Do LLMs Really Struggle at NL-FOL Translation? Revealing their Strengths via a Novel Benchmarking Strategy

Andrea Brunello[1], **Luca Geatti**[1], **Michele Mignani**[1], **Angelo Montanari**[1], **Nicola Saccomanno**[1]

[1]University of Udine, Italy
name.surname@uniud.it

Abstract

Due to its expressiveness and unambiguous nature, First-Order Logic (FOL) is a powerful formalism for representing concepts expressed in natural language (NL). This is useful, e.g., for specifying and verifying desired system properties. While translating FOL into human-readable English is relatively straightforward, the inverse problem, converting NL to FOL (NL-FOL translation), has remained a longstanding challenge, for both humans and machines. Although the emergence of Large Language Models (LLMs) promised a breakthrough, recent literature provides contrasting results on their ability to perform NL-FOL translation. In this work, we provide a threefold contribution. First, we critically examine existing datasets and protocols for evaluating NL-FOL translation performance, revealing key limitations that may cause a misrepresentation of LLMs' actual capabilities. Second, to overcome these shortcomings, we propose a novel evaluation protocol explicitly designed to distinguish genuine semantic-level logical understanding from superficial pattern recognition, memorization, and dataset contamination. Third, using this new approach, we show that state-of-the-art, dialogue-oriented LLMs demonstrate strong NL-FOL translation skills and a genuine grasp of sentence-level logic, whereas embedding-centric models perform markedly worse.

Code — https://github.com/dslab-uniud/NL-FOL-LT

Non-archival contribution — This is the Camera Ready (1 extra page) version of a paper accepted at AAAI 2026 Main Track; do not include in the Bridge proceedings. If you need to refer to this work cite Brunello et al. (2026).

Introduction

Natural language (NL) stands as humanity's primary and most intuitive means of encoding and transmitting knowledge, thanks to its expressiveness and remarkable flexibility. Nevertheless, these very characteristics, while enabling rich communication, also introduce challenges such as inherent ambiguity and often a lack of complete context. To overcome these issues, formal languages, like First-Order Logic (FOL), offer a powerful alternative. FOL provides an unambiguous and highly expressive framework for representing complex concepts, making it particularly valuable for tasks like, for instance, specifying and verifying system properties. This is especially crucial in critical scenarios, where domain experts must completely and unambiguously define desired or unwanted system behaviour to perform formal verification. However, despite their usefulness, the correct specification and interpretation of logic formulas typically requires a strong mathematical background, severely limiting their applicability by untrained personnel (Barker-Plummer et al. 2008; Barker-Plummer, Cox, and Dale 2009; Mpagouli et al. 2007). In these contexts, a system capable of automatically translating between NL, like English, and logic would be of great help. Such a system would not only make formal methods more accessible but also find important applications in emerging domains, like AI Safety. For instance, it could enable the real-time monitoring of Large Language Models (LLMs)' behavior through automatic translation into logic of their outputs and internal reasoning (e.g., Chain-of-Thought monitoring (Korbak et al. 2025)), offering a key pathway to integrating symbolic and subsymbolic AI, or it could assist in the creation and in the online updating of formal world models and safety specifications (Dalrymple et al. 2024). Translating FOL into human-readable English is a relatively straightforward process, feasible even with a simple, attribute grammar-based formula parsing task (Ranta 2011). On the opposite, the problem of natural language to FOL (NL-FOL) translation, which can be considered as a subtask of autoformalization (Szegedy 2020) and semantic parsing (Wilks and Fass 1992), has proven to be a persistent challenge for both human and artificial intelligence (Barker-Plummer, Cox, and Dale 2009; Singh, Aggarwal, and Krishnamurthy 2020). Despite recent advancements in natural language processing, the literature presents conflicting results on the ability of even state-of-the-art LLMs to achieve accurate and robust NL-FOL translation. Some works, like Yang et al. (2024), indicate good performance, while others, such as Han et al. (2024), show the opposite. These discrepancies arise because, often, studies perform evaluations on their own datasets with non-theoretically grounded protocols, hindering comparisons.

In this work, we aim to bring clarity to this situation, through three key contributions:

- We provide a critical assessment of the two most recent and comprehensive works which evaluate LLM performance in NL-FOL translation, highlighting issues with

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

both the datasets and test protocols used, which may misrepresent LLMs' actual capabilities.

- Based on our critique, we propose a novel, general, and theoretically grounded benchmarking strategy for NL-FOL translation. Departing from previous protocols, our approach restructures the formula generation task into two distinct phases and introduces additional subtasks designed to distinguish genuine logical understanding from superficial pattern recognition, memorization, and dataset contamination.
- Finally, applying our benchmarking strategy, we show that the dialogue-oriented LLMs OpenAI's GPT-4O-MINI, O3-MINI and Qwen models QWEN3-8B, QWEN3-30B-A3B achieve strong NL-FOL translation performance, with an authentic grasp of sentence-level logic. On the contrary, the state-of-the-art, according to the MTEB leaderboard (Huggingface 2025), embedding-centric models QWEN3-EMBEDDING-8B and GEMINI-EMBEDDING-001 perform markedly worse.

The remainder of the paper is structured as follows. We first provide a summary of relevant literature, focusing on NL-FOL translation while also considering the more general task of autoformalization. We then present our contributions: a principled critique of the most recent and relevant evaluation protocols for NL-FOL translation, the detailed description of our novel benchmarking strategy, and the experimental results obtained from its application. Finally, we conclude and outline future research directions. In Appendix A we provide background on FOL.

Related Work

In the literature (Szegedy 2020; Wu et al. 2022), *autoformalization* denotes the automatic translation of natural language statements into a given formalism. The focus has often been on formalizing mathematical proofs (Wu et al. 2022; Azerbayev et al. 2023; Cunningham, Bunescu, and Juedes 2023) so that they could be checked by interactive theorem provers such as Lean (De Moura et al. 2015), Isabelle (Paulson 1994), and Coq (Bertot and Castéran 2004).

More recently, researchers have moved beyond the mathematical domain, motivated by the opportunity to use autoformalization as a bridge to combine symbolic and subsymbolic AI methodologies, or by the need to formally guarantee the safety of AI systems (Szegedy 2020; Seshia, Sadigh, and Sastry 2022; Dalrymple et al. 2024). Among the considered formalisms are Structured Query Language (SQL) (Kanburoglu and Tek 2024), Linear Temporal Logic (LTL) (Brunello, Montanari, and Reynolds 2019; Mendoza et al. 2024), and First-Order Logic (FOL).

Regarding FOL, the translation from NL to this formalism has been addressed through various approaches, from rule-based methods (Abzianidze 2017; Bos and Markert 2005; Zettlemoyer and Collins 2005), to subsymbolic ones (Levkovskyi and Li 2021; Cao et al. 2019) and the use of language models like BERT and RoBERTa (Tian et al. 2021). With the advent of LLMs, several studies have investigated how to improve the initially modest performance of these models on the NL-FOL translation task (Lu, Liu et al. 2022; Yang et al. 2024; Thatikonda et al. 2024). Recently, such a task has also been used instrumentally to create pipeline for enhancing general-purpose reasoning (Pan, Albalak et al. 2023; Olausson et al. 2023; Ye et al. 2023) or automatic logical fallacies detection (Lalwani et al. 2025). The two most recent and comprehensive studies on NL-FOL translation are Han et al. (2024) and Yang et al. (2024); for brevity, we refer to them as FOLIO and MALLS, respectively. Each introduces its own dataset and benchmarking protocol, yet they provide contradictory evidence about LLMs' NL-FOL translation performance: FOLIO estimates roughly 52% accuracy for GPT-4 in a zero-shot setting (and 62% for a few-shot), while, for the same model, MALLS seems to claim a much higher capability (around 80% for their defined *LE score* over their dataset). In the following, we examine FOLIO and MALLS, highlighting methodological shortcomings that may have led to a misinterpretation of LLMs' logical competence in the NL-FOL translation task, and to the observed performance discrepancies.

Limitations of Current Evaluation Protocols

To provide a precise analysis, we propose to first decompose the NL-FOL translation task into the following two steps:

Ontology Extraction (OE): identify an appropriate logical signature (predicates, functions, constants), and associate to each logical symbol its intended semantic meaning.

Logical Translation (LT): given the signature, define a FOL formula that captures the meaning of a NL sentence.

This is crucial for multiple reasons: (i) keeping OE and LT separate allows to determine whether a model fails at extracting a signature or at translating logic; (ii) with a fixed, provided signature, LT verification is straightforward, e.g., a SMT solver (Barrett and Tinelli 2018) can compare the generated formula to the ground truth, but the same becomes much harder if each formula has its own symbols; (iii) separating OE and LT helps identify techniques that work for one subtask but not the other, improving overall NL-FOL translation; (iv) in many domains, experts can predefine an ontology once, so the system need only to perform logical translation over that fixed vocabulary; (v) providing an ontology is also useful in all domains which require strict adherence to a given syntax.

By contrast, FOLIO and MALLS collapse OE and LT into a single task, an approach that introduces several evaluation problems, which we examine in the following.

Analysis of FOLIO (Han et al. 2024)

In FOLIO, the formalization task is considered alongside Natural Language Inference (NLI) to assess the logical understanding of models. The work comes with its own, expert-written dataset composed of 487 *stories*, where a story is a list of NL *premises* $p_1, \ldots, p_n$ and their respective FOL translations $\varphi_1, \ldots, \varphi_n$. Each story may repeat multiple (on average, three) times in the dataset, associated with a different *conclusion* c. The latter is a phrase, paired with a formula ψ, and labeled with *true*, *false*, or *unknown*, depending on whether c is implied by the story, in contradiction with it, or cannot be resolved from the premises.

Formula	Truth values							
$\text{Country} - Dummy$	0	0	0	0	1	1	1	1
$\text{InEU} - \text{CountryInEU}$	0	0	1	1	0	0	1	1
$\text{EUCountry} - \text{EUCountry}$	0	1	0	1	0	1	0	1
φ	1	1	1	1	1	1	0	1
φ'	1	1	0	1	1	1	0	1

Table 1: Example of LE score calculation in MALLS.

For the NLI task, the model, given a story in NL, must determine the label of its (also NL) conclusion.

For the formalization task, the model, given $p_1, \ldots, p_n$ and c, must predict the logical translations $\varphi'_1, \ldots, \varphi'_n$ of the premises and the translation ψ' of the conclusion: the formalization is considered correct if, depending on the label of c, $\varphi'_1, \ldots, \varphi'_n$ imply ψ', or its negation, or neither. Note that this evaluation approach does not rely on the ground truth formulas $\varphi_1, \ldots, \varphi_n, \psi$ of the premises and conclusion, and it works also with translations that use a different signature than the ground truth.[1] However, it is a poor proxy for translation quality: it assigns the same score whether every sentence is mistranslated or only one is wrong, since in both situations the logical deduction step may fail.

Analysis of MALLS (Yang et al. 2024)

Unlike FOLIO, which judges translations via an SMT solver's verdict, MALLS evaluates correctness by comparing each candidate translation to the respective ground truth formula. Experiments are based on a synthetic dataset, generated by using OpenAI's GPT-4, and consisting of 28000 pairs of real-world NL statements and corresponding FOL translations. Of these, 1000 have been human-verified and serve as the test set on which the paper's results are based.

The evaluation pipeline of MALLS attempts to estimate the similarity between the LLM prediction φ' and the ground truth translation φ using two scalar values: Logical Equivalence (LE) score, and BLEU score. While this method in principle enables a more precise evaluation, there are critical issues in how scores are attributed to the models' answers.

