knowledge than on strict logical validity. In contrast, the classical ML models, based on TF-IDF, may have focused more on lexical and structural patterns in the syllogisms, which seems to have reduced content bias. Although we cannot confirm the exact reason, our results suggest that, in this task, lower content sensitivity was as important as raw accuracy.

5.2 LLMs and Prover9

This pipeline under performs in the task. Even the best performing model (LLAMA-3.1-8B-INSTRUCT) barely reaches a 63% Accuracy. An initial observation is the difference in logical operators between the aligned model's translations and Prover9's accepted syntax. Aligned models make use of the XOR operation, however Prover9 doesn't accept said operation explicitly, meaning that any attempted inference that contains XOR is not evaluated correctly. In order to circumvent such problem, if a XOR is encountered it should be modified into it's logical equivalent $(A \wedge \neg B) \vee (\neg A \wedge B)$. This was a notorious shortcoming of the pipeline, since it ruled out close to 27% of the answers for this checkpoint.

The autoformalization aspect itself is lackluster. Each syllogism is made up of two premises and a conclusion, however the FOL Translations do not always follow the same structure. An example can be seen in the following syllogism from the test dataset:

Syllogism: Not a single object that is a vehicle is a type of transportation. A number of vehicles are buses. The conclusion that some buses are not transportation necessarily follows.

FOL Translation:

- $\forall x(Vehicle(x) \rightarrow \neg IsTypeOf(x, Transportation))$
- $\exists x(Vehicle(x) \wedge Bus(x))$

As much as the model correctly translates the first two premises, the third one is nowhere to be seen. Instead, the model stops generating the translation and reverts to analyzing an example shown in the alignment dataset prompts. This might be due to an overfitting in the alignment procedure, since this behavior is often repeated in all of the aligned models.

5.2.1 Limitations and Improvement

DPO is a preference based alignment algorithm, meaning that there is no mathematical way of validating a generated response during post-training. Algorithms that follow a Verifiable-Reward paradigm (e.g. GRPO, DAPO) can avoid this pitfall and increase a model's performance, since each generation can be automatically validated to better align a model for reasoning tasks such as this one.

Prompt-wise, opting for a multi-shot approach could help models with a longer context windows. In a similar manner, using models with a higher parameter count or models with internal reasoning capabilities can also improve performance due to a higher computational power.

5.3 Ontology Reasoning

Although conceptually appealing due to its transparency and verifiability, the ontological reasoning approach yielded modest performance, achieving a combined score of 11.23 (see Table 6 for detailed metrics). A central challenge lies in prompt design. Due to time and budget constraints, we were only able to evaluate a limited set of prompts. We anticipate that systematic prompt optimization could substantially improve results in future iterations.

A major source of error concerned the translation of natural language premises into formal representations. The OwlReady2 library requires RDF/XML serialization, which is relatively verbose and prone to formatting inconsistencies. This led to recurrent translation and syntax errors in the model outputs. Requesting the model to generate premises directly in OWL format reduced such errors, but additional post-processing was still necessary to ensure compatibility with the reasoning pipeline.

Overall, our findings suggest that manual validation of LLM-generated formalizations remains necessary. As immediate future work, we plan to conduct a more systematic exploration of prompt configurations, experiment with alternative models, including open-source LLMs, and further refine the formalization pipeline to reduce dependency on manual correction.

Model	ACC	TCE	PRM
gpt-4o-mini	48.69	27.08	11.23

Table 6: Ontology reasoning results using the Task evaluation metrics

6 Conclusions

Initially, we can see that models still struggle with autoformalization. Both first order logic and ontologies posted a challenge for state-of-the-art LLMs, meaning that further research has to be carried out in order to enhance these translation capabilities.

Regarding the use of external tools, in order to fully incorporate external solvers for reasoning evaluation the development of a robust syntax parser is vital. Without it, the use of such a solver is heavily limited to generations that follow a very particular syntax. The correct use of an external solver with models capable of generating better formal statements can, in theory, solve this problem in an almost deterministic way.

