ILENS: Iterative Logical Enhancement via Neurosymbolic Computation and Common Sense

Anonymous ACL submission

Abstract

Trained on internet-scale datasets, large language models (LLMs) excel in tasks relying on surface patterns and exhibit strong common sense knowledge. However, their performance decreases on tasks requiring deeper reasoning steps. Recent techniques aim to combine the strengths of both reasoning programs and LLMs by converting natural language problems into formal logic specifications, thereby enhancing reasoning task performance. Despite these advancements, LLMs often struggle with ambiguities and complex cases, leading to reasoning errors in the formal method step. In this paper, based on the observation that LLMs can provide the implicit common sense facts when asked explicitly, we propose ILENS (**I**terative **L**ogical **E**nhancement via **N**eurosymbolic Computation and Common **S**ense), a new iterative neurosymbolic system for logical inferences which integrates the two systems in an iterative manner. Initially, we translate the problem specifications into AMR graphs, and then convert them into first-order logic (FOL) expressions to minimize inaccurate interpretations from natural language to FOL. Subsequently, we use formal theorem provers (Prover9, Mace4) to deduce the conclusion. Within this process, we ask the theorem prover to generate counterexamples based on the given premises when the theorem prover fails to provide a definite answer, then prompting the LLM to identify any implicit common sense facts. These facts are then incorporated back into the theorem to attempt proof completion. Through the iterative steps and leveraging the GPT-4 API in conjunction with Prover9 and Mace4, our new proposed ILENS system significantly reduces uncertain and error cases and achieves 80.22% accuracy on the challenging FOLIO dataset, setting a new state of the art.

1 Introduction

Recent advancements in large language models (LLMs), such as ChatGPT/InstructGPT (Ouyang et al., 2022), GPT-3 (Brown et al., 2020), GPT-4 (Achiam et al., 2023), LLAMA (Touvron et al., 2023), and PALM (Chowdhery et al., 2023), have demonstrated significant success across a variety of tasks including text generation, classification, coding, and problem-solving. LLMs are transformer-based models that operate on statistical principles. Despite their considerable success, the generation of outputs in these models relies on probabilistic token prediction (Naveed et al., 2023). However, real-world natural language (NL) is often complex and ambiguous (Nadkarni et al., 2011). Therefore, tasks that require long sequences of logical reasoning, comprehension of implicit natural language statements, or reasoning out of domain remain challenging for LLMs (Liang et al., 2022; Saparov et al., 2024; Anil et al., 2022). Although techniques such as chain of thought (CoT) (Nye et al., 2021; Wei et al., 2022; Wang et al., 2022; Huang and Chang, 2022; Kojima et al., 2022) and in-context learning (ICL) (Min et al., 2021; Dong et al., 2022; Min et al., 2022; Schick et al., 2024) have been proposed to address some of these difficulties, recent studies suggest that the inherent architecture of transformer-based language models still lacks optimal efficiency in deeper proofs logical reasoning (Dziri et al., 2024; Olausson et al., 2023).

Additionally, logical reasoning is crucial for AI-based tasks such as theorem proving, solving mathematical problems with step-by-step solutions, efficient code generation, algorithm design, and answering complex queries. These type of tasks can be challenging because they require long chain of logical reasoning or step by step problem solving. Hence, enhancing the logical capabilities of LLMs and their ability to apply common sense knowledge can significantly improve their performance in mathematics and science-based applications (Li et al., 2021; Jain et al., 2023). The use of logical reasoning can lead to more accurate

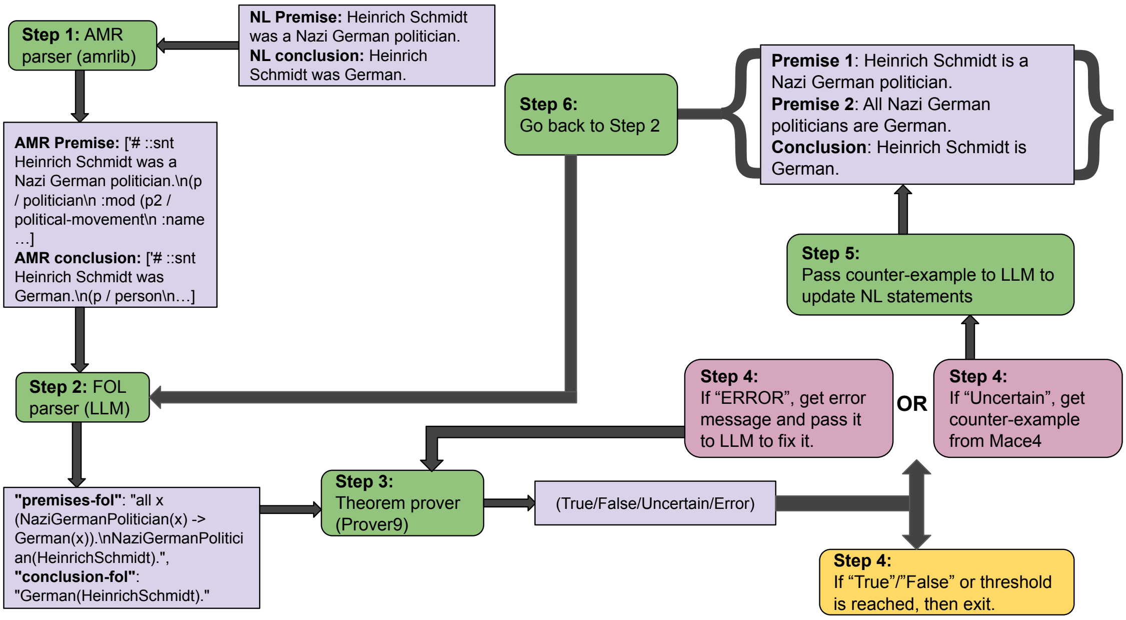


Figure 1: Iterative workflow of ILENS.

and reliable results by reducing hallucinations across a wide range of applications compared to probabilistic token prediction (Xu et al., 2024; Olausson et al., 2023; Zhang et al., 2024). Recent research has combined powerful LLMs with formal theorem provers, leveraging the strengths of both approaches to create more robust and capable systems. This integration aims to enhance the logical reasoning capabilities of LLMs, enabling them to perform tasks that require rigorous logical reasoning alongside natural language understanding (Olausson et al., 2023; Pan et al., 2023).

