Talk2Prolog: the Context, the Dataset, and the Methodology through the Lens of Computational Linguistics*

Laura Fuciarelli[1,*,†], Angelo Ferrando[2], Andrea Gatti[1,†] and Viviana Mascardi[1]

[1]Department of Computer Science, Bioengineering, Robotics and Systems Engineering, University of Genova, Genova, Italy
[2]Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, Modena, Italy

Abstract

In her Master Thesis in Computer Science defended in March 2026 at the University of Genova, Italy, the first author of this paper presented Talk2Prolog, a system designed to convert natural language sentences into Prolog facts, rules, and queries suited to represent – respectively – the beliefs, the reasoning mechanisms, and the test goals that characterize BDI (Belief-Desire-Intention) agents. Talk2Prolog is aimed at enabling natural language interaction with symbolically-grounded knowledge systems in a manner that is lightweight and methodologically rigorous.

Grounding Talk2Prolog in a well-defined methodology rooted in computational linguistics lays the foundation for Talk2Prolog **explainability**. By relying on conventional natural language processing techniques and libraries — deliberately avoiding Large Language Models (LLMs) and keeping the use of deep learning to a bare minimum — Talk2Prolog also achieves **frugality**, **sustainability**, and **transparency**.

Notwithstanding its lightweight implementation, Talk2Prolog yields encouraging results. Each sentence is processed independently with no degradation in accuracy; the system runs entirely on a standard laptop with not even internet connectivity required; and its execution speed surpasses that of LLMs still keeping comparable accuracy on tasks related with semantics and reasoning.

The architecture, implementation, and experimental results on Talk2Prolog have been presented at the 8th International Workshop on EXplainable, Trustworthy, and Responsible AI and Multi-Agent Systems (EXTRAAMAS 2026), and will appear in the EXTRAAMAS Proceedings.

This CILC 2026 paper summarizes the EXTRAAMAS contents and complements them by providing details on (i) the background and state of the art (namely, Talk2Prolog **context**); (ii) the **dataset** made up of more than 850 English-Prolog pairs, designed and implemented on purpose to measure Talk2Prolog performances; and (iii) the **methodology** that we followed to design Talk2Prolog.

Keywords

From Natural Language to Prolog, Logic Programming, Logic Translation, Semantic Parsing, Natural Language Processing, Computational Linguistics

1. Introduction and Motivation

In November 2022, ChatGPT was released to the general public. Its impressive and unprecedented capabilities dramatically changed the public perception of Artificial Intelligence (AI), to the point that people who are not Computer Science professionals often seem to associate the term "AI" with Generative AI (GenAI) and Large Language Models (LLMs), disregarding the previous 70 years of research.

Language is recognized by many scientists and philosophers as one of the features that make humans unique [1, 2, 3]. It is not surprising, hence, that since the dawn of AI in 1956 – and even more after 2022 – natural language processing (NLP) and machine translation have played a fundamental role in steering the AI direction, in boosting its success, and – as we should not forget – in entering its cold winters [4].

CILC 2026: The 41st Italian Conference on Computational Logic, June 23–25, 2026, Ferrara, Italy
*Corresponding author.
†This author was a student at the University of Genova, Italy, when the contents of this paper were conceived and implemented.
4934262@studenti.unige.it (L. Fuciarelli); angelo.ferrando@unimore.it (A. Ferrando); andrea.gatti@edu.unige.it (A. Gatti); viviana.mascardi@unige.it (V. Mascardi)
0009-0002-6012-610X (L. Fuciarelli); 0000-0002-8711-4670 (A. Ferrando); 0009-0003-0992-4058 (A. Gatti); 0000-0002-2261-9926 (V. Mascardi)

Currently, as well as in the early days of AI, the possibility to instruct machines using natural language is more and more attractive. In fact, this is one of the problems that Agentic AI and last-generation LLMs promise to solve, and that make them so fashionable: moving from general natural language to an unambiguous, artificial (programming) language. Whether this could potentially kill white-collar jobs as we know them[1], or just turn out to be the greatest capital misallocation in history[2], is open to debate.

Despite its clear potential, recent surveys on public opinion on "AI" (mainly meant as GenAI) show that people do not trust it for several reasons: it is not sustainable from an environmental, financial and geopolitical point of view, and creates significant privacy issues, especially considering the black box nature of deep learning based technologies[3].

The recent crisis in the Middle East – hopefully steering towards a de-escalation right now[4] – has shown that data centers are the perfect target for attacks[5], and that the infrastructure upon which GenAI builds upon is very fragile, with grid scarcity representing its most significant bottleneck[6]. In case of energy shortage – which is far from being an unrealistic scenario – it is likely that the huge amount of energy that data centers consume, would be directed towards much more critical infrastructures.

In this current technological and geopolitical landscape, we have designed, implemented and tested Talk2Prolog, a software that emerges as an alternative to LLMs and GenAI for moving from a controlled subset of natural language to an artificial programming language.

Talk2Prolog is driven by a well-founded rule-based approach based on a methodology, supported by an inspectable and debuggable implementation, which makes it explainable and copes with the lack of trust.

It is as lightweight, frugal, and privacy-preserving as possible. By frugal, we mean that the design and implementation of the software requires and consumes only the resources that are truly needed, no more. It is both more sustainable on a macro-level for the planet, but also on a micro-level, since it can be easily installed and run locally on a modern personal computer with no special equipment. It intentionally does not use LLMs and transformers, avoiding complex machine and deep learning techniques as much as possible.

Talk2Prolog is available online together with full documentation (Laura Fuciarelli's Master Thesis entitled "From Natural Language to Prolog") and all the necessary resources (code and dataset) to reproduce the experiments presented therein[7].

The architecture, implementation, and experimental results on Talk2Prolog have been presented at the 8th International Workshop on EXplainable, Trustworthy, and Responsible AI and Multi-Agent Systems (EXTRAAMAS 2026[8]). Its software modules are shown in Figure 1, taken from the EXTRAAMAS 2026 paper [5]: Talk2Prolog is a Python program that receives as input a Prolog file, which constitutes the pre-existing knowledge base (possibly empty), and continuously feeds it with translated sentences, if those sentences are recognized as facts or rules. If the sentence is recognized as a query, the translated Prolog query is run on the contents of this ever-growing Prolog knowledge base. As a Prolog engine we use SWI-Prolog, accessed from inside the Python code via the PySwip library.