LE score is fundamentally flawed To explain this metric and its limitations, let us consider the following example, similar to the one reported in the original publication. Suppose we have a ground truth NL sentence $p :=$ "Every country located in EU is an EU country", with ground truth translation $\varphi := \forall x\, \text{Country}(x) \wedge \text{InEU}(x) \rightarrow \text{EUCountry}(x)$; and, let us consider the LLM prediction $\varphi' := \forall y\, \text{CountryInEU}(y) \rightarrow \text{EUCountry}(y)$. The LE score is computed in two stages. First, a one-to-one mapping is established between the predicate symbols of the two formulas. In our example, InEU in φ is paired with CountryInEU in φ' and EUCountry in φ with EUCountry in φ'; any extra predicate in one formula is matched to a fresh dummy symbol in the other (here $\text{Country} - Dummy$). Next, a truth table (see Table 1) is generated. For every possible assignment of truth values to the predicates, φ and φ' are evaluated by means of *propositional logic* rules. The LE score is the fraction of columns in which the two formulas have identical truth values. Table 1 clearly shows an issue with our example: although φ' can be regarded as a correct translation of p, it gets a LE score below 1 ($7/8 = 0.875$).

Such a logical "similarity" computation is fundamentally flawed for two reasons: (i) it treats FOL formulas φ and φ' as propositional logic ones, and assigning fixed truth values to predicate symbols is wrong, since the truth of a predicate may vary also with the object(s) it involves. For example, the LE score given to the pair "$\exists x\ \text{Country}(x) \wedge \text{InEU}(x) \rightarrow \text{EUCountry}(x)$" and "$\forall x\ \text{Country}(x) \wedge \text{InEU}(x) \rightarrow \text{EUCountry}(x)$" would be 1, despite their different meanings; (ii) the columns of Table 1 can be viewed as world models, yet some of them are semantically non meaningful: any column that assigns 1 to $\text{EUCountry} - \text{EUCountry}$ while giving 0 to $\text{Country} - Dummy$ violates the obvious constraint that every EU country is a country. The LE score treats the truth values of predicate symbols as independent, while, often they are interdependent. No simple, automatic fix seems to exist for this problem; for a related discussion in another setting, see (Paris and Vencovská 2015).

BLEU score does not capture the logical semantic Given two texts, BLEU (Papineni, Roukos et al. 2002) computes the geometric mean of the modified n-gram precisions for $n = 1, \ldots, 4$, multiplied by a brevity penalty. In MALLS it is applied to the formula strings φ and φ'. However, because BLEU was designed for NL, it performs poorly on logical formulas: two expressions that differ only by a systematic predicate renaming, or that are logically but not syntactically equivalent, receive a low score. In the absence of tokenisation guidelines from the MALLS authors, if we treat every symbol in our example's φ and φ' as a separate token, we obtain an overall raw BLEU score of 0.18.

MALLS Dataset contains ground truth errors due to misspecified guidelines used in the annotation process Beyond flaws in the evaluation metrics, the pipeline used to human-verify the 1000 test instances also presents problems. The guidelines provided to annotators (Appendix A.3 of (Yang et al. 2024)) have at least two serious shortcomings.

The first is about the handling of quantifiers. The guidelines state that $\forall$ (resp., $\exists$) should be used when the NL sentence explicitly contains "Every A ..." or "For all A ..." (resp., "Some A ..." or "There exists A ..."), but "if NL does not have these explicit hints, then using either $\forall$ or $\exists$ is fine". This creates errors. Indeed, we find sentences like "A child plays with a toy in a playground" that are treated as universal assertions, and sentences like "Birds can fly, while fish can swim, and elephants can neither fly nor swim" that are translated with existential quantifications despite referring to entire subject classes. Specifically, in the guidelines it is written that, for the sentence "A turtle has a shell and can swim", a correct FOL formalization is $\exists x(\text{Turtle}(x) \wedge \text{Shell}(x) \wedge \text{CanSwim}(x))$, while a more natural formula would be $\forall x(\text{Turtle}(x) \rightarrow \text{Shell}(x) \wedge \text{CanSwim}(x))$.

The second issue is about logical connectives interchangeability. The guidelines claim that $\rightarrow, \wedge$, and $\leftrightarrow$ are

[1] The ground truth formulas are in the dataset only to enable verification that the conclusion labels provided are correct.

$$
\begin{aligned}
p &= \text{Tom likes every cat that is red} \\
\sigma &= \{Tom, Jane, like(x,y), own(x,y), cat(x), dog(x), red(x)\} \\
\gamma &= \{like(x,y) : x \text{ likes } y, cat(x) : x \text{ is a cat}, \ldots\} \\
\varphi &= \forall x((cat(x) \wedge red(x)) \rightarrow like(Tom, x)) \\
T(\varphi) &= \text{For any } x, \text{ if } x \text{ is a cat and } x \text{ is red, then Tom likes } x \\
\varphi_1 &= \forall x((cat(x) \wedge red(x)) \wedge like(Tom, x)) \\
\varphi_{eq} &= \forall x(\neg(cat(x) \wedge red(x)) \vee like(Tom, x)) \\
(\neg\varphi)_{\text{nnf}} &= \exists x(cat(x) \wedge red(x) \wedge \neg like(Tom, x))
\end{aligned}
$$

Figure 1: Example of instantiation of phrases, signatures, and formulas as used in the benchmarking strategy tasks.

sometimes interchangeable. For instance, they report that also the following formulas are correct translations for the sentence above:

$$\exists x(\text{Turtle}(x) \rightarrow \text{Shell}(x) \wedge \text{CanSwim}(x))$$
$$\exists x(\text{Turtle}(x) \leftrightarrow \text{Shell}(x) \wedge \text{CanSwim}(x))$$

This is theoretically wrong and unacceptable for rigorous logical capability assessment and critical autoformalization applications. The first formula means that "there is an entity that if it is a turtle, then it has a shell and it can swim", which can be true even if not all turtles have a shell or can swim, and also in worlds in which there are no turtles; the second formula means that "there is an entity that is a turtle if and only if it has a shell and it can swim".

Our Novel Benchmarking Strategy

Building upon the identified limitations of the existing evaluation protocols, in this section we introduce our novel benchmarking strategy to assess the proficiency of LLMs in the NL-FOL translation task. The approach is outlined here in general terms, and the design choices are open to customization; concrete applications on specific models and datasets are provided in the experimental section.

Our method relies on a dataset $\mathcal{D}$ composed of triplets (p, φ, Ω), where p represents a natural language utterance, φ is its corresponding FOL formula, and Ω is an ontology, i.e., a pair (σ, γ), where σ is the FOL signature (predicates, functions, constants) within which the formula φ is written, and γ is a natural language glossary that specifies the intended meaning of each symbol in σ. For an intuitive example of instantiation of phrases, ontologies and formulas as used in the following, please refer to Figure 1.

Given the dataset $\mathcal{D}$, to comprehensively evaluate LLM capabilities, we define three core tasks, referred to as: *logical translation*, *most similar*, and *ranking*.

Logical translation In this task, for each triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, the LLM is given a prompt containing p and Ω and must generate, using (a subset of) the symbols in σ and their descriptions in γ, a formula φ' that captures the meaning of p. We then automatically verify that φ' is logically equivalent to the reference formula φ using an SMT solver. The task succeeds if φ' is found to be equivalent to φ.

Providing Ω marks a deliberate departure from earlier studies, which asked the model to derive φ' from p alone. As discussed previously, by supplying the ontology up front, we disentangle the two stages of the NL-FOL translation pipeline, *ontology extraction* and *logical translation*, allowing us to isolate the latter and minimize confounding factors.

Observe that this task remains subject to other limitations: an LLM might have memorized the (public) dataset $\mathcal{D}$, or it could rely for the translation on purely syntactic heuristics, mechanically transforming p into φ', without genuinely comprehending the underlying logical semantics. To address these issues, we designed two additional tasks that are fundamentally different and allow us to probe the model from different angles: *most similar* and *ranking*.

Most similar Given a triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, we produce up to k random perturbations of φ.[2] Each perturbation is obtained by applying a single elementary edit to φ, which may include: replacing one Boolean connective (e.g., $\wedge$ with $\vee$); switching a quantifier (e.g., $\forall$ with $\exists$); inserting or removing a negation in front of a literal (in Appendix B, we discuss why we considered this kind of modifications).

Putting together the original formula with its perturbations, we obtain the set $\mathcal{F}_{ms} = \{\varphi, \varphi_1, \ldots, \varphi_k\}$. The model, given a prompt containing p and the (shuffled) set $\mathcal{F}_{ms}$, must select the formula whose meaning is closest to p; we call this the *FOL most similar* task. The task succeeds if φ is selected.

We also consider a parallel task in natural language. Let $T()$ be a translation function that converts a FOL formula into NL by substituting logical symbols and predicates with their glosses (details are in Appendix C). The model now chooses the sentence that best matches p from (the shuffled) $\mathcal{T}_{ms} = \{T(\varphi), T(\varphi_1), \ldots, T(\varphi_k)\}$. We refer to this as the *NL most similar* task. The task succeeds if $T(\varphi)$ is selected.

Ranking The *most similar* task merely asks the model to pick the candidate whose meaning is closest to the original utterance. Here we shift to a finer-grained setting. Given an integer k, for each triplet $(p, \varphi, \Omega = (\sigma, \gamma))$ we build the set $\mathcal{F}_r = \{\varphi,\ \varphi_1, \ldots, \varphi_k,\ \neg\varphi,\ (\neg\varphi)_{\text{nnf}}, \varphi_{\text{eq}}\}$, where: $\varphi_1, \ldots, \varphi_k$ are k logical perturbations of φ, as in the *most similar* task; $\neg\varphi$ is the outright negation of φ; $(\neg\varphi)_{\text{nnf}}$ is $\neg\varphi$ rewritten in *negation normal form*, with all negations pushed down to the predicate level, producing a formula that is syntactically distant from the original; and, φ_{eq} is a formula logically equivalent to φ. To create φ_{eq} we first select a subformula of φ at random and then apply one randomly chosen, applicable transformation among: DeMorgan's Laws $\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ or $\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$; Double Negation Law $\alpha \equiv \neg((\neg\alpha)_{\text{nnf}})$; Commutativity Law $\alpha \wedge \beta \equiv \beta \wedge \alpha$ or $\alpha \vee \beta \equiv \beta \vee \alpha$; Distributivity Law $(\alpha \wedge (\beta \vee \gamma) \equiv (\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$ or $\alpha \vee (\beta \wedge \gamma) \equiv (\alpha \vee \beta) \wedge (\alpha \vee \gamma))$; and, Implication Expansion $\alpha \rightarrow \beta \equiv \neg\alpha \vee \beta$.

We now evaluate the model on two ranking subtasks:

- *FOL ranking*: given a prompt containing p and the (shuffled) set of formulas $\mathcal{F}_r$, the model must order them from the most to the least semantically similar to the utterance p. The task succeeds if: (i) φ and φ_{eq} are placed at the top positions of the rank; and (ii) $\neg\varphi$, and $(\neg\varphi)_{\text{nnf}}$ are placed

[2] Atomic propositions, for instance, admit fewer modifications.

at the bottom positions of the rank. Elements within the top and bottom may appear in any order.

- *NL ranking*: each formula is first rendered in English via the translation function $T()$, producing $\mathcal{T}_r = \{T(\varphi), T(\varphi_1), \ldots, T(\varphi_k), T(\neg\varphi), T((\neg\varphi)_{\text{nnf}}), T(\varphi_{\text{eq}})\}$ The model then ranks these (shuffled) sentences by their semantic proximity to p. The task succeeds with conditions analogous to the *FOL ranking* ones.

Observe how the *most similar* and *ranking* tasks are far less vulnerable than the *logical translation* one to typical LLM evaluation issues such as memorisation and dataset leakage, since their candidate sets $\mathcal{F}_{ms}, \mathcal{T}_{ms}, \mathcal{F}_r, \mathcal{T}_r$ are generated on the fly and are not confined to the formulas and phrases in $\mathcal{D}$. In addition, by requiring the model to select or order candidates by semantic proximity, these tasks probe finer-grained logical understanding rather than surface pattern matching; indeed, the closest match can be syntactically distant from the original, thereby discouraging reliance on purely syntactic or mechanical heuristics.

Finally, although our benchmarking strategy has been presented with dialogue-oriented LLMs in mind, it can be adapted to embedding-centric models. In that case we retain only the *most similar* and *ranking* tasks: the model generates an embedding for each candidate utterance or formula, and semantic proximity to the embedding of p is measured with an embedding-distance metric, in our case, cosine similarity.

