Preprints.org

Article

Not peer-reviewed version

eXa-LM: A Controlled Natural Language Bridge Between Large Language Models and First-Order Logic Solvers

Francis Frydman *

Posted Date: 11 March 2026

doi: 10.20944/preprints202512.1848.v2

Keywords: neuro-symbolic; LLM; hybrid; computational linguistics

doi:10.20944/preprints202512.1848.v2

Article

eXa-LM: A Controlled Natural Language Bridge Between Large Language Models and First-Order Logic Solvers

Francis Frydman

Independent Researcher, France; francis.frydman@gmail.com

Abstract

Large language models (LLMs) have demonstrated logical reasoning abilities, but their inferences remain non-traceable and lack formal guarantees. We introduce eXa-LM, a controlled natural language (CNL) interface between LLMs and first-order logic solvers. Based on a Controlled Natural Language, our approach aims to create an explicit, verifiable, and interpretable bridge between text and formal logic. It relies on three main components: (1) a reformulation prompt that constrains the LLM to produce a set of facts and rules in CNL, (2) the semantic analyzer eXaSem translating this CNL into a Prolog program made of extended Horn clauses, and (3) the logic engine eXaLog, which integrates a second-order meta-interpreter capable of inferring ontological properties.. We evaluate eXa-LM on three standard benchmarks—PrOntoQA, ProofWriter and FOLIO—comparing it to GPT-4o baselines including Standard prompting, Chain-of-Thought, Logic-LM, LINC, and LLM-TP. Results show that eXa-LM matches or exceeds recent neuro-symbolic systems while providing full traceability of reasoning and intrinsic explainability. On FOLIO, eXa-LM achieves 92.9% accuracy, a +5.5 point gain over LLM-TP, the strongest competing GPT-4o-based method in our comparison. This approach demonstrates the feasibility of a transparent neuro-symbolic reasoning pipeline in which LLMs produce not direct inferences but formally controlled linguistic representations. eXa-LM opens the way to neuro-symbolic architectures that are safer, verifiable and extensible, ultimately integrating hypothetical, abductive or inductive reasoning. Program and data will be made publicly available upon publication.

Keywords: neuro-symbolic; LLM; hybrid; computational linguistics

---

1. Introduction

Large language models (LLMs) have recently improved their logical reasoning capacities, notably via Chain-of-Thought (CoT) techniques (Wei et al. 2022). By making intermediate reasoning steps explicit, these methods substantially improve performance on datasets such as ProofWriter (Tafjord et al. 2021) or FOLIO (Han et al. 2024). However, LLM reasoning remains fundamentally opaque, unverifiable, and subject to the vagaries of natural language: lexical inconsistencies, implicit inferences, and lack of guaranteed logical correctness (Valmeekam et al. 2023).

To address these limitations, several neuro-symbolic approaches have combined the linguistic power of LLMs with the formal rigor of logic solvers. Systems such as Logic-LM (Pan et al. 2023) or LINC (Olausson et al. 2023) translate LLM-produced reasoning into logical representations that can be executed by a first-order deduction engine. These models reach strong results but remain dependent on the quality of the LLM's internal logical translation, which is neither controlled nor interpretable and whose errors are difficult to diagnose.

In this context, we propose eXa-LM, a controlled natural language (CNL) interface between LLMs and first-order logic solvers (Schwitter 2010). Our method aims to establish an explicit and verifiable bridge between the linguistic reformulation of a text and its executable logical representation. It relies on three main components:

 doi:10.20944/preprints202512.1848.v2

- a reformulation prompt designed for the LLM to convert text into a set of facts and rules expressed in a formally defined CNL; - the semantic analyzer eXaSem that turns this CNL into a SWI-Prolog program composed of extended Horn clauses (including contraposition, xor connectors, negative facts and conclusions); - the logic meta-interpreter eXaLog capable of executing the program while integrating second-order reasoning (ontological inheritance).

This approach has three main advantages:

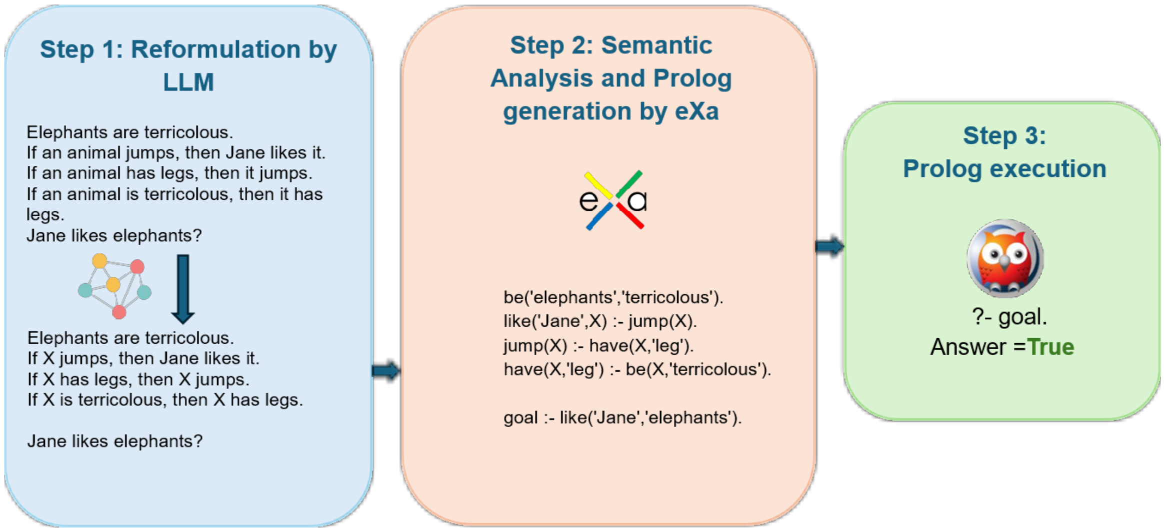

Figure 1. Schematic of the general principle of our method. Step 1: the problem text is submitted to an LLM with a reformulation prompt that converts it into elementary facts and rules. Step 2: eXa analyzes the reformulated text and turns it into an interpretable Prolog program. Step 3: SWI-Prolog executes the program and returns the answer, which is finally formatted and presented by eXa.

1. It guarantees a clear separation between linguistic processing and logical reasoning, allowing traceability and justification of every inference. 2. It provides full control over the syntax and semantics of admissible sentences, reducing reformulation and translation errors. 3. It enables direct and reproducible evaluation of system performance independent of the language model used.

We evaluate eXa-LM on three reference sets—PrOntoQA (Saparov and He 2023), ProofWriter and FOLIO—comparing it with GPT-4o (OpenAI 2024), Logic-LM, LINC and LLM-TP. Our results show that this approach, while more explainable and modular, reaches or exceeds the performance of recent neuro-symbolic systems.

2. Related Work

Logical reasoning from text approaches can be categorized into three families: (1) purely linguistic methods, (2) approaches based on large language models, and (3) neuro-symbolic or neuro-symbolic methods combining both paradigms.

1. Logical reasoning from text

Early work aimed to directly link first-order logic with explicit linguistic representations. Controlled Natural Languages (CNLs) such as Attempto Controlled English (Fuchs et al. 2008) or PENG (Schwitter 2010) have shown that a subset of natural language can be formalized unambiguously to be interpreted by a logic engine. These works inspired many symbolic translation systems (Kuhn 2014) but their linguistic coverage was limited and expressivity rigid. In NLP, efforts were made to extract logical representations from complex sentences via semantic parsing (Dong and Lapata 2018), but these translations remained dependent on costly supervised models and were hard to generalize.

2. Logical reasoning by large language models

The emergence of LLMs changed how logical reasoning is approached. Works such as Chain-of-Thought or Zero-shot CoT (Kojima et al. 2022) showed that LLMs can produce explicit reasoning

 doi:10.20944/preprints202512.1848.v2

chains that improve accuracy on deductive tasks. However, these methods rely on informal linguistic reasoning not verifiable by a logic engine. Recent studies have systematically measured these capabilities. PrOntoQA (Saparov and He 2023), ProofWriter (Tafjord et al. 2021) and FOLIO (Han et al. 2024) are now primary benchmarks. These datasets show that LLMs, while strong at surface reasoning, fail frequently when multiple inference steps or complex logical connectives are involved (Valmeekam et al. 2023).

3. Neuro-symbolic approaches

To overcome these limits, several works have integrated first-order logic engines into LLM reasoning loops. Logic-LM (Pan et al. 2023) combines a generative language model with several logic solvers depending on the task. Although effective on ProofWriter and FOLIO, the system remains dependent on the LLM's implicit translation and relies on iterative self-refinement calls to the LLM. LINC (Olausson et al. 2023) follows a similar path but uses a majority-vote mechanism to improve translation reliability, at the cost of multiple model calls. More recently, LLM-TP (Quan et al. 2026) proposes a decompose-and-formalise framework in which premise–hypothesis pairs are decomposed into entailment trees, verified bottom-up with a theorem prover, and locally refined when failures are detected. This recursive verification strategy improves explanation faithfulness and strong NLI accuracy while reducing costly global regeneration. VANESSA (Sadeddine and Suchanek 2025) proposes translation by successive rewritings toward an intermediate representation less structured than predicate logic to verify reasoning chains of small language models (SLMs).

4. Specificity of eXa-LM

eXa-LM stands out by introducing a controlled natural language (CNL) as the pivot between the language model and the logic solver. Unlike Logic-LM or LINC, the passage from text to first-order logic is not implicitly delegated to the LLM but is specified formally through a reformulation prompt based on precise syntactic and semantic rules. This ensures traceability, reproducibility and verifiability of the logical representation while allowing fine-grained control over the admissible sentence grammar.

Finally, eXa-LM uses the same semantic analyzer and logic engine for all reasoning types; the logic engine is based on an extensible Prolog meta-interpreter for new tasks.

3. Reformulation Prompt

The reformulation prompt, provided in the Appendix, is a sequence of instructions that allows the LLM used (GPT-4o, version 2025-03-26) to convert the submitted text (in one of the LLM-supported languages) into a base of facts and rules accompanied by a question, all written in a Controlled Natural Language (CNL) analyzable by eXa.