This architecture supporting a feeding-querying loop allowed us to run semantic experiments, manually checking whether Prolog facts and rules entered into the Prolog knowledge base through translation, could be used to correctly answer Prolog queries, also obtained via Talk2Prolog translation.

[1] https://shumer.dev/something-big-is-happening, accessed on July 13, 2026.

[2] https://sg.finance.yahoo.com/news/big-techs-ai-spending-greatest-163608366.html, https://www.businessinsider.com/stock-market-today-tech-stocks-selloff-openai-revenue-targets-orcl-2026-4, accessed on July 13, 2026.

[3] https://hai.stanford.edu/ai-index/2025-ai-index-report/public-opinion, https://www.turing.ac.uk/news/publications/understanding-public-attitudes-ai-second-survey, https://www.pewresearch.org/global/2025/10/15/how-people-around-the-world-view-ai/, accessed on July 13, 2026.

[4] https://www.theguardian.com/world/2026/jun/15/us-iran-peace-deal-terms-details-conditions-explained-what-do-we-know-hormuz-lebanon-israel-nuclear, accessed on July 13, 2026.

[5] https://www.forbes.com/sites/phoebeliu/2026/04/21/ai-data-centers-are-now-big-geopolitical-risk-securing-them-against-iran-attackers-drones-business/, accessed on July 13, 2026.

[6] https://enkiai.com/data-center/data-center-power-crisis-2026-the-grid-bottleneck, accessed on July 13, 2026.

[7] https://github.com/LFuciarelli/Talk2Prolog, accessed on July 13, 2026.

[8] https://extraamas.ehealth.hevs.ch/, accessed on July 13, 2026.

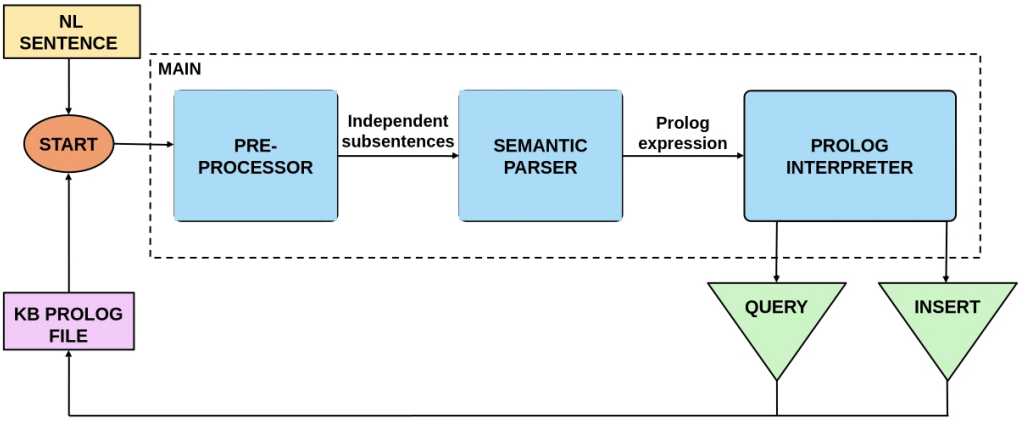

Figure 1: Talk2Prolog architecture.

If yes, this meant that both the knowledge and the query were translated into Prolog in a coherent and consistent way.

Talk2Prolog reaches 88% correct results on these semantic tests. GPT5.2 fed with all the sentences at the same time ("offline mode") and with a very simple prompt that just asks to translate sentences into Prolog without giving further instructions ("uninformed mode") outperforms Talk2Prolog by reaching 95% accuracy, while the other GPT-based combinations (mixing "online mode" – with one sentence translated at a time, independently from the others – and "informed mode" – with some grammar-driven instructions in the prompt) reach between 32% and 74% accuracy. These results convinced us that working on Talk2Prolog is worth and that – for specific tasks and under clear conditions on the input English sentences – it can compete with more flexible and scalable, but also much more resource-consuming, approaches.

This paper complements the EXTRAAMAS one by discussing the background and the state of the art (namely, Talk2Prolog context) in Section 2; the dataset consisting of more than 850 English-Prolog pairs in Section 3; and the methodology that we followed to design Talk2Prolog, rooted on Computational Linguistics, in Section 4. Section 5 concludes and outlines the future directions of our work.

2. Talk2Prolog Context

In this section we cover the basics of English grammar that are needed to understand the methodology we followed, and we present the works that are closer to ours.

2.1. Background

Sentences. A *sentence* is a string of words that expresses a complete thought in natural language.

There are four main types of sentences: *declarative*, used to state a fact; *interrogative*, used to ask a question; *exclamatory*, a modified version of a declarative sentence used to add emphasis or show emotion, urgency, or high volume; *imperative*, used to tell a command to someone or something.

There are two grammatical voices for sentences: *active*, in which the subject of the sentence performs the action of the verb, and *passive*, in which the subject receives the action of the verb.

Clauses. Sentences are formed by *clauses*.

A clause is a group of words that contains a *subject* and a *verb*. The verb forms a relationship that expresses an action or state of being of the subject.

Clauses can function as an independent sentence, but it is not always the case. In fact, sentences might contain multiple clauses, some of which depend on another clause to express a complete thought.

For this reason, sentences can be further classified based on their structure:

1. Simple sentences are composed of a single independent clause.
2. Complex sentences combine an independent clause with one or more dependent (also known as subordinate) clauses.
3. Compound sentences combine two independent clauses using a coordinating conjunction.
4. Compound-complex sentences have at least two independent clauses and at least one subordinate clause.

In "A Comprehensive Grammar of the English Language" [6], seven types of clause structure are recognized:

1. SV: subject + (intransitive) verb
2. SVO: subject + (monotransitive) verb + (direct) object
3. SVC: subject + (copular) verb + complement
4. SVA: subject + (copular) verb + adverbial
5. SVOO: subject + (ditransitive) verb + (indirect) object + (direct) object
6. SVOC: subject + (complex transitive) verb + (direct) object + complement
7. SVOA: subject + (complex transitive) verb + (direct) object + adverbial

Verbs. Verbs can also be classified: the most general categories are *lexical* or *auxiliary*.

Lexical verbs, also known as main verbs, express a clear meaning and do not rely on any other verb to be grammatical. For example, if we consider the sentence "Alice reads a book.", "reads" is a lexical verb.

Auxiliary verbs, also known as helping verbs, do not contain much meaning and are instead used to show grammatical features (e.g., tense, aspect, or modality). They are usually placed directly before a lexical verb. For example, if we consider the sentence "Alice has read a book.", "has" is an auxiliary verb (and "read" is a lexical verb).