Some words on Ontology Extraction

In this work, we focus exclusively on the second step of our proposed NL-FOL evaluation pipeline, i.e., Logical Translation, and we assume access to a predefined and correct ontology Ω. Ontology learning is a longstanding and extensively studied challenge in knowledge representation (Armary et al. 2025). Nevertheless, unified evaluation metrics and standardized benchmarks for this task remain limited (Du et al. 2024). Formulating direct approaches to OE evaluation warrants a dedicated and rigorous analysis comparable to the one we provide for LT in this paper. Alternatively, because the quality of an ontology depends heavily on both its intended application and its underlying formalism, a common evaluation strategy is to assess it indirectly by measuring its effectiveness within a relevant proxy task (Dellschaft and Staab 2006; Du et al. 2024). In our setting, such an indirect evaluation would still require first assessing a model's capabilities in Logical Translation.

For these reasons, we do not address OE here and leave this subtask to future work. Completing such a component will provide the full implementation of the proposed NL-FOL evaluation pipeline.

Experiments

In this section we first introduce the concrete models and datasets over which we applied our benchmarking strategy. Then, we outline the experimental workflow. See Appendix D for details about our computing infrastructure.

Considered Models

We evaluate six models: Four dialogue-oriented models: GPT-4O-MINI (OpenAI 2024), O3-MINI (OpenAI 2025b) QWEN3-8B, and QWEN3-30B-A3B (hereafter QWEN3-30B) (Yang et al. 2025); and two embedding-centric models, QWEN3-EMBEDDING-8B (hereafter QWEN-EMB) (Zhang et al. 2025) and GEMINI-EMBEDDING-001 (hereafter GEMINI-EMB) (Lee et al. 2025).

We selected the dialogue-oriented models to cover two points on the cost-capacity trade-off curve: O3-MINI offers in principle the highest capacity while remaining affordable for large-scale API experiments (OpenAI 2025a), whereas GPT-4O-MINI trades some performance for a lower usage cost; similar remarks hold for the two models QWEN3-8B and QWEN3-30B. Note that, for both Qwen models, the results shown in the paper refer to their *thinking* mode (Yang et al. 2025); an analysis of the performances of their *not-thinking* counterparts is provided in Appendix E.

As for the embedding-centric models, they are purpose-built for producing high-quality sentence embeddings, they are widely used in retrieval, semantic-similarity, and reranking pipelines, and they represent the state-of-the-art in that paradigm as of July 2025 (Huggingface 2025).

The selected Qwen models enable a more detailed comparative analysis between dialogue-oriented and embedding-centric architectures within the same model family, as they are built upon a shared foundational model. Specifically, QWEN3-EMBEDDING-8B and QWEN3-8B are both deriving (i.e., different finetunes) from QWEN3-8B-BASE. We did not perform an equivalent analysis for the Gemini and OpenAI families, due to the lack of publicly documented evidence clearly identifying which, if any, dialogue-based models correspond to the embedding-oriented models considered in this study, and the high computational costs such an investigation would entail.

To prevent data leakage, server-side training was disabled for OpenAI's models via the provided data controls settings. All Qwen models were run entirely on local hardware, and Google Cloud explicitly states that customer data are not used for training models (Google Cloud 2023).

Considered Datasets

We make use of two datasets, $\mathcal{D}_{\text{Stanford}}$ and $\mathcal{D}_{\text{FOLIO}}$. Here, we describe their content and the preprocessing applied to each.

$\mathcal{D}_{\text{Stanford}}$ The dataset is a 2001–2010 subset of the *Grade Grinder Corpus Release 1.0* (Barker-Plummer, Cox, and Dale 2011). Its files have never been made public and were kindly shared with us by the original authors. The corpus records students' submissions, which can be both correct and incorrect, entered via the Grade Grinder tutoring platform while they worked through exercises in *Language, Proof and Logic* (Barker-Plummer, Barwise, and Etchemendy 2011). In each exercise, a natural-language description of a scene in Tarski's World (Barker-Plummer et al. 2007) is provided, and students must supply a FOL formula that expresses the same content. Our extract consists of 159 high-quality instances. In the notation of our benchmark, each instance is composed of an utterance p paired

with its validated FOL translation φ. We manually defined an ontology $\Omega = (\sigma, \gamma)$ which applies indistinctly to every text-formula item. Appendix F provides the full ontology and presents handcrafted instances that are inspired by the dataset but do not reproduce any of its private content.

$\mathcal{D}_{\text{FOLIO}}$ The dataset released with (Han et al. 2024) contains 487 *stories*, split 70%-15%-15% into training, validation, and test sets. As we mentioned, a story comprises several *premises* pairs (a premise is an utterance p with its FOL translation φ) and, on average, is associated to three *conclusions*. For the purposes of our study we use only the training portion (339 stories), focusing solely on the premises and ignoring the conclusions. For each story we manually defined a distinct ontology $\Omega = (\sigma, \gamma)$, and attached it to all (p, φ) premise pairs in that story. Flattening the premises across stories then yielded 1667 (p, φ, Ω) triples. As a final step, to keep the dataset comparable with $\mathcal{D}_{\text{Stanford}}$, whose formulas never employ the XOR operator, we discarded any triplet whose φ contained XOR, leaving us with 1565 instances.

We excluded the MALLS (Yang et al. 2024) test set from our evaluation because, as noted earlier, its validation pipeline has serious flaws. We also omitted the synthetic LogicNLI dataset (Tian et al. 2021) since, according to the FOLIO authors (Han et al. 2024), its FOL translations exhibit limited complexity and syntactical variation. We are unaware of other well-founded, non-synthetic NL-FOL datasets, challenging enough to serve as a benchmark.

Experimental Workflow

We conducted separate experimental workflows for the dialogue-oriented and embedding-centric models, which were applied on both $\mathcal{D}_{\text{Stanford}}$ and $\mathcal{D}_{\text{FOLIO}}$ datasets. For the dialogue models we followed OpenAI's reproducibility guidelines (OpenAI 2025c). Each prompt was issued five times, changing only the seed parameter in the API call while keeping all other settings fixed.[3] For the embedding models, since they produce deterministic vectors, each query was run once. Appendix G lists all hyper-parameter values and provides the prompts used with the models. All code required to replicate our results is publicly available (see Appendix K).

Dialogue-oriented Models

We used zero-shot, Automatic Chain of Thought Prompting (Zhang et al. 2023) with all the models. Details of the prompts are provided in Appendix H.

Logical translation For each triplet $(p, \varphi, \Omega = (\sigma, \gamma))$ in a dataset, we built a system prompt that: defines the NL-FOL translation task, supplies the ontology Ω, and specifies the required output format. The accompanying user prompt contains only the utterance p. We parsed the model's reply with a Python script to extract the candidate formula φ' and checked its logical equivalence to the reference formula φ with the Z3 solver (De Moura and Bjørner 2008). If the formulas were found to be equivalent we assigned a score of 1, otherwise 0. Overall performance was computed as the average of these scores over the dataset.

Most similar Considering *FOL most similar*, given a triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, we prepared a system prompt describing the task to be performed in general terms; in the user prompt we provided the reference p and the (randomly shuffled) set $\mathcal{F}_{ms}$ containing the original FOL formula and k=8 perturbations. The output of the model was constrained, by means of the Structured Outputs feature of OpenAI API, to be an integer which, according to the model, is the position of the formula that most closely represents the meaning of p in the set. If the model selected the position in which φ occurs, then it scores 1, otherwise 0. Overall performance was computed as the average of these scores over the dataset.

Ranking Considering the *FOL Ranking*, given a triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, in a system prompt we described the task to be performed in general terms; in the user prompt we provided the reference p and the (randomly shuffled) set $\mathcal{F}_r$ containing the original FOL formula φ, k=3 perturbations, an equivalent version of φ, and two versions $\neg\varphi, (\neg\varphi)_{\text{nnf}}$ of its negation. The model's output was again constrained, in this case to be a list of integers which, according to the model, should represent the ranking of the elements in $\mathcal{F}_r$, ordered from the formula that shares the most similar meaning with p down to the least similar. We then associate the following scores to the ranking, following the criteria defined in our benchmarking strategy:

- we assign 1 if in the ranking the first two positions are the ones of φ and φ_{eq} (no matter the order) and 0 otherwise: we define the *Ranking-Equivalence score* as the average of these values over the dataset.
- we assign 1 if in the ranking the last two positions are the ones of $\neg\varphi$ and $(\neg\varphi)_{\text{nnf}}$ (no matter the order) and 0 otherwise: we define the *Ranking-Negation score* as the average of these values over the dataset.
- we assign 1 if both the conditions above are met and 0 otherwise: we define the *Ranking-Both score* as the average of these values over the dataset.

Embedding-centric Models

As previously discussed, our evaluation strategy extends also to embedding models. We start by describing the configuration of the models employed in our experiments.

For QWEN-EMB, other than the "classical" embedding generation, where a single input q is encoded, the model can also accept a user-supplied instruction I that is prepended to the input q to tailor the embedding to a specific task (Zhang et al. 2025). We test both settings: embedding q alone (denoted QWEN-EMB-PLAIN), and embedding the concatenation of I and q (denoted QWEN-EMB-INST).

Concerning GEMINI-EMB, it is possible to task the model to generate embeddings optimized for a specific task, selecting one among a list of predefined task types. We chose `SEMANTIC_SIMILARITY` and embedded only the input q.

We now describe the *most similar* and *ranking* tasks (recall that *logical translation* is not performed here).

[3] Although seeding reduces variance, OpenAI notes that some residual nondeterminism may remain.

	GPT-4O-MINI	O3-MINI	QWEN3-8B	QWEN3-30B
$\mathcal{D}_{\text{Stanford}}$	.84±.03	.94±.00	.84±.01	.85±.01
$\mathcal{D}_{\text{FOLIO}}$	.73±.01	.80±.00	.72±.00	.74±.01

Table 2: Results of the dialogue-oriented models, logical translation. Average ± standard deviation over 5 repetitions.

Task	Dataset	GPT-4O-MINI		O3-MINI		QWEN3-8B		QWEN3-30B	
		NL	FOL	NL	FOL	NL	FOL	NL	FOL
Most similar	$\mathcal{D}_{\text{Stanford}}$	.89±.02	.88±.02	.99±.00	1.0±.00	.95±.01	.96±.00	.97±.01	.98±.01
	$\mathcal{D}_{\text{FOLIO}}$	.91±.00	.89±.00	.95±.00	.95±.00	.93±.00	.92±.00	.93±.01	.94±.00
Ranking-Eq.	$\mathcal{D}_{\text{Stanford}}$	.50±.03	.57±.01	.93±.02	.98±.00	.71±.03	.79±.03	.80±.01	.91±.01
	$\mathcal{D}_{\text{FOLIO}}$	.52±.01	.64±.01	.91±.01	.94±.00	.77±.02	.85±.00	.86±.01	.90±.00
Ranking-Neg.	$\mathcal{D}_{\text{Stanford}}$	.49±.02	.51±.02	.88±.02	.91±.01	.52±.05	.56±.01	.63±.01	.62±.03
	$\mathcal{D}_{\text{FOLIO}}$	.62±.01	.62±.01	.84±.00	.85±.01	.62±.02	.64±.01	.70±.00	.72±.01
Ranking-Both	$\mathcal{D}_{\text{Stanford}}$	.28±.02	.32±.02	.82±.02	.89±.02	.39±.05	.46±.02	.53±.02	.57±.04
	$\mathcal{D}_{\text{FOLIO}}$	.39±.01	.44±.01	.78±.01	.82±.01	.51±.02	.57±.01	.62±.01	.66±.01

Table 3: Results on Most similar and Ranking for dialogue-oriented models: average ± standard deviation over 5 reps.

Most Similar Considering *FOL Most Similar*, given a triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, and having the set $F_{ms} = \{\varphi, \varphi_1, \ldots, \varphi_8\}$, we computed the embeddings $v_p, e, e_1, \ldots, e_8$ of p, $\varphi, \varphi_1, \ldots, \varphi_8$ respectively (possibly augmented with the task instruction for QWEN-EMB-INST). We assigned a score of 1 if the embedding e (of φ) exhibited the highest cosine similarity to v_p among all the other candidate embeddings, and 0 otherwise. Overall performance was computed as the average of these scores over the dataset.