Main syntactic and semantic restrictions are:

- Text in French [1], no synonyms, no ambiguous coreferences, elementary sentences (i.e., with a single non-auxiliary verb), optionally connected by "and", "or", "xor" (exclusive or).

- Rules in assertions must have the form: "If <conditions> then <conclusions1> [else <conclusions2>]".

Where:

<conditions> ::= <elementary_sentence 1> [{ and | or | xor <elementary_sentence i> }] <conclusions> ::= <elementary_sentence 1> [{ and | xor <elementary_sentence i> }]

Assertions and questions are treated by two distinct sections of the prompt with slightly different constraints. In particular, questions that are rules whose conclusion contains "or" or "and" must be reformulated using $p \rightarrow q \equiv \neg p \vee q$, preserving parentheses and distributing negation to the innermost level when needed.

The prompt also contains semantic transformation instructions (II.6, II.11, II.12, II.14). The main of these instructions is:

[1] eXa is a monolingual system originally designed for French. All examples in this paper have been translated into English for ease of understanding.

doi:10.20944/preprints202512.1848.v2

"II.6 - Replace the subject of each rule "If ... then ..." and of each fact containing a xor by "X" only when it denotes a general category, while preserving the exact structure of the rule."

For example: "If people order takeout frequently in college, then they work in student jobs on campus" will be reformulated to: "If X order takeout frequently in college, then X work in student jobs on campus"

Some syntactic reformulation instructions may later be removed from the prompt and delegated to eXa, reducing the LLM's computational burden.

4. eXa

eXa is a program that takes as input a controlled French text (according to the constraints above) composed of assertions and questions, converts it into an executable SWI-Prolog program (Wielemaker et al. 2012), and outputs answers to questions optionally accompanied by justification traces of the derivation.

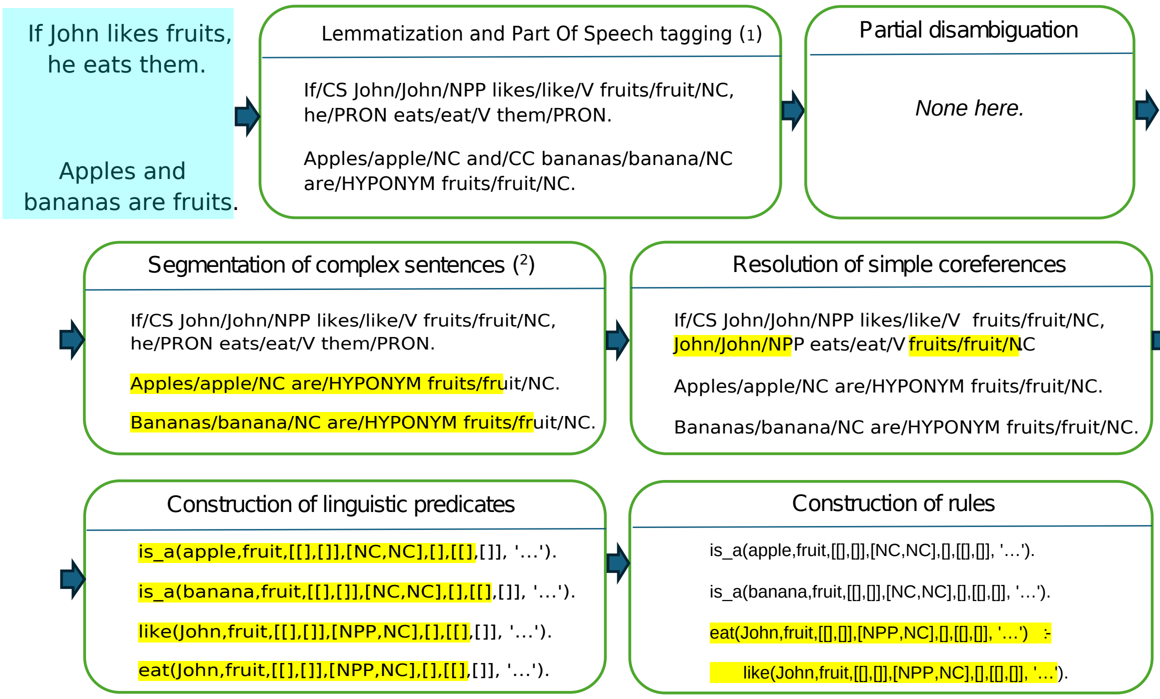

Figure 2. Full processing chain executed by module eXaSem: lemmatization and POS tagging, partial disambiguation, segmentation, resolution of simple coreferences (grammatical anaphora), construction of linguistic predicates and rule construction. Transformations at each step are highlighted in yellow in the original figure.
(1) Uses the tagged French lexicon from (Chrupala et al. 2008)
(2) (Feneyrol 1981)

4.1. eXaSem

This module applies a complex chain of linguistic processing to the input text, as described in Figure 2. eXaSem produces a unified representation we call "linguistic predicates", defined as:

```
Linguistic predicate := <verb>(
    <attributes>,
    <preposition_list>,
    <syntax_tag_list_of_attributes>,
    <verb_quantifiers>,
    <attribute_qualifiers>,
    <tagged_source_sentence> )
```

Example: "John Doe goes often from Paris to Marseille with a big car."
Produces the linguistic predicate:

 doi:10.20944/preprints202512.1848.v2


```
go('John_Doe', 'Paris', 'Lyon','car',
[[],[from],[to],[with]],
[NNP,NNP,NNP,NN],
['often'],[[],[],[],[big]],
'John/NNP Doe/NNP go/goes/VBZ
often/RB from/IN Paris/NNP
to/IN Marseille/NNP with/IN a/DT
big/JJ car/NN').
```

Note: the tagged source sentence associated with the linguistic predicate is used by the eXaGen module.