There are also more specific manners for classifying verbs, such as considering transitivity (i.e. whether the verb requires an object or not to complete its meaning) and the copula (i.e. a verb is copular if it requires a predicative complement that provides information about the subject).

2.2. Related Work

This section reviews the state of the art in (controlled) natural language to logic translation, focusing on three different approaches (symbolic, neuro-symbolic, and neural) and the available datasets and benchmarks.

Symbolic approaches represent the earliest line of research and are conceptually closer to Talk2Prolog.

An overview of the works analysed for the state of the art is shown in Figure 2.

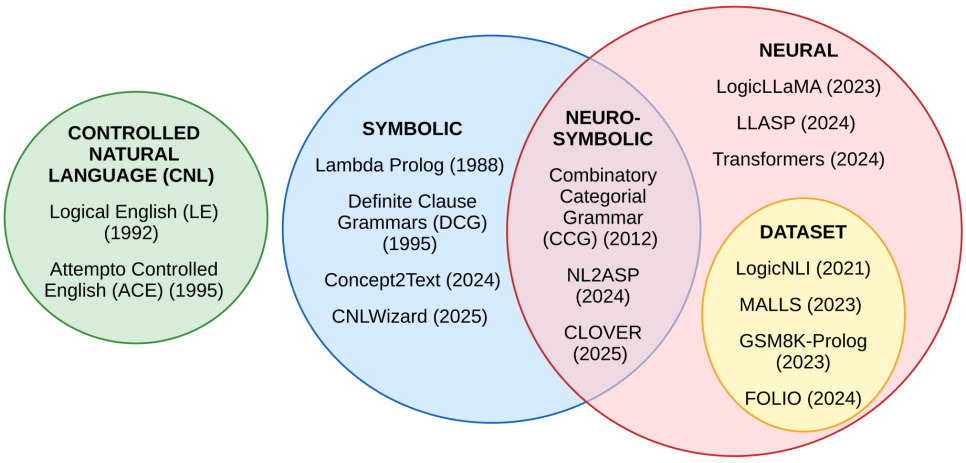

Figure 2: Overview of the Talk2Prolog related works.

Our main research question reflects our overall goal and drives the review of the state of the art. We aim to understand the following:

(RQ) How to formalize Prolog programming practices in order to develop an implementation capable of translating natural language sentences into Prolog expressions?

To conduct an efficient review, it is necessary to re-articulate the research question (RQ) in terms of more detailed secondary research questions:

- (Q1) What methodologies and tools are actually available for translating natural language text to a formal language of a logical system, and how do they compare in terms of accuracy and efficiency?
- (Q2) How do logic translation methodologies and tools handle natural language ambiguity?
- (Q3) What datasets and benchmarks, if any, are commonly used to evaluate logic translation tools?

Controlled natural language. Controlled natural languages (CNL) [7] are subsets of natural languages that can be accurately and efficiently processed by a computer, being expressive enough to allow natural usage by non-specialists. Attempto Controlled English (ACE) [8] is an example of CNL. In particular, it is a precisely defined subset of English that can automatically and unambiguously be translated into first-order logic. Thus, ACE is human- and machine-comprehensible. User specifications are translated into Prolog clauses using Definite Clause Grammar (DCG) [9] extended with feature structures implemented through GULP (Graph Unification Logic Programming) [10]. Discourse Representation Theory (DRT) is employed to resolve inter-text references, such as anaphora. The resulting Prolog clauses populate a knowledge base that can be queried through controlled natural language. Although ACE was originally intended to specify software, it has been used as a general knowledge representation language in several application domains. It translates ACE texts into Discourse Representation Structures (DRS) and, optionally, into Prolog. This knowledge base can be queried in ACE for verification and executed for simulation, prototyping, and validation of the specifications.

Another example of CNL is Logical English (LE) [11, 12]. As ACE, also LE is a subset of English that is both human- and machine-comprehensible. LE found different applications from law and education scenarios [13] [14] to financial ones [15].

While ACE is intended for general-purpose representation and reasoning, LE is syntactic sugar for programs written in a logic programming language, such as Prolog, Datalog [16], or Answer Set Programming (ASP [17]).

Symbolic approaches. Symbolic approaches can be found in different research areas such as knowledge representation, logic and reasoning. In 1988, Kreuger published the paper "A Higher Order Logic Parser for Natural Language Implemented in Lambda Prolog" [18] where he describes an implementation for F.C.N. Pereira's idea of using a sequent-calculus like system to produce Montague semantics [19] from natural language. Kreuger's implementation uses Lambda Prolog, a higher-order Prolog extension that supports the lambda-tree syntax approach to representing and manipulating formal syntactic objects [20]. In 2024, Bertini et al. [21] addressed the inverse task of Talk2Prolog by translating abstract concepts into natural language. Their system, Concept2Text, is designed as an explainable and modular alternative to LLM-based generation, emphasizing transparency and interpretability. More recently in 2025, Caruso et al. [22] proposed a framework that automatically generates grammars for translating controlled natural language into knowledge representation formalisms such as ASP. The implementation, named CNLWizard, processes a YAML-based specification and generates the corresponding grammar and imperative functions to simplify grammar development.

Neuro-symbolic approaches. Neuro-symbolic approaches use both symbolic and neural components. In 2012, Zettlemoyer et al. [23] explored the problem of mapping natural language sentences to lambda calculus representations. The proposed algorithm learns a grammar that maps sentences to

logical forms using Combinatory Categorial Grammar (CCG). The paper also presents a probabilistic extension based on a log-linear model. This model represents a distribution over syntactic and semantic analyses conditioned on the input sentence. The paper [24], dating back 2024, presents the NL2CNL dataset, designed to develop and assess tools for ASP automatic coding, and the NL2ASP tool, a two-step architecture for generating ASP programs from natural language specifications, the first step uses neural machine translation for going from natural language to CNL and then an algorithmic solution to translate CNL to ASP. In 2025, the paper [25] presented CLOVER (Compositional First-Order Logic Translation and Verification), a neuro-symbolic approach aimed at improving logical reasoning in LLMs. It first parses natural language into logical dependency structures composed of atomic subsentences and their dependents. These components are sequentially translated into First Order Logic (FOL) formulas, and SAT-based verification algorithms are applied to ensure logical correctness.