Ranking Considering *FOL Ranking*, given the triplet $(p, \varphi, \Omega = (\sigma, \gamma))$, and having the set $\mathcal{F}_r = \{\varphi, \varphi_1, \ldots, \varphi_3, \neg\varphi, (\neg\varphi)_{\text{nnf}}, \varphi_{\text{eq}}\}$, we computed the embeddings $v_p, e, e_1, \ldots, e_3, e_{\text{neg1}}, e_{\text{neg2}}, e_{\text{eq}}$ of $p, \varphi, \varphi_1, \ldots, \varphi_3, \neg\varphi, (\neg\varphi)_{\text{nnf}}, \varphi_{\text{eq}}$ respectively (possibly augmented with the task instruction for QWEN-EMB-INST). Then, the Ranking-Equivalence score was determined as in the dialogue-oriented case, considering that e and e_{eq} should be the two vectors most similar to v_p among all the candidates; for the Ranking-Negation score, e_{neg1}, and e_{neg2} should be the two least similar; for the Ranking-Both score, both the conditions outlined above should be satisfied.

For the *NL most similar* and the *NL ranking* tasks with both dialogue-oriented and embedding-centric models, the only difference is that we consider the sets $\mathcal{T}_{ms}$ and $\mathcal{T}_r$ in place of $\mathcal{F}_{ms}$ and $\mathcal{F}_r$, respectively.

Results and Discussion

Here, we present the results obtained with our benchmarking strategy, starting with the logical translation task, where only dialogue-oriented models are involved. Because we performed no prompt engineering (e.g., we did not use any few-shot exemplars) the reported scores should be viewed as lower bounds on the models' capabilities. A thorough analysis of prompting strategies, which is beyond the scope of the present paper, is left for future work.

Logical Translation Table 2 reports the overall performance of the models evaluated with our metric. All the models perform well on the considered datasets. The weakest result is the .72 achieved by QWEN3-8B on $\mathcal{D}_{\text{FOLIO}}$, meaning

Task	Dataset	QWEN-EMB-PLAIN		QWEN-EMB-INST		GEMINI-EMB	
		NL	FOL	NL	FOL	NL	FOL
Most similar	$\mathcal{D}_{\text{Stanford}}$	.62	.46	.75	.55	.69	.52
	$\mathcal{D}_{\text{FOLIO}}$	.58	.44	.82	.76	.65	.49
Ranking-Eq.	$\mathcal{D}_{\text{Stanford}}$	.30	.21	.30	.31	.30	.29
	$\mathcal{D}_{\text{FOLIO}}$	.40	.21	.42	.41	.38	.35
Ranking-Neg.	$\mathcal{D}_{\text{Stanford}}$	.51	.36	.70	.71	.67	.64
	$\mathcal{D}_{\text{FOLIO}}$	.58	.44	.76	.86	.69	.71
Ranking-Both	$\mathcal{D}_{\text{Stanford}}$	.18	.09	.25	.23	.22	.20
	$\mathcal{D}_{\text{FOLIO}}$	.30	.13	.34	.37	.34	.32

Table 4: Results on Most similar and Ranking for embedding-centric models.

that, nevertheless, in 72% of the cases the model translates a NL sentence into a FOL formula equivalent to the reference. Aligning with our expectations, O3-MINI consistently outperforms GPT-4O-MINI by a large margin; the same doesn't hold when considering QWEN3-30B and QWEN3-8B, despite the difference in size. Interestingly, all models find $\mathcal{D}_{\text{Stanford}}$ "easier" than $\mathcal{D}_{\text{FOLIO}}$ (.11–.14 difference).

We find no clear evidence that this discrepancy arises from a lower intrinsic difficulty of $\mathcal{D}_{\text{Stanford}}$, which includes formulas that challenge even advanced undergraduates (see, e.g., Brunello et al. (2025) as well as Appendix F for more details). A more plausible explanation for the observed performance lies in the quality of the dataset. As reported by Olausson et al. (2023), approximately 11% of the original validation set of $\mathcal{D}_{\text{FOLIO}}$ contains errors. Since our experiments relied on the training portion of $\mathcal{D}_{\text{FOLIO}}$, and the train–validation–test split was performed randomly, a comparable proportion of errors is likely present in our data. Given the prohibitive length of the dataset, a full manual inspection was infeasible; therefore, we focused on instances where both models GPT-4O-MINI and O3-MINI consistently produced outputs judged incorrect across all random seeds. Among the 302 such instances examined, 93 were found to have incorrect ground-truth labels, indicating a conservative lower bound of roughly 6% mistakes in the whole training dataset. Here an example (story-id 8, instance no. 3) where the dataset is wrong:

NL: "Some musicians love music."
FOL (dataset): $\exists x(\text{Musician}(x) \rightarrow \text{Love}(x, \text{music}))$
O3-MINI: $\exists x(\text{Musician}(x) \wedge \text{Love}(x, \text{music}))$

A more detailed analysis of these cases is provided in Appendix I.

Most similar We now turn to the *most similar* task, restricting to dialogue-oriented models. Here, as reported in Table 3, the performance gap between $\mathcal{D}_{\text{FOLIO}}$ and $\mathcal{D}_{\text{Stanford}}$ is smaller. This is reasonable, because the errors in $\mathcal{D}_{\text{FOLIO}}$ are less likely to influence the task outcome: a model may still select the correct formula φ as the most similar to sentence p if the perturbations in $\mathcal{F}_{ms}$ differ substantially in meaning. Errors may nevertheless still be affecting the evaluation, given the drop in O3-MINI performance from 1.00 on $\mathcal{D}_{\text{Stanford}}$ to 0.95 on $\mathcal{D}_{\text{FOLIO}}$. In future work, we plan to more thoroughly investigate this aspect. Notably, the dialogue-oriented models achieve the same results on the *most similar* task with both its FOL and NL variants; the absence of a performance gap when dealing with formulas versus NL

utterances may suggest that the models can reliably interpret the semantic content of FOL expressions.

Ranking The *ranking* task appears generally more challenging than *most similar*. A closer analysis reveals that, in fact, the latter is for the most part a subset of the former. For example, with GPT-4O-MINI on $\mathcal{D}_{\text{FOLIO}}$ (FOL), the model succeeds in *most similar* but fails in *ranking-equivalence* in 31% of the instances, whereas the opposite occurs in only 4% of the cases. This trend holds consistently across models and datasets, as detailed in Appendix J.

The performance gap between smaller and bigger models in the same family is significant: the latter ones consistently achieves markedly higher scores. A second pattern also emerges: models perform better on the FOL variant than on the NL one. The *most similar* task was not sufficiently challenging to expose such differences, remarking the importance of carefully designing evaluation tasks with enough discriminative power to probe the models' actual logical capabilities. Interestingly, *Ranking-Equivalence* proves generally easier than *Ranking-Negation*. A closer inspection reveals that the models misrank $\neg\varphi$ and $(\neg\varphi)_{\text{nnf}}$ at similar rates, despite the fact that $(\neg\varphi)_{\text{nnf}}$ is, in principle, syntactically more distant from φ (and from $\neg\varphi$ itself). This finding offers additional evidence that the models do not rely on purely syntactic information to perform the ranking task.

Overall, our results indicate that the strong performance in *logic translation* (particularly by O3-MINI) reflects a genuine grasp of the underlying logical content of both NL utterances and FOL formulas, as the models' success on the *most similar* and (for O3-MINI) *ranking* tasks cannot be convincingly explained by shallow pattern matching or incidental exposure to NL–FOL pairs from any leaked corpus.

Embedding-centric models As shown in Table 4, embedding-centric models generally underperform the dialogue-oriented ones, underlining the difference between the state-of-the-art of these two different paradigms of language models. Among the models considered, QWEN-EMB-INST achieves the highest scores, followed by GEMINI-EMB and QWEN-EMB-PLAIN: this suggests that supplying task-specific instructions when generating embeddings is indeed beneficial. Interestingly, the embedding-centric models struggle far more with the tasks' FOL variants than with the NL counterparts (with the partial exception of QWEN-EMB-INST); this is in contrast to what was observed for dialogue-oriented models. Finally, again, the embedding-centric models find *Ranking-Equivalence* easier than *Ranking-Negation*.

Overall, although our study focused on dialogue-oriented models, these preliminary results show that our benchmarking strategy is also effective for embedding-centric ones. Future work could include a deeper analysis, e.g., exploring alternative embedding-similarity metrics.

Empirical evaluation of BLEU and LE score Previously, we examined the theoretical limitations of MALLS' evaluation protocol. We now complement the discussion with a quantitative analysis. Table 5 reports the BLEU and LE scores obtained by the models for the LT task. Since each instance consists of a text–formula pair, correctness can be measured using BLEU, LE, or by checking logical equivalence between the predicted and ground-truth formulas via Z3 (our proposed correctness metric). The table also reports the point-biserial correlations computed between the instance-level BLEU and LE scores and the corresponding binary Z3 equivalence outcomes.

Model	Dataset	BLEU	LE	r_{pb}(BLEU, Our)	r_{pb}(LE, Our)
GPT-4O-MINI	$\mathcal{D}_{\text{Stanford}}$	$.86_{\pm.08}$	$.93_{\pm.01}$	.56	.49
	$\mathcal{D}_{\text{FOLIO}}$	$.85_{\pm.07}$	$.91_{\pm.00}$	.81	.62
O3-MINI	$\mathcal{D}_{\text{Stanford}}$	$.87_{\pm.06}$	$.94_{\pm.00}$	.49	.58
	$\mathcal{D}_{\text{FOLIO}}$	$.88_{\pm.05}$	$.92_{\pm.00}$	.80	.64
QWEN3-8B	$\mathcal{D}_{\text{Stanford}}$	$.84_{\pm.09}$	$.93_{\pm.01}$	.60	.44
	$\mathcal{D}_{\text{FOLIO}}$	$.84_{\pm.01}$	$.92_{\pm.01}$	.80	.57
QWEN3-30B	$\mathcal{D}_{\text{Stanford}}$	$.83_{\pm.10}$	$.93_{\pm.00}$	.66	.45
	$\mathcal{D}_{\text{FOLIO}}$	$.84_{\pm.00}$	$.91_{\pm.01}$	.83	.63

Table 5: Results on LT according to the BLEU and LE score, and their point biserial correlation (r_{pb}) with our metric.

Results show low, unstable, and dataset-dependent correlations (all with p-values $\ll .05$). For example, for O3-MINI, BLEU correlates .49 on $\mathcal{D}_{\text{Stanford}}$ but .80 on $\mathcal{D}_{\text{FOLIO}}$, with similar fluctuations for LE and other models. These inconsistencies highlight the unreliability of traditional metrics compared to our equivalence-based evaluation.

Conclusions and Future Work

We proposed a theoretically grounded benchmarking strategy for LLM-based NL-FOL translation that separates the task into Ontology Extraction and Logical Translation for fine-grained evaluation, while supplementing it with specific subtasks less vulnerable to data leakage: this mitigates the risk to confound genuine logical understanding with pattern matching or memorisation. We applied our benchmarking to dialogue-oriented and embedding-centric models, showing that the former consistently outperform the latter, and that their strong scores arise from semantic understanding rather than the use of superficial syntactical regularities or memorised NL-FOL pairs. Our results remark the value of careful task design when assessing the logical capabilities of LLMs.

Our work opens promising research directions. Other than the ones already mentioned in the paper, a key next step is to determine how difficult extracting an ontology is for current models and to develop techniques that can improve their performance. In addition, progress with NL-FOL translation requires high-quality, carefully curated datasets, possibly built (or validated) with the help of high-performing LLMs and explicitly accounting for polysemy. Finally, we encourage the community to adopt and extend our benchmarking strategy. Experimenting with alternative prompts, adding evaluation subtasks, explicitly handling NL ambiguity, and introducing an even more fine-grained scoring has the potential to reveal new dimensions of LLMs' logical capabilities.