4.2. eXaLog

eXaLog: Translator to first-order logic (FOL) with "xor" connector into extended Horn clauses, Prolog meta-interpreter and execution environment for Prolog programs.

The meta-interpreter enables: - Tracing of reasoning to produce explanations. - Interpreting negative facts and rules with negation in conclusions (extension of Horn clauses) to handle the "xor" connector, while remaining in a closed-world assumption (CWA).

Because Prolog only accepts Horn clauses, eXaLog first translates assertions into extended Horn structures (with negation).

Example transformations :

Table 1. List of transformations performed by eXaLog on assertions.

Original rule	Transformed form(s)
If $A \wedge B \Rightarrow C \wedge D$	$\rightarrow\ A \wedge B \Rightarrow C$
	$\rightarrow\ A \wedge B \Rightarrow D$
$A \oplus B$	$\rightarrow\ \neg A \Rightarrow B$
	$\rightarrow\ \neg B \Rightarrow A$
$A \Rightarrow (B \oplus C)$	$\rightarrow\ \neg B \wedge \neg C \Rightarrow \neg A$
	$\rightarrow\ A \wedge B \Rightarrow \neg C$
	$\rightarrow\ A \wedge C \Rightarrow \neg B$
$A \oplus B \Rightarrow C$	$\rightarrow\ A \wedge \neg B \Rightarrow C$
	$\rightarrow\ \neg A \wedge B \Rightarrow C$

Contrapositive rules are also added when possible to implement modus tollens, which is not native to Prolog.

This module also supports hypothetical reasoning, i.e., questions posed as rules without transforming them into logical formulas. In particular, when all assertions are rules (no facts), Prolog cannot natively produce an answer regardless of the question form.

Example: If Jean likes strawberries then he buys them. If Jean buys strawberries then he eats them. Question: If Jean likes strawberries, then does he eat them?

4.3. eXaMeta

eXaMeta: Second-order rule meta-interpreter.

This module enriches reasoning by interpreting second-order rules—rules where at least one predicate is a variable. For example, it handles inheritance of properties via ontological relations like "is_a" in syllogistic reasoning.

Example of a metarule:

```
If X<>Y and X is_a Y and RO(Y,Z)
then RO(X,Z).
```

 doi:10.20944/preprints202512.1848.v2

Where R0 is a predicate variable.

Applied to:

'Socrates is a man. Men are mortal." this metarule allows deducing "Socrates is mortal" by instantiating: X=Socrates, Y=man, Z=mortal, R0=be, without reformulating "Men are mortal" as a specific rule.

4.4. eXaGen

eXaGen: Generation of natural language answers and explanations.

This module generates human-readable explanations of the deduction chain leading to an answer.

Example:

A Twingo is a car. The Twingo does not go fast. If a car does not go fast then it drives slowly.

Q: How does a Twingo drive? A: slowly.

Q: Why? A: Because the Twingo does not go fast, a Twingo is a car, and if a car does not go fast then it drives slowly.

4.5. eXaGol

eXaGol: Relational learning (Muggleton 1991) from texts to generate hypotheses as new rules. Not used in this study, so not detailed here.

5. Experiments

We evaluate eXa-LM on three datasets commonly used for deductive reasoning from text: PrOntoQA, ProofWriter and FOLIO, and compare it to GPT-4 and GPT-4o, LINC and Logic-LM.

PrOntoQA (Saparov and He 2023) is a synthetic dataset to evaluate LLM deductive abilities. We use the same subset as Logic-LM (the hardest, depth 5), containing 500 tests. PrOntoQA only contains "is" and "is a" relations. Example:

Context.

Each wumpus is not amenable. Yumpuses are transparent. Each yumpus is a rompus. Each rompus is luminous. Rompuses are impuses. Impuses are not fruity. Each impus is a vumpus. Vumpuses are bitter. Vumpuses are jompuses. Jompuses are amenable. Every jompus is a zumpus. Zumpuses are not shy. Zumpuses are numpuses. Every numpus is cold. Every numpus is a dumpus. Every dumpus is small. Each dumpus is a tumpus. Wren is a yumpus.

Question.

Wren is amenable. A) True, B) False.

Answer.

A) True

ProofWriter (Tafjord et al. 2021) is another frequently used deductive reasoning dataset, somewhat more varied and complex than PrOntoQA. It consists of a set of simple facts and rules. We use the same random sample of 600 examples (depth 5) as Logic-LM. As eXa currently operates under closed-world assumption (CWA), we convert Unknown answers to False. Example: **Context.**

The bald eagle needs the lion. The bald eagle needs the mouse. The bear likes the bald eagle. The bear needs the lion. The bear sees the bald eagle. The lion is blue. The mouse sees the bear. If something sees the lion then it likes the bald eagle. If something likes the bear then it likes the bald eagle. If something needs the bear and the bear is blue then the bear is round. If something likes the mouse then the mouse sees the bear. If something likes the mouse and the mouse is nice then it likes the bear. If something needs the lion and it likes the bald eagle then it likes the mouse. If something likes the bear and it likes the mouse then the mouse needs the bear. If something needs the bear then it likes the bear. If something likes the bald eagle then it needs the bear.