Neural approaches. Neural approaches are based on neural networks and deep learning models. Presented in 2023, LogicLLaMA [26] is a LLaMA-7B model fine-tuned for translating natural language sentences into first-order logic. The authors introduce also MALLS, a dataset of 34K NL-FOL pairs generated with GPT-4 to train the model. LLASP [27], published in 2024, is a lightweight model fine-tuned for ASP. Trained on a dataset of ASP problem specifications, LLASP demonstrates strong performance in generating semantically correct ASP programs, outperforming larger LLMs in some cases. In 2024, Chaturvedi and Asher [28] investigated the difficulty language models face in correctly identifying predicate–argument structures. The study evaluates two tasks, (1) question answering and (2) FOL translation, under prompting and fine-tuning regimes. Results show that FOL translation is more effective for learning predicate–argument structures, as it makes hallucinations easier to detect and fully specifies semantic relationships.

Datasets and benchmarks. Logical Natural Language Inference (LogicNLI)[9] (2021) is a benchmark designed for diagnosing first-order logic (FOL) reasoning capabilities in language models [29]. The dataset separates FOL reasoning from commonsense inference and evaluates models across accuracy, robustness, generalization, and traceability. Experiments on BERT, RoBERTa, and XLNet reveal significant limitations in FOL reasoning. GSM8K-Prolog[10] [30] (2023) is a Prolog-annotated version of the GSM8K dataset, designed to train a LLaMA-7B model to generate Prolog predicates from natural language mathematics problems. Each instance includes an instruction, the math question in natural language, and the corresponding Prolog program that solves it. FOLIO[11] (2024) is a human-annotated dataset for logically complex reasoning in natural language with FOL annotations. It includes 1430 conclusions paired with 487 premise sets for deductive reasoning. Experiments show that a subset of the dataset remains challenging even for GPT-4.

Analysis

- (Q1) There are neural, symbolic, and hybrid approaches available to translate natural language text to a formal language of a logical system. The earliest research line mainly follows a symbolic approach, which involves complex translation mechanism that might be hard to implement and does not yield results as generalizable as those obtained by neural approaches, being constrained to the use of controlled natural language. However, symbolic approaches are often more portable, transparent, and lightweight than neural ones. Neural approaches rely on training data and offer more general solutions, although these technologies still struggle with logical reasoning and consistency.
- (Q2) Logic translation methodologies and tools often handle natural language ambiguity by using controlled natural language whereas neural approaches tend to be more flexible, using LLMs to disambiguate the original input text.

[9] https://huggingface.co/datasets/tasksource/LogicNLI, accessed on July 13, 2026.
[10] https://huggingface.co/datasets/Thomas-X-Yang/gsm8k-prolog, accessed on July 13, 2026.
[11] https://huggingface.co/datasets/yale-nlp/FOLIO, accessed on July 13, 2026.

- (Q3) Datasets and benchmarks exist for FOL and ASP. For Prolog, the only dataset available at the time of writing is GSM8K-Prolog but it does not focus on the direct correspondence between English text and Prolog representation that we needed. This is the reason why we designed and implemented our own dataset, presented in the next section.

3. Dataset

Although the datasets presented in the previous section informed the development of our methodology, none fully matched our domain requirements, leading us to construct a dedicated dataset.

We needed Prolog facts, rules, and queries paired with their corresponding sentences expressed in natural language. These pairs needed to be as diverse as possible, in order to represent – in an agnostic way – common Prolog programming practices and use them as a benchmark for both Talk2Prolog and GPT5.2.

We created a dataset for achieving this purpose based on Prolog resources, mainly online tutorials and manuals for the languages.

The dataset consists of twelve sheets, each containing English-Prolog pairs, for a total of 858 entries: facts websites (79 rows); facts mix (257 rows); facts variables (3 rows); queries websites (34 rows); queries mix (21 rows); rules websites (17 rows); rules mix (21 rows); 0-arity facts (GPT) (79 rows); 1-arity facts (GPT) (98 rows); 2-arity facts (GPT) (97 rows); running examples (48 rows); semantic examples (104 rows).

The word "Websites" indicates that both the English sentence and the corresponding Prolog expressions are from online resources, such as tutorials and lessons on Prolog. The "Facts Variables" was also created in the same way, but was made to contain all facts that had variables.

The word "Mix" instead indicates that the Prolog expressions are from online resources, while the English meaning is given by GPT5.2.

Finally, the word "GPT" indicates that only GPT was used to generate both the sentence and the Prolog expression.

The GPT completion of the data was driven by the idea that LLMs work as powerful synthesis tools of all the possible practices present in its training data, and hence seemed suitable for the scope of this dataset.

Some examples of facts, queries, and rules from the dataset can be seen in Table 1, Table 2 and Table 3, respectively.

Prolog	English	Arity
sunny.	it is sunny	0
cat(fubby).	fubby is a cat	1
eats(fred,oranges).	Fred eats oranges	2
odd_number(5).	5 is an odd number.	1

Table 1
Examples from the Facts Websites sheet.

Prolog	English	Arity	Variables
?- food(pizza).	Is pizza a food?	1	0
?- studies(charlie, What).	What does charlie study?	2	1
?- likes(john, mary).	Does john like mary?	2	0
?- loves(Who, What).	Who loves what?	2	2

Table 2
Examples from the Queries Websites sheet.

The last two sheets, running and semantic examples, were handcrafted by us. The running examples were created to cover different types of Prolog expressions based on our methodology. The semantic

Prolog	English	Variables
happy(lili) :- dances(lili).	Lili is happy if she dances.	0
goToPlay(ryan) :- isClosed(school), free(ryan).	Ryan will go to play if school is closed, and he is free.	0
happy(X) :- likes(X, pizza).	X is happy if X likes pizza	1
hates(X,Y) :- not(likes(X,Y)).	X hates Y if X does not like Y.	2

Table 3
Examples from the Rules Websites sheet.

examples instead were created to verify the consistency of the translations by verifying the result of querying goals related to previous translated facts and rules, as shown in Table 4.

Sentences	Result	Prolog
If X is a human, then X is mortal.		mortal(X) :- human(X).
Socrates is a human.		human(socrates).
Is Socrates mortal?	TRUE	?- mortal(socrates).
It is sunny if the sky is blue.		sunny :- blue(sky).
Is it sunny?	FALSE	?- sunny.
Alice goes to sleep if she is tired and stressed.		goes_to(alice, sleep) :- tired(alice) , stressed(alice).
Alice is stressed.		stressed(alice).
Does Alice go to sleep?	FALSE	?- goes_to(alice, sleep).
Bob cries if Alice or Bob are upset.		cries(bob) :- upset(alice) ; upset(bob).
Does Bob cry?	FALSE	?- cries(bob).
The spider bites Peter.		bites(spider, peter).
Peter is bitten by a spider.		bites(spider, peter).
Who is bitten by a spider?	peter, peter	?- bites(spider, Who).
Charlie studies literature.		studies(charlie, literature).
What does Charlie study?	literature	?- studies(charlie, What).