Acknowledgments

MM thanks Cristian Curaba for the valuable discussions and insights that contribute to this work. LG, AM, and NS acknowledge the support from the Interconnected Nord-Est Innovation Ecosystem (iNEST), which received fund-

ing from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.5 – D.D. 1058 23/06/2022, ECS00000043). In addition, Angelo Montanari acknowledges the support from the MUR PNRR project FAIR - Future AI Research (PE00000013) also funded by the European Union Next-GenerationEU. This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

Abzianidze, L. 2017. LangPro: Natural Language Theorem Prover. In *2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 115–120. Association for Computational Linguistics.

Armary, P.; El-Vaigh, C. B.; Labbani Narsis, O.; and Nicolle, C. 2025. Ontology learning towards expressiveness: A survey. *Computer Science Review*, 56: 100693.

Azerbayev, Z.; Piotrowski, B.; Schoelkopf, H.; et al. 2023. ProofNet: Autoformalizing and Formally Proving Undergraduate-Level Mathematics. *CoRR*, abs/2302.12433.

Barker-Plummer, D.; Barwise, J.; and Etchemendy, J. 2011. *Language, Proof, and Logic*. Center for the Study of Language and Information/SRI, 2nd edition. ISBN 1575866323.

Barker-Plummer, D.; Barwise, J.; Etchemendy, J.; and Liu, A. 2007. Tarski's World: Revised and Expanded Edition.

Barker-Plummer, D.; Cox, R.; and Dale, R. 2011. Student translations of natural language into logic: The Grade Grinder corpus release 1.0. In *4th international conference on educational data mining (EDM)*, 51–60.

Barker-Plummer, D.; Cox, R.; Dale, R.; and Etchemendy, J. 2008. An empirical study of errors in translating natural language into logic. In *Annual Meeting of the Cognitive Science Society*, volume 30.

Barker-Plummer, D.; Cox, R. J.; and Dale, R. 2009. Dimensions of Difficulty in Translating Natural Language into First-Order Logic. In *Educational Data Mining*.

Barrett, C. W.; and Tinelli, C. 2018. Satisfiability Modulo Theories. In *Handbook of Model Checking*, 305–343. Springer.

Bertot, Y.; and Castéran, P. 2004. *Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. EATCS Series. Springer. ISBN 978-3-642-05880-6.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds. 2021. *Handbook of Satisfiability - Second Edition*, volume 336 of *Frontiers in Artificial Intelligence and Applications*. IOS Press. ISBN 978-1-64368-160-3.

Bos, J.; and Markert, K. 2005. Recognising Textual Entailment with Robust Logical Inference. In *1st PASCAL Workshop on Machine Learning Challenges (MLCW)*, volume 3944 of *Lecture Notes in Computer Science*, 404–426. Springer.

Brunello, A.; Ferrarese, R.; Geatti, L.; Marzano, E.; Montanari, A.; Saccomanno, N.; et al. 2025. Evaluating LLMs Capabilities at Natural Language to Logic Translation: A Preliminary Investigation. In *7th International Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis (OVERLAY)*, volume 3904, 103–110. CEUR-WS.

Brunello, A.; Geatti, L.; Mignani, M.; Montanari, A.; and Saccomanno, N. 2026. Do LLMs Really Struggle at NL-FOL Translation? Revealing Strengths via a Novel Benchmarking Strategy. In *AAAI-26*. The 40th Annual AAAI Conference on Artificial Intelligence. Accepted for publication.

Brunello, A.; Montanari, A.; and Reynolds, M. 2019. Synthesis of LTL Formulas from Natural Language Texts: State of the Art and Research Directions. In *26th International Symposium on Temporal Representation and Reasoning (TIME)*, volume 147 of *LIPIcs*, 17:1–17:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Cao, R.; Zhu, S.; Liu, C.; Li, J.; and Yu, K. 2019. Semantic Parsing with Dual Learning. In *57th Conference of the Association for Computational Linguistics (ACL)*, 51–64. Association for Computational Linguistics.

Cunningham, G.; Bunescu, R. C.; and Juedes, D. 2023. Towards Autoformalization of Mathematics and Code Correctness: Experiments with Elementary Proofs. *CoRR*, abs/2301.02195.

Dalrymple, D.; Skalse, J.; Bengio, Y.; Russell, S.; et al. 2024. Towards Guaranteed Safe AI: A Framework for Ensuring Robust and Reliable AI Systems. *CoRR*, abs/2405.06624.

De Moura, L.; and Bjørner, N. 2008. Z3: An efficient SMT solver. In *14th International conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 337–340. Springer.

De Moura, L. M.; Kong, S.; Avigad, J.; et al. 2015. The Lean Theorem Prover. In *25th International Conference on Automated Deduction (CADE)*, volume 9195 of *Lecture Notes in Computer Science*, 378–388. Springer.

Dellschaft, K.; and Staab, S. 2006. On How to Perform a Gold Standard Based Evaluation of Ontology Learning. In *5th International Semantic Web Conference, (ISWC)*, volume 4273 of *Lecture Notes in Computer Science*, 228–241. Springer.

Du, R.; An, H.; Wang, K.; and Liu, W. 2024. A Short Review for Ontology Learning from Text: Stride from Shallow Learning, Deep Learning to Large Language Models Trend. *CoRR*, abs/2404.14991.

Google Cloud. 2023. Generative AI, Privacy, and Google Cloud. https://services.google.com/fh/files/misc/genai_privacy_google_cloud.pdf. Accessed: 2025-07-22.

Han, S.; Schoelkopf, H.; Zhao, Y.; Qi, Z.; et al. 2024. FOLIO: Natural Language Reasoning with First-Order Logic. In *2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 22017–22031. Association for Computational Linguistics.

Huggingface. 2025. MTEB Leaderboard. https://huggingface.co/spaces/mteb/leaderboard. Accessed: 2025-07-28.

Kanburoglu, A. B.; and Tek, F. B. 2024. Text-to-SQL: A methodical review of challenges and models. *Turkish Journal of Electrical Engineering and Computer Science*, 32(3): 403–419.

Korbak, T.; Balesni, M.; Barnes, E.; Bengio, Y.; Benton, J.; et al. 2025. Chain of Thought Monitorability: A New and Fragile Opportunity for AI Safety. *CoRR*, abs/2507.11473.

Lalwani, A.; Kim, T.; Chopra, L.; et al. 2025. Autoformalizing Natural Language to First-Order Logic: A Case Study in Logical Fallacy Detection. arXiv:2405.02318.

Lee, J.; Chen, F.; Dua, S.; et al. 2025. Gemini Embedding: Generalizable Embeddings from Gemini. *CoRR*, abs/2503.07891.

Levkovskyi, O.; and Li, W. 2021. Generating predicate logic expressions from natural language. In *2021 IEEE SoutheastCon*. Institute of Electrical and Electronics Engineers Inc.

Lu, X.; Liu, J.; et al. 2022. Parsing Natural Language into Propositional and First-Order Logic with Dual Reinforcement Learning. In *29th International Conference on Computational Linguistics (COLING)*, 5419–5431. ICCL.

Mendoza, D.; et al. 2024. Translating Natural Language to Temporal Logics with Large Language Models and Model Checkers. In *conference on Formal Methods in Computer-Aided Design (FMCAD)*, 1–11. IEEE.

Mpagouli, A.; et al. 2007. Converting first order logic into natural language: A first level approach. In *11th Panhellenic Conference on Informatics (PCI)*, 517–526.

Olausson, T.; Gu, A.; Lipkin, B.; Zhang, C. E.; et al. 2023. LINC: A Neurosymbolic Approach for Logical Reasoning by Combining Language Models with First-Order Logic Provers. In *2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5153–5176. Association for Computational Linguistics.

OpenAI. 2024. GPT-4o-mini. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/. Accessed: 2025-07-20.

OpenAI. 2025a. o3-mini benchmark. https://openai.com/it-IT/index/openai-o3-mini/. Accessed: 2025-07-20.

OpenAI. 2025b. o3-mini System Card. https://openai.com/index/o3-mini-system-card/. Accessed: 2025-07-20.

OpenAI. 2025c. OpenAI Platform Documentation: Advanced Usage. https://platform.openai.com/docs/advanced-usage. Accessed: 2025-07-22.

Pan, L.; Albalak, A.; et al. 2023. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In *2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3806–3824. Association for Computational Linguistics.

Papineni, K.; Roukos, S.; et al. 2002. Bleu: A Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 311–318. ACL.

Paris, J.; and Vencovská, A. 2015. *Pure Inductive Logic*. Perspectives in Logic. Cambridge University Press.

Paulson, L. C. 1994. *Isabelle - A Generic Theorem Prover*, volume 828 of *Lecture Notes in Computer Science*. Springer. ISBN 3-540-58244-4.

Ranta, A. 2011. Translating between language and logic: What is easy and what is difficult. In *23rd international conference on Automated deduction (CADE)*, 5–25. Springer.

Seshia, S. A.; Sadigh, D.; and Sastry, S. S. 2022. Toward verified artificial intelligence. *Communications of the ACM*, 65(7): 46–55.

Singh, H.; Aggarwal, M.; and Krishnamurthy, B. 2020. Exploring Neural Models for Parsing Natural Language into First-Order Logic. *CoRR*, abs/2002.06544.

Szegedy, C. 2020. A Promising Path Towards Autoformalization and General Artificial Intelligence. In *13th International Conference on Intelligent Computer Mathematics (CICM)*, volume 12236 of *Lecture Notes in Computer Science*, 3–20. Springer.

Thatikonda, R. K.; Han, J.; Buntine, W. L.; and Shareghi, E. 2024. Strategies for Improving NL-to-FOL Translation with LLMs: Data Generation, Incremental Fine-Tuning, and Verification. *CoRR*, abs/2409.16461.

Tian, J.; Li, Y.; Chen, W.; et al. 2021. Diagnosing the First-Order Logical Reasoning Ability Through LogicNLI. In *2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 3738–3747. Association for Computational Linguistics.

Wilks, Y.; and Fass, D. 1992. The preference semantics family. *Computers & Mathematics with Applications*, 23(2-5): 205–221.

Wu, Y.; Jiang, A. Q.; Li, W.; et al. 2022. Autoformalization with Large Language Models. In *36th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Yang, A.; et al. 2025. Qwen3 Technical Report. *CoRR*, abs/2505.09388.

Yang, Y.; Xiong, S.; Payani, A.; Shareghi, E.; and Fekri, F. 2024. Harnessing the Power of Large Language Models for Natural Language to First-Order Logic Translation. In *62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 6942–6959. ACL.

Ye, X.; Chen, Q.; Dillig, I.; and Durrett, G. 2023. SatLM: Satisfiability-Aided Language Models Using Declarative Prompting. In *37th Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Zettlemoyer, L. S.; and Collins, M. 2005. Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *21st Conference in Uncertainty in Artificial Intelligence (UAI)*, 658–666. AUAI.

Zhang, Y.; Li, M.; Long, D.; et al. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *CoRR*, abs/2506.05176.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

A First-Order Logic

In this section, we recall the background notions of First-Order Logic (FOL) that are behind the contents presented the main paper. We begin with the notion of first-order signature.

Definition 1 (First-order signature)**.** *A first-order signature σ is a set $\mathcal{V} \cup \mathcal{C} \cup \mathcal{F} \cup \mathcal{R}$ such that:*

- *$\mathcal{V}$ is a (countably infinite) set of* variables*;*
- *$\mathcal{C}$ is a set of* constant symbols*;*
- *$\mathcal{F}$ is a set of* function symbols*, each with its arity;*
- *$\mathcal{R}$ is a set of* relation symbols*, each with its arity;*

and $\mathcal{V} \cap \mathcal{C} \cap \mathcal{F} \cap \mathcal{R} = \emptyset$.

The set of *terms* in FOL is defined inductively as follows: (i) every variable in $\mathcal{V}$ and every constant in $\mathcal{C}$ is a term; (ii) if $t_1, \ldots, t_n$ are terms and $f \in \mathcal{F}$ is a function symbol with arity n, then $f(t_1, \ldots, t_n)$ is a term.