Question.

The bald eagle sees the bear. A) True, B) False, C) Unknown.

Answer.

C) Unknown → B) False

FOLIO (Han et al. 2024) is a set of linguistically and logically complex texts, written by experts and inspired by real-world knowledge. Resolving questions requires complex first-order logical reasoning. We use the same FOLIO test set as LINC and Logic-LM: 204 examples minus 22 errata reported by LINC authors, i.e., 182 examples. As for ProofWriter, we map Unknown to False for CWA.

Example: **Context.**

If people perform in school talent shows often, then they attend and are very engaged with school events. People either perform in school talent shows often or are inactive and disinterested members of their community. If people chaperone high school dances, then they are not students who attend the school. All people who are inactive and disinterested members of their community chaperone high school dances. All young children and teenagers who wish to further their academic careers and educational opportunities are students who attend the school. Bonnie either both attends and is very engaged with school events and is a student who attends the school, or she neither attends and is very engaged with school events nor is a student who attends the school.

Question.

If Bonnie is either both a young child or teenager who wishes to further her academic career and educational opportunities and chaperones high school dances or neither is a young child nor teenager who wishes to further her academic career and educational opportunities, then Bonnie is either a student who attends the school or is an inactive and disinterested member of the community. A) True, B) False, C) Uncertain.

Answer.

A) True.

We compare eXa-LM results to four references: - GPT-4o, in two modes: (1) Standard: "Answer yes or no, using only the provided data"; (2) Chain-of-Thought (CoT): "Answer yes or no, using only the provided data, after decomposing the reasoning step-by-step." - Three neuro-symbolic systems: LINC, Logic-LM and LLM-TP representing the state of the art in combining LLMs and logic solvers.

6. Results and Discussion

For PrOntoQA and ProofWriter, GPT-4o was only used to translate texts into French; eXa's semantic analyzer can interpret the texts directly without further transformation. The perfect (100%) results for these two datasets are therefore unsurprising, since the tasks reduce to controlled semantic translation rather than complex generation.

Table 2. Exact-match accuracy percentages obtained with LLMs alone (Standard and CoT), with LINC, Logic-LM, LLM-TP, and eXa (our method), without (1) and with (2) GPT-4o prompting. The best results are shown in bold green, the second-best ones in bold black.

Dataset	Std.	CoT	Logic-LM	LINC	LLM-TP	eXa (ours)
PrOntoQA	91.0	**99.0**	88.5	92.0	96.0	**100.0 (1)** (+1.0)
ProofWriter	52.4	55.1	91.1	95.1	**98.7**	**100.0 (1)** (+1.3)
FOLIO	63.2	67.6	84.1	86.3	**87.4**	**92.9 (2)** (+5.5)

(*) Results reproduced from the corresponding cited publications.

For FOLIO, we used a 20-step reformulation prompt (see Appendix) with in-context examples. GPT-4o reliably followed the instructions, producing outputs conforming to the CNL in a single pass in all but one case.

On FOLIO, eXa-LM reaches 92.9%, compared with 87.4 for LLM-TP, 86.3 for LINC, 84.1 for Logic-LM, 67.6 for GPT-4o CoT, and 63.2 for GPT-4o Standard in Table 2. In this comparison, eXa-LM

 doi:10.20944/preprints202512.1848.v2

achieves the strongest reported performance while preserving full traceability and explicit formal control of the reasoning process.

Further gains are currently limited by the remaining weaknesses of eXa-LM discussed below.

7. Impact of Closed-World Conversion (CWA)

Converting OWA to CWA maps ternary labels (True/False/Unknown) to binary (True/False). A random baseline accuracy rises from 1/3 to 1/2 under this mapping. We computed expected adjustments and found small differences (ProofWriter: negligible, FOLIO: 0.0118 absolute difference).

8. Error Analysis

Table 3. summarizes the errors and their sources.

Error Source	Count
LLM reformulation	1
eXa semantic analysis	5
Missing explicit information in input	8
Total	**13**

We attribute the low number of LLM reformulation errors to improved instruction following. Semantic-analysis errors can be reduced by extending grammar and parser coverage. The largest category involves missing explicit information in the initial text: eXa-LM cannot rely on implicit world knowledge the LLM might otherwise introduce.

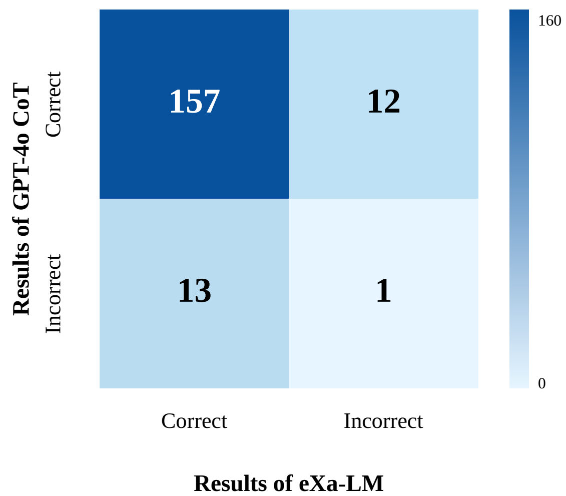

Figure 3. Confusion matrix comparing **GPT-4o CoT** and **eXa-LM** on FOLIO. Darker blue indicates higher agreement counts.