Table 4
Semantic examples sheet.

Inconsistencies between the Prolog representation and the corresponding English meaning, and small grammar mistakes found in the online resources listed in the dataset were manually corrected.

In order to check whether English-Prolog pairs in the dataset were agreed upon by Prolog users, we submitted 15 out of our 48 running examples to Prolog experts, average users, and new learners to have the opportunity to receive their feedback. We received 33 answers in total, with an average agreement on the Prolog translation greater than 60%, apart from one specific sentence that we removed from the dataset. More details on the dataset validation stage can be found in [5].

All resource links are specified in the dataset, which is freely available to the research community on the Talk2Prolog GitHub Repository. To the best of our knowledge, no public dataset as the one we have developed exists. We hope that it may turn out to be useful for the research community.

4. Methodology

In this section we introduce the methodology that we followed to design and implement Talk2Prolog. It describes a systematic and reproducible procedure to translate a sentence expressed in formal, simplified English, into a plausible and practice-compliant Prolog representation, which could be a fact, a rule, or a query.

By plausible, we mean that the *semantic* information of the original sentence is retained as much as possible in the resulting Prolog expression, allowing the user to query and assert predicates without compromising consistency, real-world applicability, and human interpretability. This fulfills the necessity to infer new information based on already given knowledge.

By practice-compliant instead, we mean that the *syntax* of the representation is not only coherent with Prolog syntax rules, but also based on common Prolog practices.

This methodology consists of a set of rules that standardize the practices often adopted informally by Prolog users, paving the way to semantic interoperability of different Prolog knowledge bases, and was created based on the real examples retrieved through the dataset and the feedback provided by the survey mentioned in the previous section.

Based on this feedback we made the following assumptions on the sentences and the resulting Prolog translations:

- We only considered sentences in present tense, although the methodology could be applied in the same way for the past simple.
- Both definite and indefinite articles were discarded from the final result, since they are not common in Prolog representations.
- We only considered resulting terms with an arity ranging from zero to two.
- We disregarded sentences containing modal verbs, since they would require modal logic to be represented, going beyond Prolog expressivenes.

Due to our background, the main use case we considered is the possibility of translating natural language sentences prompted by a user to Prolog expressions that could be used inside a multi-agent systems, such as belief-desire-intention (BDI) programs [31, 32] implemented in Jason [33].

Considering a BDI architecture, facts and rules could represent beliefs, while queries correspond to test goals. This focus influenced certain decisions over others for the approach taken in the methodology translation. For example, in BDI agents implemented using Jason, the default in most existing examples and demos is not using any sort of temporal logic to model time, since the information is considered to be continuously updated when needed, being added when new or removed when no more relevant.

The rules of methodology can be summarized as follows.

4.1. Simple sentences

We consider positive declarative sentences.

- SV clauses become 1-arity facts of the form V(S). For example, the sentence “The cat sleeps.” becomes the Prolog expression `sleeps(cat).` See Figure 3.
- SVA clauses become 2-arity facts of the form V(S, A). For example, the sentence “Alice speaks calmly.” becomes the Prolog expression `speaks(alice, calmly).` See Figure 4.
- SVO clauses become 2-arity facts of the form V(S, O). For example, the sentence “Bob likes coffee.” becomes the Prolog expression `likes(bob, coffee).` See Figure 5.
- SVC clauses become 1-arity facts of the form C(S) if V is the verb “to be”. For example, the sentence “Alice is a student.” becomes the Prolog expression `student(alice).` See Figure 6. Otherwise, it can be represented as a 2-arity fact of the form V(S, C) (for example, in the case of the verb “to seem”).
- Expletive constructions, which are clauses that have “dummy” subjects that exist only to satisfy a syntactic requirement (such as “it” or “there”) depend only on the complement to be translated. A very common dummy subject is when the word “it” is used to describe the weather. An example is the sentence “It is raining”, which becomes the Prolog expression `raining.` See Figure 7.

4.2. Interrogative sentences

Interrogative sentences become queries.

The methodology considers two types of interrogative sentences: yes/no questions and “Wh-” questions.

Yes/no questions use inversion or auxiliary verbs, which are ignored during translation, therefore the sentence is treated using the same rules seen for the declarative sentences, with the addition of the

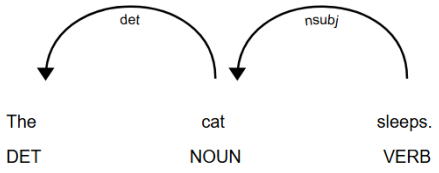

Figure 3: SV: The cat sleeps.

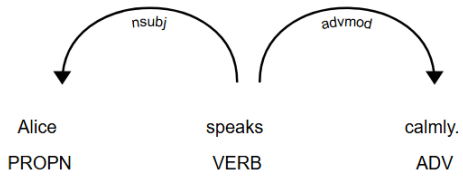

Figure 4: SVA: Alice speaks calmly.

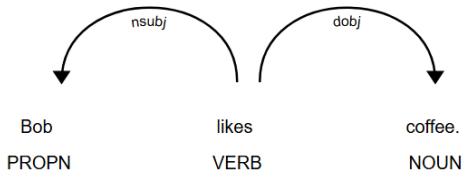

Figure 5: SVO: Bob likes coffee.

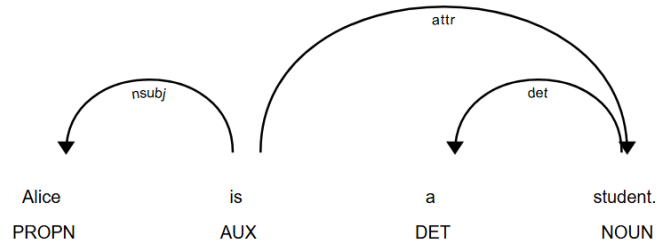

Figure 6: SVC: Alice is a student.

symbol ?– as a prefix for the final Prolog expression. For example, the yes/no question "Does Charlie prefer tea?" becomes ?– prefers (charlie , tea). See Figure 8.