In this work, we implement ILENS (**I**terative **L**ogical **E**nhancement via **N**eurosymbolic Computation and Common **S**ense), a system that combines LLMs with a theorem prover in an iterative fashion, thereby enhancing the logical capabilities of LLMs. ILENS is an iterative neurosymbolic system where the language model converts the natural language statements first to abstract meaning representation (AMR) (Banarescu et al., 2013; Knight et al., 2021) using a parser, then translates the AMR graphs to first-order logic (FOL) expressions (Enderton, 2001; Barker-Plummer et al., 2011). The translated FOL expressions are fed to the theorem provers (Prover9, Mace4) (McCune, 2005–2010) to determine the truth value of the inference. In the cases of indefinite responses or syntax errors generated by the prover, our system improves them. If the theorem prover fails to provide a definite answer, we ask it to generate a counter-example. This example is then used as a reference for the language model to find any missing links or facts in the given natural language premises (NL) to enrich the NL premises. The improved NL statements are then passed through the parsers and theorem prover to get updated inference output. If there is a syntax error in the FOL expressions, the language model is prompted to fix the error and the improved FOL is passed through the prover again. This iteration continues until the theorem prover can find a definite answer or reach a specified threshold.

ILENS leverages the ability of LLM to follow instructions and its strength of common sense knowledge for FOL translation while offloading logical reasoning deduction to a formal theorem prover. Hence, the success of ILENS lies in two novel ideas. The first key idea is **translating AMR to FOL** rather than directly translating FOL from NL. Abstract Meaning Representation (AMR) tends to be more structured and semantically clearer, making it potentially easier to translate into FOL without errors whereas translating natural language statements to first-order logic (FOL) can be more complex due to the ambiguity and nuances of human language (as shown in Figure 2). Our baseline system performs well with this added step in FOL translation with improved FOL expressions in secenarios where the NL statements have implicit information. The second key idea is to **iterate logical reasoning** with theorem prover which updates the NL premises with any missing facts provided by LLM based on the counter-example from the theorem prover. By adding this iterative step, our system improves accuracy significantly from the baseline system by 29%. We can summarize our contributions:

- We introduce ILENS, a new iterative neurosymbolic system for logical inferences. Our system successfully improves the accuracy on task that requires logic reasoning with LLMs and external theorem prover setting a new state of the art. On FOLIO dataset (Han et al., 2022) the accuracy achieved is 80.22% which is about 7.7% higher than the previous benchmark model (Olausson et al., 2023).
- Our experiments demonstrates that combining common sense knowledge of LLM and iterative logic inference by formal theorem provers can improve the task of logic inference deduction.
- We do a thorough comparison of ILENS to baseline systems and also perform error analysis.

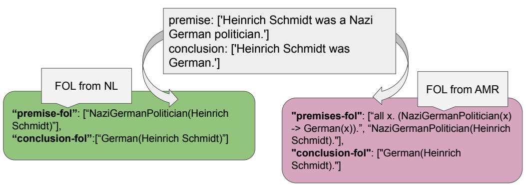


Figure 2: Comparing conversion of first-order logic (FOL) from abstract meaning representation (AMR) and natural language (NL).

2 Related work

Semantic parsing with language models is the process of converting natural language into a structured, machine-readable representation which has seen significant advancements with the advent of LLMs. Semantic parsing traditionally involves mapping natural language utterances to formal representations like AMR, lambda calculus, or SQL queries (Ge and Mooney, 2005; Kamath and Das, 2018). The goal is to capture the underlying meaning of the input text in a way that facilitates further processing by downstream applications (Wang et al., 2015; Berant and Liang, 2014). Works by (Zhang et al., 2019; Bevilacqua et al., 2021) use a transformer-based architecture to achieve state-of-the-art results in AMR parsing tasks. Recent developments have leveraged the power of LMs, such as GPT-3, BERT, and T5, to enhance semantic parsing capabilities (Raffel et al., 2020; Shin and Van Durme, 2021; Hahn et al., 2022; Wong et al., 2023). LogicLLAMA (Yang et al., 2023) can directly translate natural language into FOL rules along with correct predictions made by GPT-3.5, which achieves performance comparable to GPT-4 with a fraction of the cost.

Common sense reasoning in language models allows language models to interpret implicit information, disambiguate meanings, and make logical inferences that are intuitive for humans. COMET (Bosselut et al., 2019) uses the transformer model to generate inferential knowledge, augmenting the language model's ability to reason about everyday scenarios. Models like BERT and GPT-3 have been fine-tuned on datasets specifically designed to improve the capability of common sense reasoning (Talmor et al., 2018; Rajani et al., 2019; Sap et al., 2020; Liu et al., 2021; Bian et al., 2023).