To better understand failure modes, we manually analyzed the 13 errors across the FOLIO dataset. Examples illustrating each category are presented below.

Example 1 – LLM reformulation error.

Context: A man is either kind or evil. No ugly person is handsome. All evil people are ugly. All gentlemen are handsome. All CEOs are gentlemen. Harry is a gentleman.

Question: If Harry is either both evil and ugly, or neither evil nor ugly, then Harry is neither kind nor a CEO?

LLM Reformulation (GPT-4o): Harry is a gentleman. If X is a man, then X is kind xor X is evil. If X is ugly, then X is not handsome. If X is evil, then X is ugly. If X is a gentleman, then X is handsome. If X is a CEO, then X is a gentleman. If (Harry is evil and Harry is ugly) xor (Harry is not evil and Harry is not ugly), then Harry is kind or Harry is a CEO?

Error: Negations in the question's conclusion ("neither kind nor a CEO") were omitted during GPT-4o's reformulation.

Example 2 – Semantic analysis error (eXa).

Context: Diamond Mine is a professional wrestling stable, formed in WWE. Roderick Strong leads Diamond Mine. Diamond Mine includes the Creed Brothers and Ivy Nile. Imperium has a feud with Diamond Mine.

Question: Imperium doesn't have a feud with a professional wrestling stable that includes Ivy Nile?

Error: eXa cannot decompose the relative clause ("that includes Ivy Nile"), which prevents semantic analysis.

Example 3 – Missing explicit information in the input.

Context: *Badults* is a British sitcom series starring members of Pappy's. *Badults* piloted in July 2013 on BBC Three. The working title *The Secret Dude Society* was used for *Badults*. Andrew Collins was the script editor for *Badults*.

Question: No members of Pappy's have starred for a show piloting on BBC Two or BBC Three?

Error: Nothing in the input explicitly states that a "sitcom series" is a "show", which prevents correct inference.

---

9. Ablation Study

We ablate eXa by replacing eXaLog, the logical module, with the reformulation followed by GPT-4o CoT. Results are shown in Table 4.

Table 4. Ablation study. Contribution of **eXa-LM** is **+2.14** points compared to GPT-4o CoT (*McNemar* $p = 0.44$).

Model / Setting	Accuracy (%)
GPT-4o CoT (baseline)	91.85
Reformulation + GPT-4o CoT (ablation)	90.76
eXa-LM (ours)	**92.90**

10. Independence of the LLM

We chose GPT-4o for the reformulation prompt because it gave the best results. The prompt was also tested with Mistral-Large2 (Mistral AI 2024) with 81.3% accuracy (vs. 92.90% for GPT-4o). The difference is mainly due to Mistral's difficulty in consistently executing the abstraction instruction (II.6).

11. Computational Costs

Execution costs of eXa-LM decompose into: (1) cost of reformulation prompt execution by GPT-4o, and (2) cost of running eXa on a standard CPU (Intel i9-185H).

Table 5. Average **eXa-LM** execution time per task (in seconds).

Dataset	Average execution time (s)
PrOntoQA	0.21
ProofWriter	0.27
FOLIO	0.21

doi:10.20944/preprints202512.1848.v2

Table 6. Token consumption and average execution time for the **FOLIO** reformulation prompt.

Metric	GPT-4o CoT	eXa-LM
Prompt + Task tokens	350	1,100
Output tokens	850	1,200
Total tokens	1,200	2,300
Execution time (s)	∼3	∼5

12. Scalability

Qualitatively, scalability concerns extending task variety and grammar coverage. Quantitatively, scalability depends on LLM response time for reformulation and eXa execution time. We measured eXa execution time as a function of number of generated Prolog clauses: execution time grows approximately linearly at 0.005583 s per clause (R^2 ≈ 0.989). Figure 4 illustrates this relation.