"Wh-" questions are divided into subject and object questions. They start with a question word, such as "Who" and "What". In subject questions, the question word is the subject of the sentence, while in object questions, the question word is the complement object, and it uses inversion and an auxiliary verb. Again, both types of questions are treated as declarative sentences during translation, with the addition.

For example, the subject question "Who paints the canvas." becomes the Prolog expression ?– paints(Who, canvas)., while the object question "What does Bob paint?" becomes ?– paints(bob, What). See Figure 9 and Figure 10, respectively.

4.3. Complex sentences

Complex sentences become rules, where the independent clause becomes the head and the dependent clauses the body.

In this methodology, we only consider complex sentences that use the word "if" as subordinating

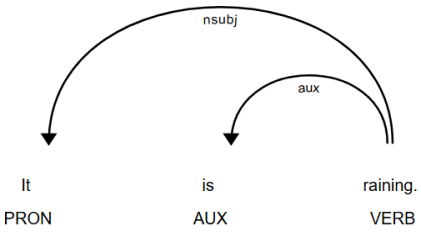

Figure 7: Expletive construction: It is raining.

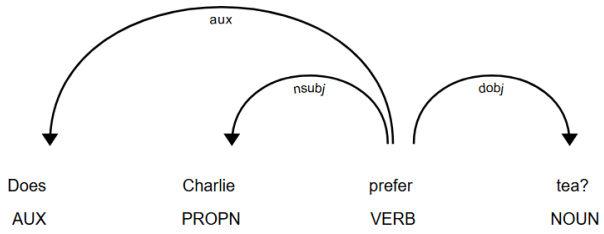

Figure 8: Yes/no question: Does Charlie prefer tea?

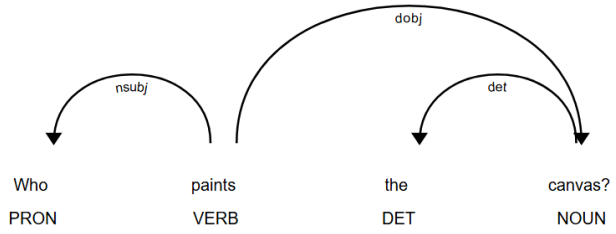

Figure 9: Subject question: Who paints the canvas?

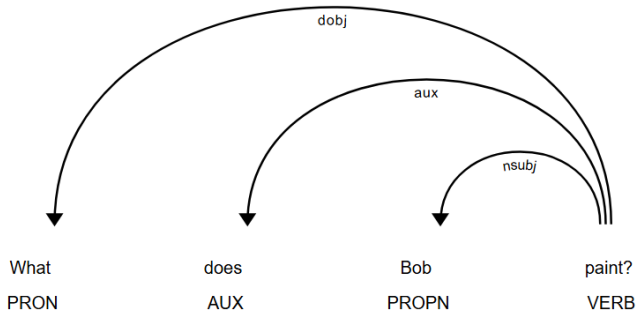

Figure 10: Object question: What does bob paint?

conjunction and with zero or one coordinating conjunction to join the dependent clauses, which must be “and” or “or”.

The independent clause is represented in Prolog using the same rules applied for the translation of simple sentences.

The subordinating conjunction “if” is represented as the logical implication symbol “:-” in Prolog, while the coordinating conjunctions “and” and “or” are represented in Prolog, respectively, with the symbols “,” and “;”.

The dependent clause might contain compound elements, such as:

1. Compound subject: When two or more nouns, pronouns, or noun phrases act together as the subject of the clause, usually connected by a coordinating conjunction.
2. Compound predicate: When the predicate of the clause has two or more verbs or verb phrases connected by a coordinating conjunction.
3. Compound subject complements: When two or more predicate nouns, pronouns, or adjectives appear after a linking verb.
4. Compound direct object: When two or more nouns receive the action of a transitive verb.

In the case of presence of compound elements, the dependent clause is divided into more individual clauses with one subject, one verb, and, if present, one direct object or subject complement. Anaphoric pronouns are substituted with their corresponding noun and subject-verb agreement is considered, in order to make the resulting clauses independent and grammatically correct.

These clauses correspond to the premises and are processed individually as simple sentences.

For example, the sentence "Bob is happy if he gets a good grade or learns a new topic." becomes happy(bob) :- gets(bob, good(grade)); learns(bob, new(topic)). See Figure 8

4.4. Other features of our translation methodology

Based on the practices we understood while creating the dataset, and feedback we collected via the survey, in our translation

- passive sentences are treated as active ones;
- anaphoric pronouns are resolved using preceding nouns when possible;
- indefinite pronouns, single capital letters (except for "I"), and questions words (e.g. "Who", "What") become variables;
- nesting is used for negative sentences or in the presence of two verbs or adjectives;
- subject-verb agreement is always considered;
- verb prepositions and compound names are concatenated using the underscore.
- quantities use a predefined predicate (e.g. "age", "quantity").

5. Conclusions and Future Works

The focus of this paper is on the software engineering stages that come before Talk2Prolog implementation and testing, namely understanding the Talk2Prolog context, and designing the development methodology, along with a dataset to run reproducible experiments.

Nevertheless, Talk2Prolog has been implemented, and experiments have been run to compare it against GPT5.2. We summarize the main findings here, directing the reader to the Talk2Prolog GitHub repository or to the EXTRAAMAS paper [5] for all the details.

As far as experiments aimed at measuring the capability to generate syntactically correct Prolog expressions are concerned, Talk2Prolog is less flexible than GPT5.2, accepting a more limited range of sentence inputs. In fact, it is more prone to generate a syntactically incorrect expression, or to not generate any expression at all.

On the other hand, when Talk2Prolog generates a syntactically correct Prolog expression, it is usually coherent with the other sentences it is able to translate. This is an extremely important feature to ensure semantic interoperability. It means that Prolog programs generated by Talk2Prolog in different moments, by different users, starting from slightly different sentences, will still be coherent and interoperable, because Talk2Prolog is always true to itself. This does not happen with GPT5.2, that – if fed with one sentence in each different prompt ("online mode") and not with all the relevant sentences together in the same prompt ("offline mode") – is not coherent with itself across different runs of experiments. As a result, Talk2Prolog returned more correct results than GPT5.2 in comparable (online) semantic experiments, since representing the meaning of the original sentence and keeping the consistency in the generated expressions are particularly relevant for these tasks.

In time performance testing, Talk2Prolog typically outperforms GPT5.2 in speed, which is not surprising at all.