Definition 2 (Syntax of FOL)**.** *An* atomic formula *over the signature $\sigma = \mathcal{V} \cup \mathcal{C} \cup \mathcal{F} \cup \mathcal{R}$ is a string of type $p(t_1, \ldots, t_n)$ such that $p \in \mathcal{R}$ is a relation symbol with arity n and t_i is a term, for all $i \in \{1, \ldots, n\}$. The set of* FOL formulas φ over the signature σ *is inductively defined as follows:*

$$\begin{aligned}\varphi \coloneqq\ & p(t_1, \ldots, t_n) \mid \\ & \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi \mid \\ & \exists x \varphi \mid \forall x \varphi\end{aligned}$$

A *literal* is an atomic formula $p(t_1, \ldots, t_n)$ (*positive literal*) or its negation $\neg p(t_1, \ldots, t_n)$ (*negative literal*).

Among the logical symbols mentioned above, parentheses can also be used to clarify which subexpressions a logical connective applies to. For improved readability, parentheses are sometimes omitted; in such cases, the following operator precedence is assumed (from highest to lowest): $\{\neg, \exists, \forall\}$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

A variable x is said to be *quantified* if it falls within the scope of a quantifier such as $\forall x$ (universal) or $\exists x$ (existential). Otherwise, it is considered a *free variable*. A formula may contain both free and quantified variables: when a formula contains no free variables, it is called a *closed formula* or a *statement*.

We now define the semantics of FOL. The core of this definition is the notion of *σ-structure*, for any signature σ.

Definition 3 (σ-structure)**.** *Let $\sigma = \mathcal{V} \cup \mathcal{C} \cup \mathcal{F} \cup \mathcal{R}$ be a signature. A σ-structure $\mathfrak{A}$ is given by:*

- *a* domain $\mathbb{D}_{\mathfrak{A}} \neq \emptyset$*;*
- *for all $c \in \mathcal{C}$, an element $c_{\mathfrak{A}} \in \mathbb{D}_{\mathfrak{A}}$;*
- *for all $f \in \mathcal{F}$ of arity n, a function $f_{\mathfrak{A}} : (\mathbb{D}_{\mathfrak{A}})^n \rightarrow \mathbb{D}_{\mathfrak{A}}$;*
- *for all $p \in \mathcal{R}$ or arity n, a set $p_{\mathfrak{A}} \subseteq (\mathbb{D}_{\mathfrak{A}})^n$.*

Given a σ-structure $\mathfrak{A}$ and a variable evaluation $V : \mathcal{V} \rightarrow \mathbb{D}_{\mathfrak{A}}$, for each term t, we define $\mathfrak{A}_V(t)$ inductively as follows: (i) if $t \in \mathcal{V}$, then $\mathfrak{A}_V(t) = V(t)$; (ii) if $t \in \mathcal{C}$, then $\mathfrak{A}_V(t) = t_{\mathfrak{A}}$; (iii) if $t = f(t_1, \ldots, t_n)$ with $f \in \mathcal{F}$, then $\mathfrak{A}_V(t) = f_{\mathfrak{A}}(\mathfrak{A}_V(t_1), \ldots, \mathfrak{A}_V(t_n))$. In the following, we define the semantics of FOL.

Definition 4. *Let φ be an FOL formula over the signature σ, let $\mathfrak{A}$ be a σ-structure and let V be a variable evaluation. We define the fact that $\mathfrak{A}_V$* satisfies *φ, denoted with $\mathfrak{A}_V \models \varphi$, inductively as follows:*

- *$\mathfrak{A}_V \models p(t_1, \ldots, t_n)$ iff $(V(t_1), \ldots, V(t_n)) \in p_{\mathfrak{A}}$, for all $p \in \mathcal{R}$;*
- *$\mathfrak{A}_V \models \neg\varphi$ iff $\mathfrak{A}_V \not\models \varphi$;*
- *$\mathfrak{A}_V \models \varphi \wedge \varphi'$ iff $\mathfrak{A}_V \models \varphi$ and $\mathfrak{A}_V \models \varphi'$;*
- *$\mathfrak{A}_V \models \varphi \vee \varphi'$ iff $\mathfrak{A}_V \models \varphi$ or $\mathfrak{A}_V \models \varphi'$;*
- *$\mathfrak{A}_V \models \varphi \rightarrow \varphi'$ iff $\mathfrak{A}_V \not\models \varphi$ or $\mathfrak{A}_V \models \varphi'$;*
- *$\mathfrak{A}_V \models \varphi \leftrightarrow \varphi'$ iff $\mathfrak{A}_V \models \varphi \rightarrow \varphi'$ and $\mathfrak{A}_V \models \varphi' \rightarrow \varphi$;*
- *$\mathfrak{A}_V \models \exists x \varphi$ iff there exists a $d \in \mathbb{D}_{\mathfrak{A}}$ such that $\mathfrak{A}_{V'} \models \varphi$, where $V'(x) = d$ and $V'(y) = V(y)$ for all $y \neq x$;*
- *$\mathfrak{A}_V \models \forall x \varphi$ iff for all $d \in \mathbb{D}_{\mathfrak{A}}$ we have that $\mathfrak{A}_{V'} \models \varphi$, where $V'(x) = d$ and $V'(y) = V(y)$ for all $y \neq x$.*

We say that $\mathfrak{A}$ *satisfies* φ, denoted with $\mathfrak{A} \models \varphi$, iff $\mathfrak{A}_V \models \varphi$, for all variable evaluations V. Given two FOL formulas φ and φ' over the signature σ, we say that *φ is equivalent to* φ' when $\mathfrak{A}_V \models \varphi$ iff $\mathfrak{A}_V \models \varphi'$, for all σ-structures $\mathfrak{A}$ and all variable assignments V: this is the same to say that $\mathfrak{A}_V \models \varphi \leftrightarrow \varphi'$ for all σ-structures $\mathfrak{A}$ and all variable assignments V.

It is worth to notice that these notions, here presented in the general case, become clearer when φ is a statement. In this case, since there are no free variables, the fact that $\mathfrak{A}_V \models \varphi$ doesn't depend on the variable assignment V. For any two statement φ and φ', they are *equivalent* when for any σ-structure $\mathfrak{A}$, we have that $\mathfrak{A} \models \varphi$ iff $\mathfrak{A} \models \varphi'$, or equivalently, $\mathfrak{A} \models \varphi \leftrightarrow \varphi'$.

For every FOL formula φ, it is possible to transform it into an equivalent formula, over the same signature, such that the negation ($\neg$) appears only in front of atomic formulas. This normal form is called *Negation Normal Form* (NNF, for short). We define $(\cdot)_{\text{nnf}}$ as the function such that, given in input any FOL formula φ it returns its equivalent formula in NNF.

To verify the equivalence between two formulas φ and φ' over the same signature σ, we employ SMT (*Satisfiability Modulo Theories*) solvers (Barrett and Tinelli 2018). These solvers are highly efficient tools for deciding the *satisfiability* of first-order formulas; that is, they determine whether, given a first-order formula φ over a signature σ as input, there exists a σ-structure $\mathfrak{A}$ and a variable assignment V such that $\mathfrak{A}_V \models \varphi$.

An SMT solver typically consists of two main components: a Boolean solver that handles the propositional structure of φ (often based on the DPLL or CDCL algorithms, as in classical SAT solvers (Biere et al. 2021)) and a *theory solver* that resolves conjunctions of formulas belonging to a specific theory, such as LRA (Linear Real Arithmetic). Over the past decades, driven by the remarkable efficiency of SAT solvers, SMT solvers have evolved into powerful tools for automated reasoning, and they now play a crucial role in tasks such as formal verification and planning. In our experiments, we used the SMT solver Z3 (De Moura and Bjørner 2008).

We reduce the problem of checking equivalence between two statements φ and φ' over the signature σ to a satisfi-

ability problem. Specifically, the statements φ and φ' are equivalent if and only if the statement $\neg(\varphi \leftrightarrow \varphi')$ is not satisfiable. Formally, $\mathfrak{A} \models \varphi \leftrightarrow \varphi'$ for all σ-structures $\mathfrak{A}$ if and only if there does not exist a σ-structure $\mathfrak{A}$ such that $\mathfrak{A} \models \neg(\varphi \leftrightarrow \varphi')$.

B On the Kind of φ Perturbations Chosen

A FOL formula can be modified in several ways. In the main paper, we discuss some logical perturbations that arise from changing a boolean connector with another, by replacing a quantifier with another, or inserting/removing a negation in front of a literal. Other than these modifications, we could also, for instance, have replaced a constant/relational symbol with another (preserving the original arity). We tried that, but in preliminary experiments we found that these types of changes are trivial to spot even for GPT-4O-MINI in the *most similar* task. Accordingly, we restricted our perturbations to those described in the paper, since they seem to provide a more challenging benchmark.

C Translation Function $T()$

In the main paper, we discuss the use of the function $T()$ for our experiments, which is used to translate a FOL formula into an NL utterance: in this appendix, we give more details on how we have implemented $T()$, what are the choices we made, and what are the changes that can be done in further works.

A straightforward manner to perform FOL to NL translation is a rule-based substitution scheme: replacing each symbol in a formula φ with a fixed natural language rendering. Note that there are various ways in which these translations can be made: for instance, we can translate $\forall$ with "for any" or with "for all", $\rightarrow$ with "imply" or "if ... then ...". We need also to specify how to deal with relational and constant symbols, in order to translate atomic formulas such as $\mathrm{Cube}(A)$ in $\mathcal{D}_{\text{Stanford}}$ or $\mathrm{Love}(x_1, x_2)$ in $\mathcal{D}_{\text{FOLIO}}$. For each relational symbol we associate a *positive meaning*, i.e., a string in NL that describes the meaning of the predicate: for instance, the positive meaning of $\mathrm{Cube}(x_1)$ is "x_1 is a cube", and of $\mathrm{Love}(x_1, x_2)$ is "x_1 loves x_2". Then, to complete the translation, we replace the placeholders x_1 and x_2 with the meaning of the predicate arguments: the variable symbols are not modified, while to each constant symbol is associated a meaning (in this case to the constant A is associated "A"). For the sake of readibility, for the translation we provide also a *negative meaning* for each predicate symbol: for instance, $\mathrm{Love}(x_1, x_2)$ has a negative meaning "x_1 doesn't love x_2" and this will be used to translate $\neg\,\mathrm{Love}(x_1, x_2)$. Note that we exclude the functional symbols in our description, since we don't have them in the signatures used.

Once established how to translate the logical operators, the predicate and the constant symbols, one can easily define a translation function $T()$. To give a more precise idea, we define here inductively the translation function used in the paper:

- for any variable x, $T(x)$ is "x"; for any constant symbol c, its translation $T(c)$ is the meaning of c;
- for any atomic formula $\varphi = p(t_1, \ldots, t_n)$, its translation $T(\varphi)$ is given by the positive meaning of the predicate symbol p, in which we replace the placeholders $x_1, \ldots, x_n$ with $T(t_1), \ldots, T(t_n)$;
- for any negative literal $\varphi = \neg p(t_1, \ldots, t_n)$, its translation $T(\varphi)$ is given by the negative meaning of the predicate symbol p, in which we replace the placeholders $x_1, \ldots, x_n$ with $T(t_1), \ldots, T(t_n)$;
- $T(\varphi \wedge \psi)$ is "$T(\varphi)$ and $T(\psi)$";
- $T(\varphi \vee \psi)$ is "$T(\varphi)$ or $T(\psi)$";
- $T(\varphi \rightarrow \psi)$ is "if $T(\varphi)$, then $T(\psi)$";
- $T(\varphi \leftrightarrow \psi)$ is "$T(\varphi)$ if and only if $T(\psi)$";
- $T(\neg\varphi)$ is "it's false that $T(\varphi)$", if φ is not atomic;
- $T(\exists x\varphi)$ is "there is x such that $T(\varphi)$";
- $T(\forall x\varphi)$ is "for all x $T(\varphi)$".

We then remove superfluous spaces or parentheses that can be still present in the translation, we add a period at the end of the sentence and capitalize the first character.