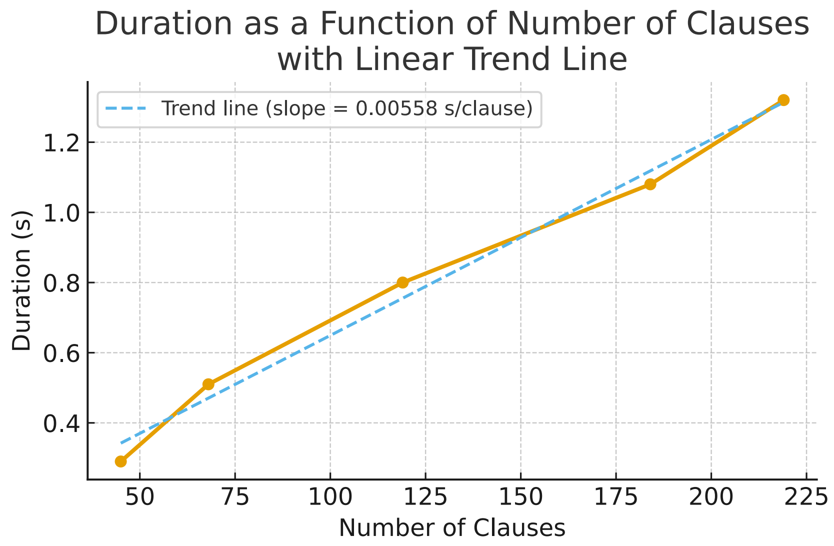

Figure 4. Execution time of **eXa-LM** as a function of the number of generated Prolog clauses. The relationship is approximately linear ($R^2 \approx 0.989$), with an average slope of 0.005583 s per clause.

13. Conclusion

We introduced eXa-LM, a controlled natural language bridge between LLMs and first-order logic solvers ensuring traceability and explainability while remaining competitive with recent neuro-symbolic systems. Constraining LLM outputs to a CNL and executing them via a symbolic pipeline yields competitive performance and full auditability of deductions. eXa-LM highlights the feasibility of a transparent neuro-symbolic architecture where linguistic processing and logical reasoning are explicitly separated and controllable.

14. Limitations

1. **Limited linguistic coverage and expressivity.** The CNL grammar is deliberately narrow, excluding nested relatives and complex passives; linguistic predicates do not encode tense or grammatical gender. *Future work:* Extend grammar via semi-supervised learning and enrich predicates.
2. **Absence of abductive and inductive reasoning. eXa-LM** implements deduction and some hypothetical reasoning; abduction is not implemented and induction is present but unused. *Future work:* Leverage **eXaGOL** for rule induction and implement abduction.
3. **Limits of logical representation. eXaLog** extends Prolog but does not cover full FOL (nested quantifiers, multiple quantifier scopes). *Future work:* Support richer FOL constructs and partial skolemization strategies.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 March 2026 doi:10.20944/preprints202512.1848.v2

4. **Closed-world assumption (CWA).** Simplifies reasoning but limits inference in partially known contexts. *Future work:* Study mixed CWA/OWA variants.
5. **Extraction of implicit knowledge from LLMs.** Some FOLIO errors stem from missing intermediate relations. *Future work:* Design prompting methods to surface missing intermediate relations.
6. **Robustness evaluation.** Current evaluation uses inference-focused datasets. *Future work:* Test on broader QA datasets (**SQuAD** (Rajpurkar et al. 2016), **Natural Questions** (Kwiatkowski et al. 2019), **BoolQ** (Clark et al. 2019), **DROP** (Dua et al. 2019), **HotpotQA** (Yang et al. 2018))
7. **Portability and multilingualism. eXa** is French-first. *Future work:* Implement a native English CNL and assess transferability.

15. Ethical Considerations

We discuss energy costs associated with large models (Strubell et al. 2020) and note that our experiments use in-context learning rather than model training, limiting carbon footprint relative to full retraining. No sensitive personal data were used. LLMs were used only for translation and minor stylistic edits; all scientific contributions come from the authors.

Appendix A. Reformulation Prompt

Please perform the following actions in the exact order indicated. At each step, apply the instructions to the full result of the previous step.

Translate the text below into French, always using the same translation for each word and expression. When two forms are equivalent, keep the simpler. When several expressions of different specificity are synonyms in context, normalize by keeping the more specific expression.
Example: carte, identité, carte d'identité, carte nationale d'identité → carte nationale d'identité.
Translate the titles of works mentioned.

I – QUESTION PROCESSING:

- **I.1** – Isolate the translated question.
- **I.2** – When an expression in the question corresponds to an exclusive or, rewrite it using "xor".
- **I.3** – When the question is a rule "If ... then ...", and its conclusion contains a conjunction ("and"), a disjunction ("or"), or a conjunction of negations ("neither"), rewrite it using: "If a then b → ((negation of a) or (b))." Parentheses should be preserved and negation distributed inward.
- **I.4** – When the question contains a propositional operator (e.g., "It is true that...", "It is false to say that..."), apply it to the question then remove the operator.

Set the question result aside. Subsequent treatments must not affect the question.

II – ASSERTION PROCESSING:

- **II.1** – Decompose assertions into elementary sentences (single non-auxiliary verb) or rules ("If ... then ...").
- **II.2** – Rewrite sentences beginning with "All" or "No", and those whose subject is a general category, into rules "If ... then ...".
- **II.3** – Replace biconditional rules "If and only if a then b" by:
  - If a then b.
  - If b then a.
- **II.4** – Rewrite exclusive-or expressions using "xor".
- **II.5** – Replace facts of the form "a and b and ... xor not a and not b ..." by rules "If a then b. If b then a. ..."
- **II.6** – Replace the subject of each "If ... then ..." rule and each fact containing xor by "X" only when it denotes a general category, preserving rule structure.
- **II.7** – When a rule contains conjunctions of the form <subject> <verb> <comp1> and <comp2>..., rewrite as:
  - <subject> <verb> <comp1> and <subject> <verb> <comp2>.
- **II.8** – Replace "At least one" by "Some".
- **II.9** – Replace pronouns in each rule with the corresponding name or variable present in that rule.
- **II.10** – Eliminate tautological rules where a condition is identical to the conclusion.
- **II.11** – If there are rules of the form:
  - If X is a <generic2> then <generic1> <action-verb> X.
  - Add: If X is a <generic2> and Y <action-verb> X then Y is an X.
- **II.12** – If a condition generalizes another condition in the same rule, remove it.
- **II.13** – Combine the complete result: facts, rules and question after processing.
- **II.14** – When two expressions represent the same state in different forms (stative vs. resultative), simplify to the stative form.
- **II.15** – If there are no facts (only rules) and the question is of the form "All A are B", rewrite the question as a rule: "If X is an A then X is a B."
- **II.16** – When equivalence, synonymy or aliasing between denominations is explicited, unify across all rules, facts and the question.