While we are encouraged by our experiments with Talk2Prolog, we also acknowledge its many limitations, ranging from restricted linguistic coverage of the methodology – that heavily depends on handcrafted linguistic rules and assumptions – to scalability and robustness with unconstrained language.

While extending the system to broader grammatical constructions, multilingual settings, or richer logical representations may require substantial manual engineering effort, we believe that the effort is worth and we are heading towards these objectives. On the other hand, we are also exploring domains where the limited and controlled English sentences that Talk2Prolog can translate right now, are almost enough for the domain applications. Together with Domenico Spingardi from Grinding Technologies and the Master student Sara Samiei, we have been working on natural language commands given to grinding machines. In that case, we used Rasa [34] as the main tool to perform the translation and we compared it with some state of the art LLMs. We believe that Talk2Prolog would be an extremely suitable alternative, given the simplicity and regular structure of those commands[12]. At the time of writing, the master thesis by Sahar Harati is exploring translation of natural sentences into Prolog, in the protein domain. While this thesis is mainly oriented towards experiments with Agentic AI tools, an evaluation of Talk2Prolog on the protein restricted domain is under way.

The opinions collected in the dataset validation survey mentioned in Section 3 are another rich source of suggestions for the future developments.

One of the most suggested features was the inclusion of arithmetic comparison operators in the translation, to allow the user to perform calculations using numerical values, with the expression “greater than” (resp. “less than”, “equal to”, etc...) being represented by the operator “>” (resp. “<”, “=:=”) in Prolog.

This idea could be further extended to similar expressions, such as “more than” being translated in the same way as “greater than”.

Many issues raised in the survey regarded the lack of context for the sentences. For this reason, a possible extension would be to manage texts containing more than one sentence and keep track of references.

Another way to improve the methodology is to distinguish between proper and common nouns. Common nouns can be used as term names, wrapping an id or a name. Common nouns can also be translated as variables, if preceded by an indefinite article.

A further extension could be adding words and collocations to express the degree of certainty of a sentence, such as “maybe”, “definitely”, and “likely”, associating a value representing the certainty of the representation with their Prolog representations. This addition could possibly be represented using ProbLog [35], a probabilistic extension of Prolog where every clause is labeled with a probability.

Declaration on Generative AI

The contents presented in this paper are the result of a creative and original work designed and carried out by the first author, Laura Fuciarelli, under the supervision and guidance of the other three authors.

We used Claude, Anthropic’s AI assistant[13] to speed up (i) the drawing of Figure 1; (ii) the rewording of a few sentences (less than 10% of the final document) in order to avoid overlaps with the companion EXTRAAMAS paper; (iii) the summarization of some contents taken either from Laura Fuciarelli’s Master Thesis, or from online resources (again, less than 10% of the final document).

Claude’s results were manually reworded and reworked by the authors, to align them with the style and purposes of the paper.

The authors are the sole responsible for the contents of this paper.

[12] See Sara Samiei Master Thesis, https://github.com/sarasamiee/nl2gcode-cnc-thesis, accessed on July 13, 2026.
[13] https://claude.ai/new, accessed on July 13, 2026.

References

[1] D. C. Dennett, The role of language in intelligence, in: J. Khalfa (Ed.), What is Intelligence?, Cambridge University Press, 1994.

[2] M. Pagel, Q&A: What is human language, when did it evolve and why should we care?, BMC Biology 15 (2017) 64. URL: https://doi.org/10.1186/s12915-017-0405-3. doi:10.1186/s12915-017-0405-3.

[3] S. Johansson, The Dawn of Language: How We Came to Talk, MacLehose Press, 2021.

[4] J. Lighthill, Artificial Intelligence: A general survey, Artificial Intelligence: a paper symposium (1973).

[5] L. Fuciarelli, A. Ferrando, A. Gatti, V. Mascardi, Talk2Prolog: from controlled natural language to Prolog, frugally and explainably, in: 8th International Workshop on EXplainable, Trustworthy, and Responsible AI and Multi-Agent Systems (EXTRAAMAS 2026), 2026.

[6] R. Quirk, S. Greenbaum, G. Leech, J. Svartvik, A comprehensive grammar of the English language, London: Longman, 1985.

[7] T. Kuhn, A survey and classification of controlled natural languages, Comput. Linguistics 40 (2014) 121–170. URL: https://doi.org/10.1162/COLI_a_00168. doi:10.1162/COLI_A_00168.

[8] R. Schwitter, B. Hamburger, N. E. Fuchs, Attempto: Specifications in controlled natural language, in: A. Krall, U. Geske (Eds.), 11. Workshop Logische Programmierung, Technische Universität Wien, 27.-29. September 1995, Proceedings. GMD-Studien Nr. 270, 1995, pp. 151–160.

[9] F. C. Pereira, D. H. Warren, Definite clause grammars for language analysis—a survey of the formalism and a comparison with augmented transition networks, Artificial intelligence 13 (1980) 231–278.

[10] M. A. Covington, An extension of prolog for unification-based grammar (1989).

[11] R. A. Kowalski, Legislation as logic programs, in: G. Comyn, N. E. Fuchs, M. Ratcliffe (Eds.), Logic Programming in Action, Second International Logic Programming Summer School, LPSS '92, Zurich, Switzerland, September 7-11, 1992, Proceedings, volume 636 of *Lecture Notes in Computer Science*, Springer, 1992, pp. 203–230. URL: https://doi.org/10.1007/3-540-55930-2_15. doi:10.1007/3-540-55930-2_15.

[12] R. Kowalski, Logical English, Proceedings of Logic and Practice of Programming (LPOP) (2020). URL: https://www.doc.ic.ac.uk/~rak/papers/LPOP.pdf.

[13] G. Sartor, J. A. Dávila, M. Billi, G. Contissa, G. Pisano, R. A. Kowalski, Integration of logical English and s(CASP), in: J. Arias, R. Calegari, L. Dickens, W. Faber, J. Fandinno, G. Gupta, M. Hecher, D. Inclezan, E. LeBlanc, M. Morak, E. Salazar, J. Zangari (Eds.), Proceedings of the International Conference on Logic Programming 2022 Workshops co-located with ICLP 2022, volume 3193 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022. URL: https://ceur-ws.org/Vol-3193/paper5GDE.pdf.

[14] R. A. Kowalski, J. A. Dávila, G. Sartor, M. Calejo, Logical English for law and education, in: D. S. Warren, V. Dahl, T. Eiter, M. V. Hermenegildo, R. A. Kowalski, F. Rossi (Eds.), Prolog: The Next 50 Years, volume 13900 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 287–299. URL: https://doi.org/10.1007/978-3-031-35254-6_24. doi:10.1007/978-3-031-35254-6_24.