Finally, it's worth pointing out that classification methodologies can still offer insight into these sort of tasks. These methods tend to have a low content bias, meaning that incorporating in an adequate manner can alleviate some pressure from LLM based systems.

Acknowledgments

This work has been supported by PAPIIT paper IG-400325. Francisco F. López-Ponce thanks SECHITI (CVU: 2045472). F. Bobillo, I. Huitzil and L. Pitarch received support from Projects PID2024-159530OB-I00 (funded by MICIU/AEI/10.13039/501100011033/FEDER, UE) and UZ2024-IyA-02 (funded by University of Zaragoza). Iván Saavedra thanks the PCIC-UNAM graduate program and SECIHTI for academic support (CVU: 935701).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2025. Matharena: Evaluating llms on uncontaminated math competitions.

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

Ishita Dasgupta, Andrew K. Lampinen, Stephanie C. Y. Chan, Hannah R. Sheahan, Antonia Creswell, Dharshan Kumaran, James L. McClelland, and Felix Hill. 2024. Language models show human-like content effects on reasoning tasks. *Preprint*, arXiv:2207.07051.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, and 16 others. 2024. FOLIO: Natural language reasoning with first-order logic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22017–22031, Miami, Florida, USA. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Francisco F. López-Ponce and Gemma Bel-Enguix. 2025. Into the limits of logic: Alignment methods for formal logical reasoning. In *Proceedings of The 3rd Workshop on Mathematical Natural Language Processing (MathNLP 2025)*, pages 112–123, Suzhou, China. Association for Computational Linguistics.

W. McCune. 2005–2010. Prover9 and mace4. `http://www.cs.unm.edu/~mccune/prover9/`.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

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.

Auguste Poiroux, Gail Weiss, Viktor Kunčak, and Antoine Bosselut. 2025. Reliable evaluation and benchmarks for statement autoformalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17947–17969, Suzhou, China. Association for Computational Linguistics.

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, Miami, Florida, USA. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

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, Vienna, Austria. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. Openai gpt-5 system card. *Preprint*, arXiv:2601.03267.

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)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Yuhuai Wu, Albert Q. Jiang, Wenda Li, Markus N. Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. *Preprint*, arXiv:2205.12615.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.

A Prompts

A.1 FOL Autoformalization

Given a problem description and a question, the task is to parse the problem and the question into first order logic formulas. The grammar of the first order logic formula is defined as follows:

1 logical conjunction of expr1 and expr2: $expr1 \wedge expr2$

2 logical disjunction of expr1 and expr2: $expr1 \vee expr2$

3 logical exclusive disjunction of expr1 and expr2: $expr1 \oplus expr2$

4 logical negation of expr1: $\neg expr1$

5 expr1 implies expr2: $expr1 \rightarrow expr2$

6 expr1 if and only if expr2: $expr1 \leftrightarrow expr2$

7 logical universal quantification: $\forall x$

8 logical existential quantification: $\exists x$

Problem: All people who regularly drink coffee are dependent on caffeine. People either regularly drink coffee or joke about being addicted to caffeine. No one who jokes about being addicted to caffeine is unaware that caffeine is a drug. Rina is either a student and unaware that caffeine is a drug, or neither a student nor unaware that caffeine is a drug. If Rina is not a person dependent on caffeine and a student, then Rina is either a person dependent on caffeine and a student, or neither a person dependent on caffeine nor a student. Predicates:

- $Dependent(x)$ ::: x is a person dependent on caffeine.
- $Drinks(x)$ ::: x regularly drinks coffee.

- $Jokes(x)$ ::: x jokes about being addicted to caffeine.
- $Unaware(x)$ ::: x is unaware that caffeine is a drug.
- $Student(x)$ ::: x is a student.