Reasoning through neurosymbolic approaches in LLMs combines neural networks with symbolic reasoning systems to integrate structured knowledge and logical inference capabilities (Hitzler et al., 2022). The integration of structured knowledge bases with neural models can enhance the reasoning abilities of LLMs (Zhang et al., 2023b). Recent studies have shown how LLMs have a significant gap in logical reasoning when compared to human judgment (Press et al., 2022; Wang et al., 2024; Gu et al., 2024).

Given these extensive backgrounds, several works have been done regarding the optimal methods for integrating LLMs with symbolic components to enhance logical reasoning capabilities. Arabshahi et al. (2021) show how combining a neurosymbolic system with common sense through conversation can complete its reasoning chains. Similarly, Manhaeve et al. (2021) develops systems that combine neural and symbolic components to perform complex reasoning tasks. In DSR-LM (Zhang et al., 2023a), pre-trained LMs govern the perception of factual knowledge, and a symbolic module performs deductive reasoning. Logic-LM proposed by

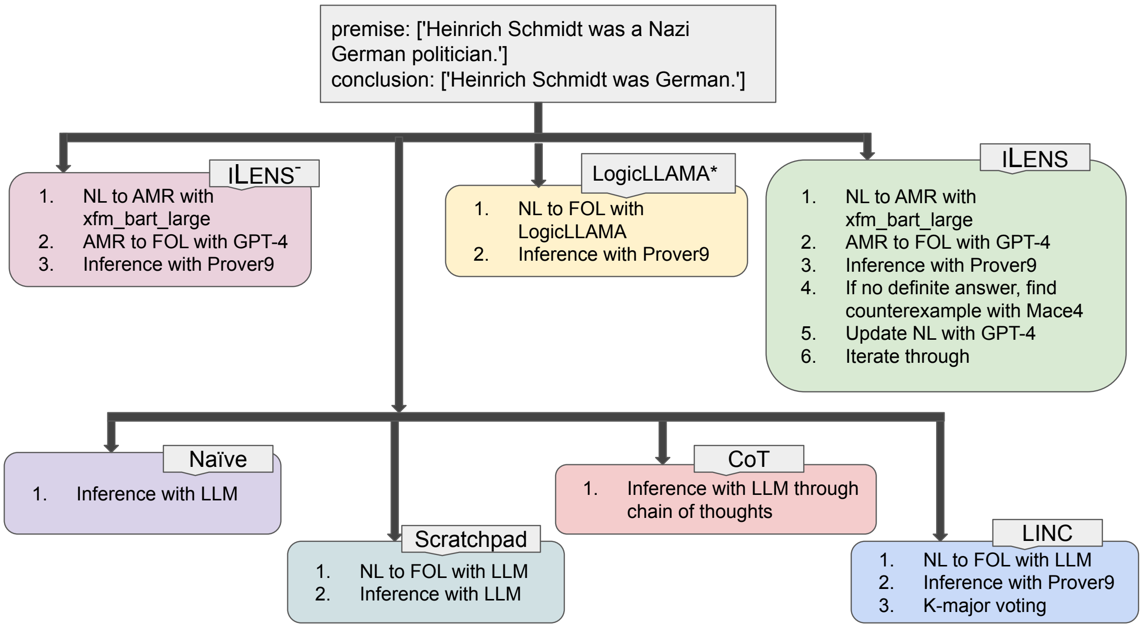


Figure 3: Brief description of the models used to compare the performance of ILENS.

(Pan et al., 2023) integrates LLMs with symbolic solvers to improve logical problem-solving. Additionally, they introduce a self-refinement module that learns to modify inaccurate logical formulations using error messages from the symbolic theorem prover as feedback. However, their idea of self-refinement only focuses on syntax correction, which is significantly different from our approach and contributions. SATLM (Ye et al., 2023) uses an LLM to generate a declarative task specification rather than an imperative program and leverage an off-the-shelf automated theorem prover to derive the final answer. LINC (Olausson et al., 2023) uses LLMs as the semantic parser and offloads the logical reasoning task to an external theorem prover. Our work is inspired by LINC, where our methodology enhances the natural language understanding of LLMs by converting nature language premises to AMR, thereby reducing ambiguity. Moreover, iterative logical reasoning can incrementally add more facts or rules to the theorem prover, resulting in significant performance improvements in our system.

Tool usage for application task augments LMs with external tools such as mathematical and scientific computation tools, code interpreters, knowledge base retrieval systems, and translation services. This approach leverages the strengths of both the LMs and specialized tools, resulting in a more powerful and versatile system. Tool usage can be done in two ways. **First**, *External Tool Integration Without Direct LM Awareness* where the language model is not directly aware of the external tool or the procedures it uses. The integration occurs at a higher level, where the outputs from the LM are processed by external systems to perform specific tasks, such as external code interpreters (Gao et al., 2023; Drori et al., 2022; Azerbayev et al., 2022). For theorem proving, existing works (Wu et al., 2022; Jiang et al., 2022) rely on external theorem provers to get inferences. We follow this approach in our work by invoking external theorem provers (Prover9, Mace4) for logical reasoning. **Second**, *Direct Tool Invocation by the LM* where the LM is responsible for invoking external tools through API calls (Schick et al., 2024; Thoppilan et al., 2022).

3 ILENS

ILENS (Iterative Logical Enhancement via Neurosymbolic Computation and Common Sense) is an iterative neurosymbolic system augmented with external theorem provers (Prover9, Mace4) for end-to-end logical reasoning. Our framework (Figure 1) consists of six stages.