Obviously, different choices could have been made both in how to translate the logical operators and in how to specify the translation process. For example, not using the negative meaning of the predicate symbols, or leaving the parentheses of the original FOL formulas within the NL translations to obtain an unambiguous utterance.

D Computing Infrastructure

We performed the local experiments on a Dell PowerEdge R750 running Red Hat Enterprise Linux 8.7 (Ootpa). The system is equipped with two Intel(R) Xeon(R) Platinum 8360Y CPUs at 2.40 GHz (72 physical cores in total), 512 GB of RAM, 4 TB of disk storage, and an NVIDIA A100 GPU with 80 GB of VRAM.

E Qwen: *Thinking* and *not thinking* mode

For the Qwen models QWEN3-8B and QWEN3-30B-A3B, we conducted experiments in both *thinking* and *non-thinking* modes, as described in (Yang et al. 2025). While the main paper reports results for the *thinking* mode, in this section we provide a comparison with the *non-thinking* counterpart. Since, in the *Logical Translation* task, the two modes yield identical average accuracy, we focus here on the *Most Similar* and *Ranking* tasks; the results are reported in Table 6. Overall, the comparison suggests the following:

- For QWEN3-8B, the performance differences between the thinking and non-thinking modes are marginal. In several cases, the thinking mode even produces a slight decrease in accuracy (notably in *Ranking-Equivalence* and *Ranking-Both*).
- For QWEN3-30B-A3B, in contrast, the thinking mode generally improves performance, yielding substantial gains in certain tasks, up to a 6 point increment in accuracy for *Ranking-Equivalence* and *Ranking-Both*.

Task	Dataset	QWEN3-8B		QWEN3-30B	
		NL	FOL	NL	FOL
Most similar	$\mathcal{D}_{\text{Stanford}}$	.97	.97	.98	.98
	$\mathcal{D}_{\text{FOLIO}}$	.94	.94	.94	.93
Ranking-Equivalence	$\mathcal{D}_{\text{Stanford}}$	.69	.79	.75	.84
	$\mathcal{D}_{\text{FOLIO}}$	.78	.84	.84	.86
Ranking-Negation	$\mathcal{D}_{\text{Stanford}}$	.53	.62	.58	.56
	$\mathcal{D}_{\text{FOLIO}}$	.67	.71	.67	.68
Ranking-Both	$\mathcal{D}_{\text{Stanford}}$	.39	.51	.46	.50
	$\mathcal{D}_{\text{FOLIO}}$	.56	.62	.59	.60

Table 6: Results for QWEN's family models in the *not thinking* mode (average across the seeds)

F Details about $\mathcal{D}_{\text{Stanford}}$ Dataset

Each NL sentence in the dataset is a description of a configuration of the Tarski's World (Barker-Plummer et al. 2007). Hence, the predicate symbols used represent geometric and spatial relations. They include:

- Arity 1: "Cube", "Tet", "Dodec", "Small", "Large", "Medium";
- Arity 2: "Smaller", "Larger", "LeftOf", "RightOf", "BackOf", "FrontOf", "SameRow", "SameCol", "SameSize", "SameShape", "Adjoins";
- Arity 3: "Between".

As outlined in (Brunello et al. 2025), the dataset is considered challenging also by advanced undergraduate students. Some handcrafted examples of NL–FOL pairs, inspired by randomly sampled instances from the dataset, are shown in Table 7. We have conducted some simple experiments to compare the (syntactic) complexity of the FOL instances in $\mathcal{D}_{\text{Stanford}}$ with those in $\mathcal{D}_{\text{FOLIO}}$, finding the following:

- Percentage of atomic sentences: $\mathcal{D}_{\text{Stanford}}$ 0% (our extract does not include them), $\mathcal{D}_{\text{FOLIO}}$ 14.4%.
- In $\mathcal{D}_{\text{Stanford}}$ 2.6 logical operators (boolean operators and quantifiers) are used on average within a FOL formula, while in $\mathcal{D}_{\text{FOLIO}}$ 1.8.
- The 48% of $\mathcal{D}_{\text{Stanford}}$ instances have three or more operators, while in FOLIO only the 23%; with four or more operators, we get 19% in $\mathcal{D}_{\text{Stanford}}$, while only 3% in $\mathcal{D}_{\text{FOLIO}}$; with five or more operators, we get 6% in $\mathcal{D}_{\text{Stanford}}$ and 0.1% in $\mathcal{D}_{\text{FOLIO}}$
- As for the quantifiers: there are 63% instances in $\mathcal{D}_{\text{Stanford}}$ with one quantifier, and 64% in $\mathcal{D}_{\text{FOLIO}}$; with two, 18% in $\mathcal{D}_{\text{Stanford}}$ and 1% in $\mathcal{D}_{\text{FOLIO}}$.

While these statistics attest to the logical complexity of the FOL translations in $\mathcal{D}_{\text{Stanford}}$ (and allow a comparison with $\mathcal{D}_{\text{FOLIO}}$), drawing parallel conclusions about the paired NL sentences is more difficult. One could argue that, despite the complex formulas, the dataset might still be relatively easy if the utterances make the intended logic sufficiently explicit. Quantifying such "explicitness", however, is inherently challenging.

Nonetheless, we believe that the performance of embedding-centric models (Table 4 in the main paper) provides some indirect evidence. If the reference sentence p and its logical translation φ were closely aligned also at the structural/syntax level (other than semantic one), the considered embedding models should perform well on the *most similar* task, because the embeddings of p and $T(\varphi)$ (or, maybe to a lesser extent, of p and φ) would be expected to lie close to each other in embedding space. Instead, the modest performance of the embeddig-centric models on $\mathcal{D}_{\text{Stanford}}$ (less than or equal to $\mathcal{D}_{\text{FOLIO}}$'s), despite their success on NL similarity benchmarks, suggests this is not the case.

G Experiments (Hyper-)Parameters

In this appendix we discuss the (hyper-)parameters that we have used for the experiments.

Regarding the dialogue-based models, all the experiments we discuss in the main paper have been performed through the OpenAI Chat Completions API: in particular, all the (hyper-)parameters are left with their default values except the *seed*, the *max completion tokens*, and the *response format*. The seed parameter is used for the sake of reproducibility and the values it can assume are 3, 12, 26, 85, and 107. We set the max completion tokens to 2500 for GPT-4O-MINI and 10000 for O3-MINI, i.e., the smallest limits that allowed us to obtain complete answers from the models. As already mentioned in the paper, we used the *structured outputs* feature for all the experiments discussed: we require the model to output the reasoning (i.e., a string), and then the final answer, that is, depending on the specific task, a string (for *logical translation*), an integer (for *most similar*), or a list of integers (for *ranking*).

H Prompts

In Template 1-6, we list the prompts we used for the various experiments and types of models. Some prompt templates include placeholders, delimited by <\ and \>, which are replaced with specific values at inference time.

Dialogue-oriented models

As a general rule for dialogue-oriented models, we placed the task specification (and any required output format) in the system prompt, and the instance-specific input in the user prompt. Note that, for the *most similar* and *ranking* tasks, we provide here only the prompts for the NL version of the experiments: with minor changes it is possible to obtain the FOL versions. The full prompts can be found in the code of our experiments within the accompanying supplementary material.

Embedding-centric models

In the case of QWEN-EMB-INST, if the embedding is about a FOL formula the instruction I is given by the string "Encode the first-order logic meaning of the following first-order formula: "; if we are dealing with a NL sentence, then we provide the string "Encode the first-order logic meaning of the following natural-language sentence: ".

I Wrong/Incomplete Translations in $\mathcal{D}_{\text{FOLIO}}$

In the main article, we noted how (Olausson et al. 2023) reported that approximately 11% of the original validation set

NL sentence	FOL translation
If D is a tetrahedron, then it's to the left of B if and only if it is also in front of C	$\text{Tet}(D) \rightarrow (\text{LeftOf}(D, B) \leftrightarrow \text{FrontOf}(D, C))$
Every dodecaedron has a cube that is to its left but is neither in back of or in front of it	$\forall x(\text{Dodec}(x) \rightarrow \exists y(\text{Cube}(y) \wedge \text{LeftOf}(y, x) \wedge \neg\,\text{FrontOf}(y, x) \wedge \neg\,\text{BackOf}(y, x))$
Neither A nor C is left to either C or B	$\neg\,\text{LeftOf}(A, C) \wedge \neg\,\text{LeftOf}(A, B) \wedge \neg\,\text{LeftOf}(C, B) \wedge \neg\,\text{LeftOf}(C, C)$
Some cube is not medium	$\exists x(\text{Cube}(x) \wedge \neg\,\text{Medium}(x))$
F is larger or smaller than D only if it's small	$(\text{Larger}(F, D) \vee \text{Smaller}(F, D)) \rightarrow \text{Small}(F)$
Every tetrahedron to the left of a dodecahedron is also larger than it	$\forall x \forall y((\text{Tet}(x) \wedge \text{Dodec}(y) \wedge \text{LeftOf}(x, y)) \rightarrow \text{Larger}(x, y))$

Table 7: Handcrafted examples inspired by randomly sampled instances of $\mathcal{D}_{\text{Stanford}}$.

of $\mathcal{D}_{\text{FOLIO}}$ contains errors. For our experiments, we chose to work with the training set instead, due to its larger number of instances. However, since the training, validation, and test sets were randomly sampled from the same pool (Han et al. 2024), it is reasonable to assume that a similar error rate may affect the portion of the dataset we used. Thus, we conducted a preliminary assessment (intended to be extended in future work) of the training split by analyzing the instances where the models GPT-4O-MINI and O3-MINI consistently produced outputs judged incorrect across all random seeds.

In the following, we present selected cases from the analyzed instances in which the ground truth FOL formula incorrectly encodes the NL utterance, leading to O3-MINI being wrongly judged as incorrect. All the relational symbols used by O3-MINI are allowed in the translation process:

Ex 1: story-id = 8, instance no. 3:
NL: "Some musicians love music"
FOL: $\exists x(\text{Musician}(x) \rightarrow \text{Love}(x, \text{music}))$,
O3-MINI: $\exists x(\text{Musician}(x) \wedge \text{Love}(x, \text{music}))$;

Ex 2: story-id = 409, instance no. 5,
NL: "If K is not beneficial to people and also not an apple, then it is a fruit with the color red.",
FOL: $\neg(\text{Benefit}(k) \wedge \text{Apple}(k)) \rightarrow \text{RedFruit}(k)$,
O3-MINI: $(\neg\,\text{Benefit}(k) \wedge \neg\,\text{Apple}(k)) \rightarrow \text{RedFruit}(k)$;

Ex 3: story-id = 68, instance no. 2,
NL: "If a film is directedy by a person, the person is a filmmaker.",
FOL: $\forall x \exists y(\text{DirectedBy}(y, x) \rightarrow \text{Filmmaker}(x))$,
O3-MINI: $\forall x \forall y((\text{IsFilm}(x) \wedge \text{DirectedBy}(x, y)) \rightarrow \text{Filmmaker}(y))$;

Ex 4: story-id = 118, instance no. 4,
NL: "A musician can be a singer or a writer.",
FOL: $\exists x(\text{Musician}(x) \wedge \text{Singer}(x) \vee \text{Writer}(x))$,
O3-MINI: $\forall x(\text{Musician}(x) \rightarrow (\text{Singer}(x) \vee \text{Writer}(x)))$;

Ex 5: story-id = 284, instance no. 1,
NL: "Each building is tall.",
FOL: $\forall x(\text{Building}(x) \rightarrow \neg\,\text{Tall}(x))$,
O3-MINI: $\forall x(\text{Building}(x) \rightarrow \text{Tall}(x))$;

Ex 6: story-id = 423, instance no. 4,
NL: "All students not in summer camp do not have class during the weekend.",
FOL: $\forall x(\text{SummerCamp}(x) \rightarrow \text{NoClass}(x))$,
O3-MINI: $\forall x(\neg\,\text{SummerCamp}(x) \rightarrow \text{NoClass}(x))$;