[15] R. A. Kowalski, A. Datoo, Logical English meets legal English for swaps and derivatives, Artif. Intell. Law 30 (2022) 163–197. URL: https://doi.org/10.1007/s10506-021-09295-3. doi:10.1007/S10506-021-09295-3.

[16] S. Ceri, G. Gottlob, L. Tanca, What you always wanted to know about Datalog (and never dared to ask), IEEE Trans. Knowl. Data Eng. 1 (1989) 146–166. URL: https://doi.org/10.1109/69.43410. doi:10.1109/69.43410.

[17] V. Lifschitz, Answer set programming, volume 3, Springer Cham, 2019.

[18] P. Kreuger, A higher order logic parser for natural language implemented in Lambda Prolog, in: T. O'Shea, V. S. Sgurev (Eds.), Artificial Intelligence III: Methodology, Systems, Applications - Proceedings of the Third International Conference on Artificial Intelligence: Methodology, Systems, Applications, AIMSA 1988, Varna, Bulgaria, September 20-23, 1988, North-Holland, 1988,

pp. 299–306.
[19] D. R. Dowty, R. Wall, S. Peters, Introduction to Montague semantics, Springer Science & Business Media, 2012.
[20] X. Qi, An implementation of the language Lambda Prolog organized around higher-order pattern unification, CoRR abs/0911.5203 (2009). URL: http://arxiv.org/abs/0911.5203. arXiv:0911.5203.
[21] F. Bertini, A. D. Palù, F. Fabiano, A. Formisano, F. Zaglio, Concept2Text: An explainable multilingual rewriting of concepts into natural language, in: E. D. Angelis, M. Proietti (Eds.), Proceedings of the 39th Italian Conference on Computational Logic, Rome, Italy, June 26-28, 2024, volume 3733 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3733/paper14.pdf.
[22] S. Caruso, C. Dodaro, M. Maratea, A. Tarzariol, A general framework for representing controlled natural language sentences and translation to KR formalisms, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025, ijcai.org, 2025, pp. 4401–4409. URL: https://doi.org/10.24963/ijcai.2025/490. doi:10.24963/IJCAI.2025/490.
[23] L. S. Zettlemoyer, M. Collins, Learning to map sentences to logical form: structured classification with probabilistic categorial grammars, in: Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence, UAI'05, AUAI Press, Arlington, Virginia, USA, 2005, p. 658–666.
[24] M. A. B. Santana, I. Kareem, F. Ricca, Towards automatic composition of ASP programs from natural language specifications, in: Proceedings of IJCAI 2024, Jeju, South Korea, August 3-9, 2024, ijcai.org, 2024, pp. 6198–6206. URL: https://www.ijcai.org/proceedings/2024/685.
[25] H. Ryu, G. Kim, H. S. Lee, E. Yang, Divide and translate: Compositional first-order logic translation and verification for complex logical reasoning, in: Y. Yue, A. Garg, N. Peng, F. Sha, R. Yu (Eds.), International Conference on Learning Representations, volume 2025, 2025, pp. 24935–24964. URL: https://proceedings.iclr.cc/paper_files/paper/2025/file/3e592c571de69a43d7a870ea89c7e33a-Paper-Conference.pdf.
[26] Y. Yang, S. Xiong, A. Payani, E. Shareghi, F. Fekri, Harnessing the power of large language models for natural language to first-order logic translation, in: L. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, Association for Computational Linguistics, 2024, pp. 6942–6959. URL: https://doi.org/10.18653/v1/2024.acl-long.375. doi:10.18653/V1/2024.ACL-LONG.375.
[27] E. Coppolillo, F. Calimeri, G. Manco, S. Perri, F. Ricca, LLASP: fine-tuning large language models for answer set programming, in: P. Marquis, M. Ortiz, M. Pagnucco (Eds.), Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024, 2024. URL: https://doi.org/10.24963/kr.2024/78. doi:10.24963/KR.2024/78.
[28] A. Chaturvedi, N. Asher, Learning semantic structure through first-order-logic translation, in: Y. Al-Onaizan, M. Bansal, Y. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024, volume EMNLP 2024 of *Findings of ACL*, Association for Computational Linguistics, 2024, pp. 6669–6680. URL: https://doi.org/10.18653/v1/2024.findings-emnlp.390. doi:10.18653/V1/2024.FINDINGS-EMNLP.390.
[29] J. Tian, Y. Li, W. Chen, L. Xiao, H. He, Y. Jin, Diagnosing the first-order logical reasoning ability through LogicNLI, in: M. Moens, X. Huang, L. Specia, S. W. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Association for Computational Linguistics, 2021, pp. 3738–3747. URL: https://doi.org/10.18653/v1/2021.emnlp-main.303. doi:10.18653/V1/2021.EMNLP-MAIN.303.
[30] X. Yang, Y. Tam, Exploring an LM to generate prolog predicates from mathematics questions, CoRR abs/2309.03667 (2023). URL: https://doi.org/10.48550/arXiv.2309.03667. doi:10.48550/ARXIV.2309.03667. arXiv:2309.03667.
[31] A. S. Rao, M. P. Georgeff, BDI agents: From theory to practice, in: Proceedings of the First

International Conference on Multiagent Systems, June 12-14, 1995, San Francisco, California, USA, The MIT Press, 1995, pp. 312–319.

[32] A. S. Rao, AgentSpeak(L): BDI agents speak out in a logical computable language, in: 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Eindhoven, The Netherlands, January 22-25, 1996, volume 1038 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 42–55. URL: https://doi.org/10.1007/BFb0031845. doi:10.1007/BFb0031845.

[33] R. Bordini, J. Hübner, M. Wooldridge, Programming Multi-Agent Systems in AgentSpeak Using Jason, volume 8, John Wiley & Sons, Ltd, United Kingdom, 2007. doi:10.1002/9780470061848.

[34] T. Bocklisch, J. Faulkner, N. Pawlowski, A. Nichol, Rasa: Open source language understanding and dialogue management, CoRR abs/1712.05181 (2017). URL: http://arxiv.org/abs/1712.05181. arXiv:1712.05181.

[35] L. D. Raedt, A. Kimmig, H. Toivonen, Problog: A probabilistic prolog and its application in link discovery, in: M. M. Veloso (Ed.), IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007, 2007, pp. 2462–2467. URL: http://ijcai.org/Proceedings/07/Papers/396.pdf.