- **Step 1:** We use a semantic parser (Goodman, 2020) with LM to translate NL premises and conclusion pair to AMR graphs. The given premises and conclusion pair is converted to AMR graphs using an LM pre-trained with AMR dataset. More details on the LM used can be found in Section 4.3.

- **Step 2:** The AMR graphs are translated to FOL expressions using GPT-4 API through prompts with ICL. More details are provided in Section 4.2 and Appendix B
- **Step 3:** The translated FOL is then fed to the formal theorem prover to determine the truth value of the inference. We use Prover9, an automated theorem-proving system for first-order and equational logic extensively utilized within the logic research community (McCune, 2005) for inference deduction. For the response of Prover9, we closely follow the work done in LINC (Olausson et al., 2023), therefore the prover either returns a value from the set {True, False, Uncertain} or raises an exception due to incorrect FOL syntax (e.g., if the formulae have unbalanced parentheses or any unknown symbols or operators are not recognized by the prover).
- **Step 4:** If Prover9 is able to determine a definitive response, the program proceeds with the next example. However, the process involves iterative methods when more complex scenarios arise. Specifically, if Prover9 returns an {Uncertain} response which indicates the inability to find a definitive solution, the FOL premises are forwarded to Mace4, a software tool designed to find finite models for first-order logic statements and is often used in conjunction with theorem provers such as Prover9. In this case, Mace4 attempts to identify a counterexample to the premises and advances the process to **Step 5**. On the other hand, if an error occurs during inference, the error message is passed to the LLM along with the FOL expressions. The LLM then attempts to correct the errors in the FOL statements before going back to **Step 3**.
- **Step 5:** The counterexample and the NL statements are then sent to the LLM (GPT-4 API) to find any missing fact or value. Using its common sense ability, the LLM updates the NL statements with the newly found missing fact.
- **Step 6:** The updated NL premise and conclusion pair goes back to the FOL parser in **Step 2** and continues through the process till the prover is able to find a definite inference.

The success of ILENS hinges on accurately translating NL statements into FOL expressions and augmenting both Prover9 and Mace4 to perform logical inference and identify any missing links in the given premises. Given the complexity of human language, our framework prioritizes formal provers over LLMs to capture every nuance of factual information. This approach ensures there are no hallucinations or incorrect representations in the updated NL statements in **Step 4** and **Step 5**. However, a notable drawback is the potential for semantic and syntax errors in the FOL expressions produced by the LLM. To mitigate this, we first convert NL statements into AMR graphs and then transform these AMR graphs into FOL expressions. The key advantage of AMR lies in its structured representation of entities and relationships in natural language, thereby reducing the ambiguity of NL statements (as shown in Figure: 2).

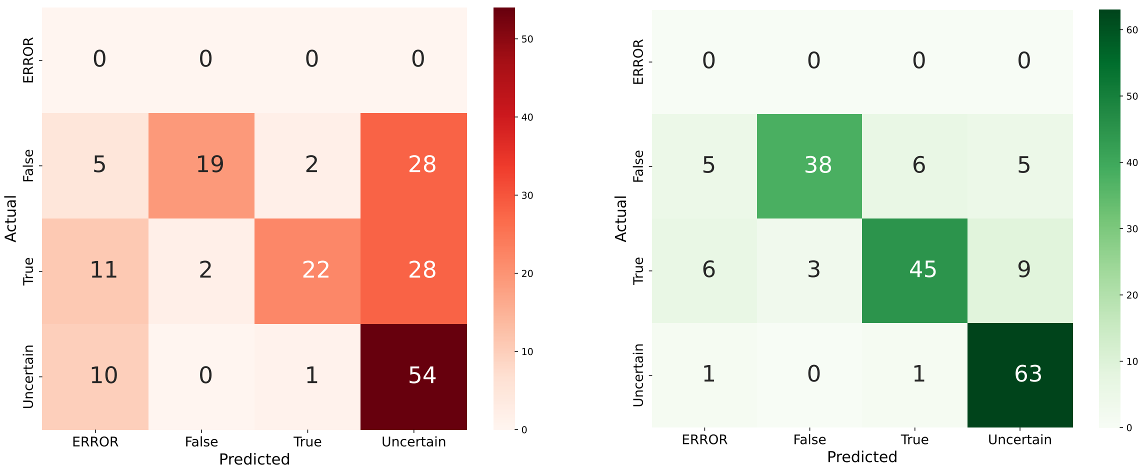
4 Experimental Setup

In this section, we introduce our experimental setup, including the dataset and models used, as well as the baselines against which ILENS is compared. We provide the source code link[1] to our experiments.

4.1 Dataset

For our experiments, we use the FOLIO dataset (Han et al., 2022), which is a collection of annotated natural language statements converted into first-order logic (FOL) expressions, designed for evaluating the performance of logical inference systems. We have considered the validation set of FOLIO for evaluation. Additionally, the dataset requires pre-processing in order to have the right syntax compatible with Prover9, which can reference LINC's pre-processing step (Olausson et al., 2023). The original FOLIO validation set contains 204 examples. However, after pre-processing the dataset, 22 examples contained syntax errors, leaving 182 examples for evaluation. The pre-processing is conducted to ensure that the FOL expressions in FOLIO are in the correct syntactical format for Prover9 and Mace4. We run the 182 examples on our systems ILENS^{-} and ILENS. (System details are included in Section 4.3 and 4.4). More information on pre-processing is in Appendix A

[1] Code: Project Code



(a) Confusion matrix of iLens$^-$.

(b) Confusion matrix of iLens(Mode-2).