Ex 7: story-id = 321, instance no. 5,
NL: "If Mark Zuckerberg is neither an entrepreneur nor a person who hates working for others, then Mark Zuckerberg is not a risk-averse person.",
FOL: $\neg\,\text{Entrepreneurs}(\text{markZuckerberg}) \vee \neg\,\text{HateWorkingForOthers}(\text{markZuckerberg}) \rightarrow \neg\,\text{RiskAverse}(\text{markZuckerberg}))$,
O3-MINI: $(\neg\,\text{Entrepreneurs}(\text{markZuckerberg}) \wedge \neg\,\text{HateWorkingForOthers}(\text{markZuckerberg})) \rightarrow \neg\,\text{RiskAverse}(\text{markZuckerberg})$;

Ex 8: story-id = 346, instance no. 2,
NL: "All Olympic gold medal winners are good athletes.",
FOL: $\forall x(\text{OlympicGoldMedalWinner}(x) \rightarrow \text{Athlete}(x))$,
O3-MINI: $\forall x(\text{OlympicGoldMedalWinner}(x) \rightarrow (\text{Athlete}(x) \wedge \text{GoodAtSports}(x)))$;

Ex 9: story-id = 171, instance no. 1,
NL: "Some fish may sting.",
FOL: $\exists x \exists y(\text{Fish}(x) \rightarrow \text{Sting}(x, y))$,
O3-MINI: $\exists x(\text{Fish}(x) \wedge \exists y\, \text{Sting}(x, y))$;

Ex 10: story-id = 477, instance no. 3,
NL: "An APP is either related to YouTube or Instagram.",
FOL: $\forall x(\text{Youtube}(x) \vee \text{Instagram}(x))$,
O3-MINI: $\forall x(\text{App}(x) \rightarrow (\text{Youtube}(x) \vee \text{Instagram}(x)))$;

Ex 11: story-id = 146, instance no. 3,
NL: "There is a strong analogy of human aging in the poem Callus 4.",
FOL: $\text{Poem}(\text{callus}\,4) \rightarrow \text{AgingAnalogy}(\text{callus}\,4)$
O3-MINI: $\text{Poem}(\text{callus}\,4) \wedge \text{AgingAnalogy}(\text{callus}\,4)$.

In addition to logical errors in the ground truth FOL formulas, we also observed cases of imprecise or incomplete translations, which can affect evaluation outcomes. Notably, such imprecision may have had limited impact on the original FOLIO evaluation pipeline, which assesses formalization correctness at a coarser level, namely, across full stories. However, this tolerance does not extend to our evaluation protocol, which operates at a finer granularity, judging translation quality at the level of individual sentences.

For example, given a set of premises $p_1, \ldots, p_n$, the corresponding formalizations $\varphi_1, \ldots, \varphi_n$ may collectively offer an adequate representation of the overall narrative. However, the individual mappings $p_i \mapsto \varphi_i$ might not be exhaustive on their own: information missing from, e.g., φ_1 could be implicitly captured in, e.g., φ_2, due to redundancy or overlap between the sentences, which share a same domain.

Below, we provide examples of instances where such imprecisions occur and O3-MINI's answer is wrongly judged

incorrect. Also here, all predicate symbols used by O3-MINI are permitted during the translation process.

Ex 1: story-id = 226, instance no. 2,
NL: "William Dickinson attended Westminster school for high school and then the University of Edinburgh.",
FOL: $\text{Attended}(\text{williamdickinson}, \text{westminster}) \wedge \text{Highschool}(\text{westminster}) \wedge \text{Attended}(\text{williamdickinson}, \text{universityofedinburgh})$,
O3-MINI: $\text{Attended}(\text{williamdickinson}, \text{westminster}) \wedge \text{Highschool}(\text{westminster}) \wedge \text{Attended}(\text{williamdickinson}, \text{universityofedinburgh}) \wedge \text{University}(\text{universityofedinburgh})$;

Ex 2: story-id = 284, instance no. 1,
NL: "If a song is not "Perfect", Sam will never play it.",
FOL: $\forall x(\neg \text{Be}(x, \text{perfect}) \rightarrow \neg \text{Play}(\text{sam}, x))$,
O3-MINI: $\forall x((\text{Song}(x) \wedge \neg \text{Be}(x, \text{perfect})) \rightarrow \neg \text{Play}(\text{sam}, x))$;

Ex 3: story-id = 235, instance no. 2,
NL: "In 2016, a new television series named Westworld debuted on HBO.",
FOL: $\text{Debut}(\text{westworld}, year2016)$,
O3-MINI: $\text{TVSeries}(\text{westworld}) \wedge \text{Debut}(\text{westworld}, year2016)$;

Ex 4: story-id = 235, instance no. 3,
NL: "The TV series Westworld is adapted from the original film in 1973, which was written and directed by Michael Crichton.",
FOL: $\exists x(\text{Adapt}(\text{westworld}, x) \wedge \text{Produce}(x, year1973) \wedge \text{Write}(\text{michael}, x) \wedge \text{Direct}(\text{michael}, x))$
O3-MINI: $\text{TVSeries}(\text{westworld}) \wedge \exists f(\text{Adapt}(\text{westworld}, f) \wedge \text{Produce}(f, year1973) \wedge \text{Write}(\text{michael}, f) \wedge \text{Direct}(\text{michael}, f))$;

Ex 5: story-id = 235, instance no. 4,
NL: "The 1973 film Westworld is about robots that malfunction and begin killing the human visitors.",
FOL: $\text{About}(\text{westworld}, \text{robots})$,
O3-MINI: $\text{Produce}(\text{westworld}, year1973) \wedge \text{About}(\text{westworld}, \text{robots})$.

J Qualitative analysis

Relation between most similar and ranking tasks

As already stated in the main paper, *most similar* is generally easier than *ranking-equivalence*: Table 8 provides a quantitative evidence for the claim, showing in how many instances (micro-average, 5 seeds) a model succeeds in one task and not in the other. For the sake of conciseness, we consider only the OpenAI models.

Relation between models in the same family

Another relevant analysis concerns the relationships between models within the same family and across different families. Our observations can be summarized as follows:

- Within the OpenAI family, the largest model (O3-MINI) not only outperforms the smallest one (GPT-4O-MINI), but the set of instances on which the former fails is generally a subset of those on which the latter fails. In contrast, this pattern is not as clear within the Qwen family.
- The performances of GPT-4O-MINI and QWEN3-8B are comparable. However, among the instances where a failure occurs, approximately half are shared failures and half are not. This suggests that there exists a subset of instances consistently difficult for both models, while each model also struggles on additional instances that the other does not.

```
You are an expert evaluator specializing in
    translating natural language sentences into a
    logical formalism. Your task is to formalize a
    given sentence in First Order Logic (FOL).

Instructions:
You can use the following symbols:
-  logical symbols: ∀ (for all), ∃ (exists), → (
    implies), ↔ (is equivalent to), ∧ (and), ∨ (or)
    , ¬ (not)
- predicate symbols: <\predicate_symbols\>
- contant symbols: <\constant_symbols\>
- variable symbols: x,y,z,...
- non logical symbols: parenthesis ()

Output Format:
Return the First Order Logic sentence that best
    represents the meaning of the given sentence.

Input Format:
Sentence: {sentence}
```

Template 1: *Logical translation* (system prompt)

```
Sentence: <\sentence\>
```

Template 2: *Logical translation* (user prompt)

K Data and Code Availability

We provide all the necessary code to allow interested researchers to reproduce our experiments.

`LINK`: https://github.com/dslab-uniud/NL-FOL-LT

For detailed instructions on executing the code and reproducing the experiments, please refer to the `readme.md` file located in the home folder.

Concerning data availability, the $\mathcal{D}_{\text{FOLIO}}$ dataset is accessible from following the original authors (Han et al. 2024). As for $\mathcal{D}_{\text{Stanford}}$, it is not publicly available by decision of the original authors. Yet, it is possible to ask them access to the dataset for research-related purposes; nevertheless, we included a detailed description of its structure and provided handcrafted instance examples that are representative of the dataset's content.

Model	Dataset	MS AND NOT R		NOT MS AND R		MS AND R		NOT MS AND NOT R	
		NL	FOL	NL	FOL	NL	FOL	NL	FOL
GPT-4O-MINI	$\mathcal{D}_{\text{Stanford}}$	.44	.35	.04	.03	.45	.54	.07	.08
	$\mathcal{D}_{\text{FOLIO}}$	.42	.31	.04	.04	.49	.59	.05	.06
O3-MINI	$\mathcal{D}_{\text{Stanford}}$	.07	.02	.01	.00	.92	.97	.00	.00
	$\mathcal{D}_{\text{FOLIO}}$	.06	.03	.02	.02	.90	.93	.03	.03

Table 8: Relation between the most similar (MS) and ranking (R) tasks for GPT-4O-MINI and O3-MINI. X AND NOT Y means task X succeeded while task Y did not.

Mod1	Mod2	Dataset	MOD1 AND MOD2	MOD1 AND NOT MOD2	NOT MOD1 AND MOD2	NOT MOD1 AND NOT MOD2
GPT-4O-MINI	O3-MINI	$\mathcal{D}_{\text{Stanford}}$	.80	.03	.11	.06
		$\mathcal{D}_{\text{FOLIO}}$	.71	.02	.08	.19
QWEN3-8B	QWEN3-30B-A3B	$\mathcal{D}_{\text{Stanford}}$	.76	.08	.09	.07
		$\mathcal{D}_{\text{FOLIO}}$	.66	.06	.07	.21
GPT-4O-MINI	QWEN3-8B	$\mathcal{D}_{\text{Stanford}}$	.77	.06	.08	.09
		$\mathcal{D}_{\text{FOLIO}}$	.66	.07	.06	.21

Table 9: Relation between different models in the Logical Translation task

```
You are an expert evaluator specializing in semantic
    similarity assessment. Your task is to identify
    the rephrasing that best preserves the original
    meaning of a given sentence.
Instructions:

- You will receive one original sentence followed by
    multiple rephrased versions
- Evaluate each rephrasing based solely on semantic/
    logic equivalence (meaning preservation)
- Ignore differences in grammar, syntax, word order,
    or writing style
- Select and return the rephrasing that most
    accurately conveys the same meaning as the
    original sentence

Evaluation Criteria:

- Prioritize semantic accuracy over grammatical
    correctness
- Focus on whether the logical meaning is the same or
     not

Output Format:
Return only the  number of the selected rephrasing
    that best matches the original sentence's
    meaning.

Input Format:
Sentence: {sentence}

Rephrasing 1: {rephrasing_1}
Rephrasing 2: {rephrasing_2}
...
Rephrasing n: {rephrasing_n}
```

Template 3: NL version of *most similar* (system prompt)

```
Sentence: <\reference\>
<\list_of_sentences_in_F_ms\>
```

Template 4: NL version of *most similar* (user prompt)

```
You are an expert evaluator specializing in ranking
    some NL sentences according to their semantic
    similarity. You will be given a reference and
    other sentences; your task is to rank these
    sentences depending on whether they convey the
    same meaning of the reference or not.

Instructions:
- Ignore difference in grammar, syntax, style, or
    word order. You are only interested in the
    semantic behind the sentences. It is possible
    that two sentences have a different logical
    structure but they convey the same logical
    meaning and this is the only thing you have to
    focus on;
- If one of the sentences is equivalent to the
    reference, it should be ranked first;
- If one of the sentences is equivalent to the
    negation of the reference, i.e. if it has the
    opposite meaning of the reference, it should be
    ranked last.
- If two sentences are equivalent, they should be
    ranked in adjacent positions

Input Format:
Reference: {reference}
Sentence1: {sentence1}
Sentence2: {sentence2}
...
SentenceN : {sentenceN}

Output Format:
Return, after a reasoning stage, the numbers of the
    sentences in order, from the number of the
    sentence that is the most similar to the
    reference to the one that is the least. ([
    number_of_the_statement_ranked_first,
    number_of_the_statement_ranked_second, ...
    number_of_the_statement_ranked_last]).
```

Template 5: NL version of *ranking* (system prompt)

```
Sentence: <\reference\>
<\list_of_sentences_in_$\mathcal{F}_r$\>
```

Template 6: NL version of *ranking* (user prompt)