References

Chrupala, Grzegorz, Georgiana Dinu, and Josef van Genabith. 2008, May. Learning morphology with Morfette. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, and D. Tapias (Eds.), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Clark, Christopher, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019, June. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In J. Burstein, C. Doran, and T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 2924–2936. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1300.

Dong, Li and Mirella Lapata. 2018, July. Coarse-to-fine decoding for neural semantic parsing. In I. Gurevych and Y. Miyao (Eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, pp. 731–742. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1068.

Dua, Dheeru, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019, June. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In J. Burstein, C. Doran, and T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 2368–2378. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1246.

Feneyrol, Claude. 1981. La segmentation automatique de la phrase dans le cadre de l'analyse du français. In *Actes du Congrès international informatique et sciences humaines – L.A.S.L.A., Université de Liège*.

Fuchs, Norbert, Kaarel Kaljurand, and Tobias Kuhn. 2008, 09. Attempto controlled english for knowledge representation. In *Reasoning Web 2008, Lecture Notes in Computer Science*, Volume 5224, pp. 104–124. Springer. https://doi.org/10.1007/978-3-540-85658-0_3.

Han, Simeng, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Wenfei Zhou, James Coady, David Peng, Yujie Qiao, Luke Benson, Lucy Sun, Alexander Wardle-Solano, Hannah Szabó, Ekaterina Zubova, Matthew Burtell, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Alexander Fabbri, Wojciech Maciej Kryscinski, Semih Yavuz, Ye Liu, Xi Victoria Lin, Shafiq Joty, Yingbo Zhou, Caiming Xiong, Rex Ying, Arman Cohan, and Dragomir Radev. 2024, November. FOLIO: Natural language reasoning with first-order logic. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, pp. 22017–22031. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.1229.

Kojima, Takeshi, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Kuhn, Tobias. 2014, March. A survey and classification of controlled natural languages. *Computational Linguistics 40*(1), 121–170. https://doi.org/10.1162/COLI_a_00168.

Kwiatkowski, Tom, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics 7*, 452–466. https://doi.org/10.1162/tacl_a_00276.

Mistral AI. 2024. Mistral large 2 (v24.07). Paris, France.

Muggleton, Stephen. 1991, February. Inductive logic programming. *New Gen. Comput. 8*(4), 295–318. https://doi.org/10.1007/BF03037089.

Olausson, Theo, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023, December. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In H. Bouamor, J. Pino, and K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 5153–5176. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.313.

OpenAI. 2024. Gpt-4o system card.

Pan, Liangming, Alon Albalak, Xinyi Wang, and William Wang. 2023, December. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In H. Bouamor, J. Pino, and K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Singapore, pp. 3806–3824. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.248.

Quan, Xin, Marco Valentino, Louise A. Dennis, and André Freitas. 2026. Decompose-and-formalise: Recursively verifiable natural language inference.

Rajpurkar, Pranav, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016, November. SQuAD: 100,000+ questions for machine comprehension of text. In J. Su, K. Duh, and X. Carreras (Eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, pp. 2383–2392. Association for Computational Linguistics. https://doi.org/10.18653/v1/D16-1264.

Sadeddine, Zacchary and Fabian M. Suchanek. 2025, July. Verifying the steps of deductive reasoning chains. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria, pp. 456–475. Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-acl.25.

Saparov, Abulhair and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought.

Schwitter, Rolf. 2010. Controlled natural languages for knowledge representation. In *Coling 2010: Posters*, pp. 1113–1121.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2020, Apr. Energy and policy considerations for modern deep learning research. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Volume 34, pp. 13693–13696. https://doi.org/10.1609/aaai.v34i09.7123.

Tafjord, Oyvind, Bhavana Dalvi, and Peter Clark. 2021, August. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In C. Zong, F. Xia, W. Li, and R. Navigli (Eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online, pp. 3621–3634. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-acl.317.

Valmeekam, Karthik, Shubham Sreedharan, and Subbarao Kambhampati. 2023. Can llms really reason and plan? In *Proceedings of AAAI 2023*.

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

Wielemaker, Jan, Tom Schrijvers, Markus Triska, and TorbjÖrn Lager. 2012, January. Swi-prolog. *Theory Pract. Log. Program. 12*(1–2), 67–96. https://doi.org/10.1017/S1471068411000494.

Yang, Zhilin, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018, October-November. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii (Eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 2369–2380. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1259.