Figure 4: Performance of iLens$^-$ (baseline) and iLens(Mode-2).

4.2 In-context learning

We have chosen six diverse examples from the FOLIO training set for in-context learning (ICL). For our baseline experiment, we use only four out of the six examples. Here, the NL statements in the examples are converted to AMR graphs. Along with the FOL expressions and AMR graphs, the updated examples from the training set are provided to the GPT-4 API through prompts. For iLens, we run the experiment with two different modes. Mode-1 uses two out of six examples for ICL and iteration happens only twice. Mode-2 uses six examples for ICL and iteration happens four times. More details on ICL can be found under Appendices A and B

4.3 Models

In our experiments, we use different language models for different tasks. For converting NL statements to AMR graphs, we use parse_xfm_bart_large (base model:facebook/bart-large, version: 0.1.0 date: 2022-02-16, size: 1.4GB, smatch score: 83.7 SMATCH) for sentence to graph conversion. It is trained and scored on AMR-3 (LDC2020T02) (Knight et al., 2021) using num_beams=4. For more information, please refer to the model on GitHub.[2] We use GPT-4 (OpenAI, 2023) API [3] [4] for AMR to FOL conversion as well as for updating the NL statements with missing links found from Mace4. We consider temperature $T = 0$ for our baseline system and temperature $T = 0.2$ for the main system. We use the NLTK [5] extension of Prover9 and Mace4 which is a python extension for the provers.

4.4 Baselines

We compare iLens to three baselines namely iLens$^-$, LINC (Olausson et al., 2023), and LogicLLAMA*. For iLens$^-$, we consider our framework without the iteration process (**Step 4**, **Step 5** and **Step 6**). From LINC, we consider their Naïve, Scratchpad, CoT and original models for GPT-3.5 and GPT-4. In Naïve, the model is given the NL premises and is asked to directly generate the label. In Scratchpad, the model is asked to first generate FOL expressions corresponding to the premises, and then generate the label. In CoT, standard technique of CoT prompting is used to deduce the truth value. In LINC, the LLM is used as a semantic parser and logical inference is done by Prover9. LogicLLAMA* is a joint system with LogicLLAMA (Yang et al., 2023) for FOL translation and Prover9 for logic inference. For better understanding of the baseline models used to compare iLens, we show a schematic representation in Figure 3.

5 Results & Analysis

In this section we provide details of our experimental results, comparisons of our systems with the other baselines, and an analysis of different syntax and semantic errors.

[2] The parse xfm model

[3] https://openai.com/index/gpt-4/

[4] The exact number of parameters in GPT-4 has not been officially disclosed by OpenAI. However, reports and credible sources suggest that GPT-4 is significantly larger than its predecessor, GPT-3, which has 175 billion parameters.

[5] NLTK python extension

5.1 Experimental results

Figure 4 shows the performance of our baseline system ILENS$^-$ and iterative system ILENS. We can observe from ILENS$^-$'s performance that even though it is able to deduce truth values correctly in 95 out of 182 cases, there are still 26 syntax errors and 110 indefinite logic inferences. This signifies that AMR to FOL conversion can help during the translation step. However, ILENS$^-$ does not include the iteration process, therefore there is no improved logical reasoning. We discuss error analysis in detail under Section 5.3. Our iterated system ILENS has been configured into two modes based on Section 4. **Mode-1** runs with two iterations and two-shot learning and **Mode-2** runs with four iterations and six examples for few-shot learning. In figure 4 (b) we provide the performance of ILENS **Mode-2**. As we can see with increased iterations and examples for few-shot learning, the performance improves significantly. The syntax errors as well as the indefinite logic inferences have been reduced by increasing the prediction accuracy to 36%, 35%, and 13% for "True", "False" and "Uncertain" labels respectively. We provide more information on ILENS **Mode-1** under Appendix C. Due to resource limitation, we were unable to experiment beyond four iterations. With more iterations, the system may be able to resolve the syntax errors however the possibility of hallucination by LLM needs to be further explored.

5.2 Comparison with other baseline models

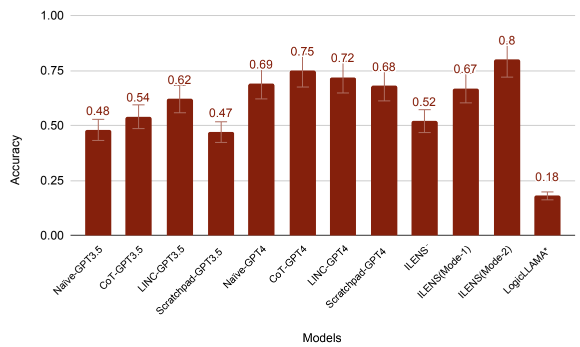


Figure 5: Results of our systems ILENS$^-$, ILENS with the baseline models.

We compare our baseline and iterative systems (Mode-1 and Mode-2) Figure 5 with the other baseline models described in Section 4.4. From this figure, we can see that ILENS(Mode-2) surpasses the performance of the previous models by achieving 80.22% accuracy which is about 4.92% higher than CoT-GPT4 and 7.72% higher than LINC-GPT4 (Olausson et al., 2023). Additionally, we notice a significant improvement in performance with added iterations between ILENS$^-$ and ILENS(Mode-1 and Mode-2). This clearly shows how iterative logic inference with external theorem provers like Prover9 and Mace4 can help improve logical reasoning. Further, we notice that by increasing the iteration from two to four, the Mode-2 system improves its performance by 13%. Additionally, we investigate individual predictions of different labels and show the result in Figure 6.