Premises:

- $\forall x(Drinks(x) \rightarrow Dependent(x))$ ::: All people who regularly drink coffee are dependent on caffeine.
- $\forall x(Drinks(x) \oplus Jokes(x))$ ::: People either regularly drink coffee or joke about being addicted to caffeine.
- $\forall x(Jokes(x) \rightarrow \neg Unaware(x))$ ::: No one who jokes about being addicted to caffeine is unaware that caffeine is a drug.
- $(Student(rina) \land Unaware(rina)) \oplus \neg(Student(rina) \lor Unaware(rina))$ ::: Rina is either a student and unaware that caffeine is a drug, or neither a student nor unaware that caffeine is a drug.
- $\neg(Dependent(rina) \land Student(rina)) \rightarrow (Dependent(rina) \land Student(rina)) \oplus \neg(Dependent(rina) \lor Student(rina))$ ::: If Rina is not a person dependent on caffeine and a student, then Rina is either a person dependent on caffeine and a student, or neither a person dependent on caffeine nor a student.

Problem:
{}
Predicates:

A.2 Ontology Reasoning

You are an expert in knowledge representation, Description Logics, and the Semantic Web. Your task is to convert a syllogism expressed in natural language into an OWL TBox schema using RDF/XML
Input characteristics

The input consists of two premises and a conclusion.

Core principle

Do NOT attempt to resolve, correct, or normalize logical inconsistencies. Always preserve the polarity (affirmative or negative) exactly as stated in each sentence, including the conclusion. Maintain the categories and properties as stated. Change only connectors, verbs and adverbs which relate to logical conditions into logical relations existent in OWL.

Mapping rules

Common nouns denoting kinds → OWL Classes (singular, capitalized). Specific entities (e.g., "el león") → model as OWL Classes (not individuals). Verbs or verbal predicates → OWL Object Properties (lowercase, infinitive).

Output constraints

Output only RDF/XML usable via owlready2. No explanations, comments, or extra text. Keep class and property names consistent across all axioms.

Goal

The output ontology may be logically inconsistent and must reflect this faithfully. The ontology should be suitable for automated inconsistency detection with a DL reasoner.

Example:

```
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.semanticweb.org/fbobillo/ontologies/silogismo#"
xml:base="http://www.semanticweb.org/ fbobillo/ontologies/silogismo"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xml="http://www.w3.org/XML/1998/namespace"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

<owl:Ontology rdf:about="http://www.semanticweb.org/fbobillo/ontologies/silogismo"/>

<!-- http://www.semanticweb.org/fbobillo/ontologies/silogismo#respira -->
<owl:ObjectProperty
  rdf:about="http://www.semanticweb.org/fbobillo/ontologies/silogismo#respira"/>

<!-- http://www.semanticweb.org/fbobillo/ontologies/silogismo#Animal -->
```

```
<owl:Class
  rdf:about="http://www.semanticweb.org/fbobillo/ontologies/silogismo#Animal">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource=
        "http://www.semanticweb.org/fbobillo/ontologies/silogismo#respira"/>
      <owl:someValuesFrom rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.semanticweb.org/fbobillo/ontologies/silogismo#Leon -->
<owl:Class rdf:about="http://www.semanticweb.org/fbobillo/ontologies/silogismo#Leon">
  <rdfs:subClassOf
    rdf:resource="http://www.semanticweb.org/fbobillo/ontologies/silogismo#Animal"/>
  <rdfs:subClassOf>
    <owl:Class>
      <owl:complementOf>
        <owl:Restriction>
          <owl:onProperty rdf:resource=
            "http://www.semantic web.org/fbobillo/ontologies/silogismo#respira"/>
          <owl:someValuesFrom rdf:resource=
            "http://www.w3.org/2002/07/owl#Thing"/>
        </owl:Restriction>
      </owl:complementOf>
    </owl:Class>
  </rdfs:subClassOf>
</owl:Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->
<owl:Class rdf:about="http://www.w3.org/2002/07/owl#Thing"/>

</rdf:RDF>
```

Syllogism to solve:
{syllogism}