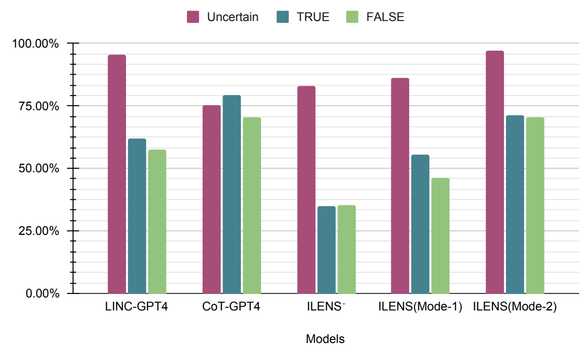


Figure 6: Comparison of accuracy in different categories across different models and systems. The categories considered here are "True", "False" and "Uncertain".

5.3 Error analysis

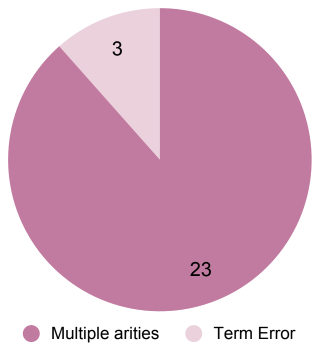
We perform a thorough error analysis on the models' performances. We notice that FOL expressions generated from GPT-4 API has some types of errors.

FOL generated by LLM with syntax errors

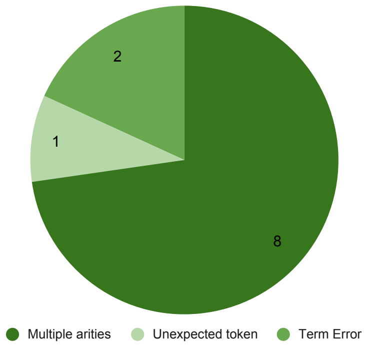
Figure 7 shows the details of different syntax errors generated by the API in FOL expressions.

* **Arity issues:** FOL expressions sometimes contain multiple arities or symbols/arities are used as both relation and function. One such example is:
NL Premise: "If Greyhound is not an airline, then there are no Greyhound planes."
NL Conclusion: "A Greyhound is not a Boeing 707."
FOL Premise: "-Airline(Greyhound) -> -exists x. (Plane(x) & Greyhound(x))"
FOL Conclusion:"-Boeing707(Greyhound)"
Here, "Greyhound" is used both as a predicate and a constant value which is conflicting to Prover9.

* **Term error:** When the FOL statement has unnecessary tokens which is not readable by the



(a) Syntax errors of ILENS$^-$.



(b) Syntax errors of ILENS(Mode-2).

Figure 7: Performance of ILENS$^-$ (baseline) and ILENS(Mode-2).

Prover9, it causes this type of error. For example: *FOL Premise:* "all x. (design_style(x, Zaha_Hadid) -> Timeless(x))" will generate an error like "sread_term error"

* **Unexpected token:** This type of error arises when the format of the FOL does not match the Prover9 format, for instance, when there are unbalanced parentheses or any unexpected symbols like *FOL Premise:* "all x, y, z. ((LocatedAt(x, y) & LocatedAt(y, z)) -> LocatedAt(x, z))

Unable to assume facts when given information is incomplete

We observe that ILENS is not able to deduce logic inference correctly when there is incomplete information. Incomplete information is different from missing links, missing rules, or implicit information hidden in the Natural Language. Here is one such example:

NL Premise: ["All rabbits are cute. ",
"Some turtles exist. ",
"An animal is either a rabbit or a squirrel.",
"If something is skittish, then it is not still.",
"All squirrels are skittish.",
"Rock is still."]
NL Conclusion: "Rock is a turtle or cute."
Actual label: True
Predicted label: Uncertain

Unless the LLM assumes some information about "Rock", it will not be able to get a definite answer for the inference.

The error analysis shows us the scenarios where our system fails to successfully deduce logic inference. The scenarios include FOL expressions generated by GPT-4 with syntax errors and NL statements with incomplete information.

6 Conclusion and Future Work

We present ILENS, an iterative neurosymbolic system augmented with external theorem provers. ILENS is built on two novel ideas: using abstract meaning representation (AMR) to convert text into first-order logic (FOL) expressions, and iterating logic inference using counterexamples generated by Mace4 to improve logical reasoning. Our system significantly outperforms baseline models using similar evaluation techniques. We successfully demonstrate that increasing the number of iterations can enhance the performance of logic inference. This work supports the hypothesis that augmenting an external theorem prover with a large language model (LLM) can improve truth value inference deduction. Thus, the success of ILENS paves the way for future research in neurosymbolic computation for reasoning from natural languages. Future work could explore integrating other forms of symbolic reasoning, expanding the range of natural language inputs, and enhancing the scalability of such systems.

Limitations

Due to limited resources, we were able to run our experiments on one dataset (FOLIO validation (Han et al., 2022)) and with iteration up to four. However, the workflow of our system is not specific to any dataset. Therefore, with some simple data pre-processing it can be used on multiple datasets and with different provers.

Ethics Statement

No ethical violations were anticipated or encountered during the course of this research. As such, no ethical approval was required for this work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. 2022. Exploring length generalization in large language models. *Advances in Neural Information Processing Systems*, 35:38546–38556.

Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. 2021. Conversational neuro-symbolic commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4902–4911.

Zhangir Azerbayev, Bartosz Piotrowski, and Jeremy Avigad. 2022. Proofnet: A benchmark for autoformalizing and formally proving undergraduate-level mathematics problems. In *Second MATH-AI Workshop*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

David Barker-Plummer, Jon Barwise, and John Etchemendy. 2011. *Language, proof, and logic*. Center for the Study of Language and Information/SRI.

Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1415–1425.

Michele Bevilacqua, Rexhina Blloshmi, and Roberto Navigli. 2021. One spring to rule them both: Symmetric amr semantic parsing and generation without a complex pipeline. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12564–12573.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin Dong. 2023. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. *arXiv preprint arXiv:2303.16421*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Iddo Drori, Sarah Zhang, Reece Shuttleworth, Leonard Tang, Albert Lu, Elizabeth Ke, Kevin Liu, Linda Chen, Sunny Tran, Newman Cheng, et al. 2022. A neural network solves, explains, and generates university math problems by program synthesis and few-shot learning at human level. *Proceedings of the National Academy of Sciences*, 119(32):e2123433119.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Herbert B Enderton. 2001. *A mathematical introduction to logic*. Elsevier.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Ruifang Ge and Raymond Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 9–16.

Michael Wayne Goodman. 2020. Penman: An open-source library and tool for amr graphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 312–319.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. 2024. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*.

Christopher Hahn, Frederik Schmitt, Julia J Tillman, Niklas Metzger, Julian Siber, and Bernd Finkbeiner. 2022. Formal specifications from natural language. *arXiv preprint arXiv:2206.01962*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming

Xiong, and Dragomir Radev. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Pascal Hitzler, Aaron Eberhart, Monireh Ebrahimi, Md Kamruzzaman Sarker, and Lu Zhou. 2022. Neurosymbolic approaches in artificial intelligence. *National Science Review*, 9(6):nwac035.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. 2022. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. *arXiv preprint arXiv:1812.00978*.

Kevin Knight, Bianca Badarau, Laura Baranescu, Claire Bonial, Madalina Bardocz, Kira Griffitt, Ulf Hermjakob, Daniel Marcu, Martha Palmer, Tim O'Gorman, et al. 2021. Abstract meaning representation (amr) annotation release 3.0.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2021. A systematic investigation of commonsense knowledge in large language models. *arXiv preprint arXiv:2111.00607*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2021. Generated knowledge prompting for commonsense reasoning. *arXiv preprint arXiv:2110.08387*.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2021. Neural probabilistic logic programming in deepproblog. *Artificial Intelligence*, 298:103504.

W McCune. 2005. Prover9 and mace4, software available online at http://www. cs. unm. edu/˜ mccune.

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

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. 2011. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Nick Barnes, and Ajmal Mian. 2023. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

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. *arXiv preprint arXiv:2310.15164*.

OpenAI. 2023. Gpt-4: Generative pre-trained transformer.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. *arXiv preprint arXiv:2305.12295*.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. 2022. Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 27–33.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2024. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Richard Shin and Benjamin Van Durme. 2021. Few-shot semantic parsing with language models trained on code. *arXiv preprint arXiv:2112.08696*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024. Can llms reason with rules? logic scaffolding for stress-testing and improving llms. *arXiv preprint arXiv:2402.11442*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a semantic parser overnight. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1332–1342.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Lionel Wong, Gabriel Grand, Alexander K Lew, Noah D Goodman, Vikash K Mansinghka, Jacob Andreas, and Joshua B Tenenbaum. 2023. From word models to world models: Translating from natural language to the probabilistic language of thought. *arXiv preprint arXiv:2306.12672*.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. *Advances in Neural Information Processing Systems*, 35:32353–32368.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Yuan Yang, Siheng Xiong, Ali Payani, Ehsan Shareghi, and Faramarz Fekri. 2023. Harnessing the power of large language models for natural language to first-order logic translation. *arXiv preprint arXiv:2305.15541*.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satisfiability-aided language models using declarative prompting. *arXiv preprint arXiv:2305.09656*.

Hanlin Zhang, Jiani Huang, Ziyang Li, Mayur Naik, and Eric Xing. 2023a. Improved logical reasoning of language models via differentiable symbolic programming. *arXiv preprint arXiv:2305.03742*.

Honghua Zhang, Meihua Dang, Nanyun Peng, and Guy Van den Broeck. 2023b. Tractable control for autoregressive language generation. In *International Conference on Machine Learning*, pages 40932–40945. PMLR.

Jiawei Zhang, Chejian Xu, Yu Gai, Freddy Lecue, Dawn Song, and Bo Li. 2024. Knowhalu: Hallucination detection via multi-form knowledge based factual checking. *arXiv preprint arXiv:2404.02935*.

Sheng Zhang, Xutai Ma, Kevin Duh, and Benjamin Van Durme. 2019. Amr parsing as sequence-to-graph transduction. *arXiv preprint arXiv:1905.08704*.

A FOLIO Dataset

A.1 Pre-processing

In order to pre-process the data, we follow the same technique done in LINC (Olausson et al., 2023). We first reformat the dataset with correct symbols accepted by Prover9 and Mace4. For one of our baselines LogicLLAMA*, we preprocess the dataset generated through LogicLLAMA model (Yang et al., 2023) the same way.

A.2 Few-shot examples

For in-context learning (ICL) we consider the training set of the FOLIO dataset. We pick six diverse examples whose labels were "True", "False" and "Uncertain". We consider the following examples from the Yale-LILY/FOLIO website with the following 24, 61, 149, 262, 264, 685. We do not provide the labels of the examples during few-shot learning. For our baseline and ILENS(Mode-1) we randomly pick four and two out of these six examples respectively.

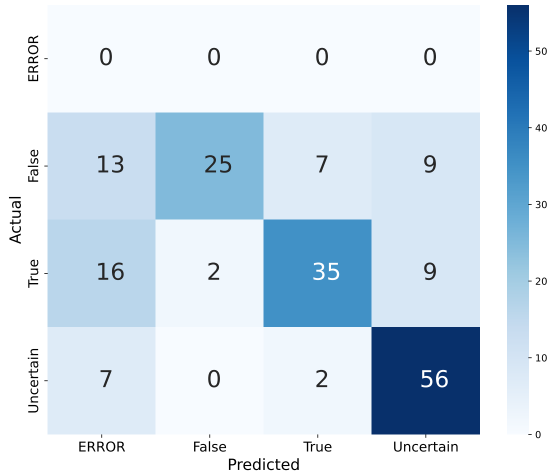
B Prompts used for FOL generation

For our system, we use in-context learning, therefore using prompts to ask the GPT-4 API to generate FOL expressions or correct/update FOL expressions. We have provided the details of the prompts we have used in Table: 1

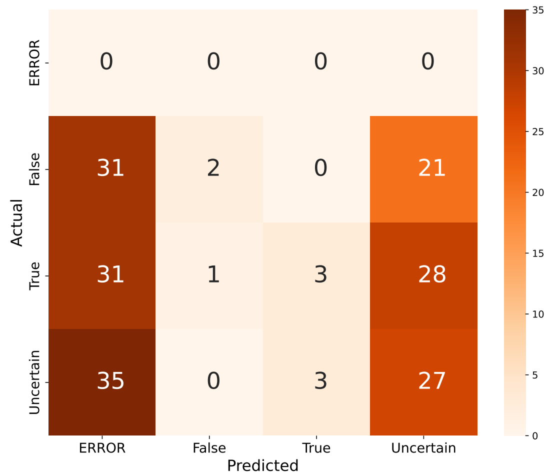
C Detailed result of ILENS (Mode-1) and LogicaLLAMA*

We provide below the performances of ILENS(Mode-1) and LogicLLAMA* in Figure 8. For ILENS(Mode-1), we can see a clear improvement in performance over ILENS^{-}. However, there still exists considerable amount of errors which is overcome by ILENS(Mode-2) indicating more iterations for logic inference can be helpful to draw the truth values.

LogicaLLAMA* on the other hand, performs very poorly on FOLIO dataset. After generating the validation dataset with LogicLLAMA (Yang et al., 2023) we preprocess the dataset (mentioned under Appendix A). Then we pass it through the Prover9 for logic inference. As we can see from Figure 8(b), it contains many errors in the conversion. The errors include the ones mentioned in Section 5.3 and several other syntax errors. Also, we know that the conversion done by LogicLLAMA* is incorrect semantically since most of the labels are incorrectly predicted "Uncertain".



(a) Confusion matrix of ILENS(Mode-1)



(b) Confusion matrix of LogicLLAMA*

Figure 8: Confusion Matrix of ILENS(Mode-1) and LogicLLAMA*.

iLens^{-}	I will provide you with premises and conclusions in AMR, and you will convert them into First Order Logic (FOL) expressions. Follow the given examples for format and syntax. <Examples:> Ensure that: - Symbols are consistently used as either predicates or functions. - Quantifiers are correctly placed. - No quotations are required for any proper noun. - The FOL expressions are valid and well-formed for use in theorem provers like Prover9. - Make sure the FOL expressions are consistent, syntactically correct, and have balanced parentheses. - Make sure the output is not like a chat response. Your output should be a dictionary with the keys "premise-fol" for $premise\_graphs\_list$ with all FOL expressions /in a single list and "conclusion-fol" for $conclusion\_graphs\_list$ with all FOL expressions in a single list.
iLens (Update with counter-example)	Your task is to read and understand the $counter\_example$ generated from Mace4 and use common sense knowledge to find any missing information or logic chain and generate First Order Logic (FOL) from the /provided natural language $premises$ and $conclusion$. Follow the given example for format and syntax. <Example:> Ensure that: - You do not use symbols/arities as both relation and function. - The FOL expressions are valid and well-formed for use in theorem provers like Prover9 with consistent arities. - The FOL expressions are consistent, syntactically correct, and have balanced parentheses. - You do not describe your answer like a chat. - You do not put quotations around any proper nouns or person's names. - You do not use decimal numbers - You respond only with the JSON dictionary and nothing else. - Your output includes both premise and conclusion expressions. Your output should be a dictionary with the keys "premises-FOL" for premises with all FOL expressions in a single list and "conclusion-FOL" for conclusion with FOL expression in a single list.
iLens (Fix error)	Your task is to fix some errors in first order logic statements. I will provide you with the $error$, the $premise\_fol$, and the $conclusion\_fol$ such that they do not contain that error. Follow the given examples for format and syntax: <Examples:> Ensure that: - You use common sense and do not use symbols/arities as both relation and function. - The FOL expressions are valid and well-formed for use in theorem provers like Prover9 with consistent arities. - The FOL expressions are consistent, syntactically correct, and have balanced parentheses. - You do not describe your answer like a chat. - You do not use decimal numbers. - You do not put quotations around any proper nouns or person's names. - You respond only with the JSON dictionary and nothing else. - Your output includes both premise and conclusion expressions. Your output must be a JSON dictionary with the keys "premises-FOL" for premises (a single list of FOL expressions) and "conclusion-FOL" for the conclusion (a single list of FOL expressions).

Table 1: 2-6-shot prompts for iLens^{-}, iLens(Mode-1 and Mode-2)