
Using Large Language Models to Translate English Questions into Abductive Reasoning Problems

Andrew S. Gordon

GORDON@ICT.USC.EDU

Institute for Creative Technologies, University of Southern California, Los Angeles, CA 90094 USA

Abstract

The 100-question Triangle-COPA evaluation was an early benchmark for logic-based commonsense reasoning systems inspired by the 1000-question Choice of Plausible Alternatives evaluation, but where English multiple-choice questions were represented as conjunctions of first-order logical literals. Large Language Models (LLMs) have been able to achieve perfect scores on the Triangle-COPA evaluation for several years using only the English glosses of questions rather than their logical forms. In this research, we explore an alternative approach where LLMs are used to automatically convert the English questions into logical form to be solved with a logic-based abductive reasoning engine. As a knowledge base, we likewise use LLMs to automatically convert English glosses of relevant domain knowledge into axioms in first-order logic. Our experiments show consistent use of logical forms in question and knowledge formalization for use in automated abductive reasoning, with correct solutions often matching inferences derived from hand-authored formalisms. Although this approach is not competitive with traditional LLM usage on this benchmark, our experiments inform future efforts to integrate symbolic reasoning subsystems into contemporary AI agents.

1. Introduction

Progress in automated commonsense reasoning was supported in part by challenging benchmarking evaluation question sets, e.g., the Choice Of Plausible Alternatives (COPA) evaluation (Roemmele et al., 2011). The 1000 natural language multiple-choice questions in this evaluation were difficult for AI systems when it was introduced in 2011, but steady advancements in empirical natural language processing over ten years and advances in generalized pretrained transformers effectively solved this challenge (Radford et al., 2018; Brown et al., 2020). Today, LLMs of various sizes achieve human-level performance on this commonsense reasoning benchmark.

Aiming to scaffold analogous progress in formal commonsense reasoning and visual understanding, Maslan et al. (2015) created the Triangle-COPA evaluation in 2015. The 100 questions in Triangle-COPA mirrored the format of the previous COPA evaluation, but augmented each question with both its formal representation as first-order logical literals and its visual representation as an animated movie in the abstract style of the classic Heider-Simmel experimental film (Heider & Simmel, 1944) depicting two triangles and a circle moving around a box with a door. Soon after its introduction, Gordon (2016) demonstrated that nearly all Triangle-COPA questions (91 of 100) could be solved using probability-ordered logical abduction (Etcetera Abduction) coupled with a

hand-authored knowledge base of 252 commonsense axioms encoded as definite clauses in first-order logic.

As attention in commonsense reasoning research continued to shift away from formal logic over the last decade, the logic representations in the Triangle-COPA evaluation garnered little further research interest. Chemburkar et al. (2024) showed that a frontier model from 2023 could answer 90 Triangle-COPA questions correctly using only the English text, and could achieve a perfect score if prompted to anthropomorphize the shapes referenced in each question as if they were people. With excellent performance on natural language reasoning and continued progress on visual reasoning, continued pursuit of logic-based approaches seemed of questionable value. Although the benefits of transparent and verifiably-correct reasoning were clear, the knowledge bottleneck remained. The labor costs and expertise required to hand-author formal axioms in robust knowledge bases would prevent automated reasoning from being a viable tool across a broad range of problems at scale.

In this paper, we describe an alternative approach to solving Triangle-COPA problems that integrates LLMs and logical abductive reasoning. First, we explore whether an LLM can automatically convert the natural language representations of Triangle-COPA questions into logical form as conjunctions of first-order logical literals using predicate-argument forms of its own invention. Second, we explore whether domain knowledge expressed in natural language can likewise be formalized by an LLM into logical axioms of first-order definite clauses, again using LLM-devised logical forms. Third, we attempt to solve these formalized questions using the formalized knowledge base using probability-ordered logical abduction (Etcetera Abduction), and compare the structure of the resulting proofs obtained using the hand-authored knowledge base of Gordon (2016). Our experiments show consistent use of logical forms in question and knowledge formalization for use in automated abductive reasoning, with correct solutions often matching inferences derived from hand-authored formalisms. Although this approach is not competitive with traditional LLM usage on this benchmark, our experiments inform future efforts to integrate symbolic reasoning subsystems into contemporary AI agents.

2. Related Work

A longstanding goal in natural language processing research has been the automatic conversion of natural language statements into formal representations, i.e., semantic parsing, in support of the fluid integration of natural language and automated reasoning methods. Innumerable computational approaches have been pursued over the past many decades, highlighted by key advances in frame semantics (Schank, 1975), annotated treebanks (Marcus et al., 1993), statistical parsing of syntax (Magerman, 1995), broad-coverage lexical databases (Fellbaum, 2005), and the automatic generation of large-scale commonsense knowledge bases from lexical resources (Ovchinnikova, 2012). In retrospect, the history of this line of research owes much of its success to the laborious curation of large linguistic resources (treebanks and semantic networks) and the statistical machine learning frameworks that enabled accurate pattern recognition over natural language text. Where large parallel corpora between language and logical forms can be curated, the conversion process between the two could be learned through statistical means.

However, recent approaches to semantic parsing have changed dramatically due to advances in LLMs (Radford et al., 2018; Brown et al., 2020), which are remarkably adept at dealing with both the structure and semantics of natural language despite being trained in a (largely) unsupervised manner, i.e., by minimizing the perplexity of next-word prediction over massive-scale text corpora. To apply language models in semantic parsing to convert natural language into a formal language, recent research has gravitated toward a two-step process termed Semantic Parsing via Paraphrasing (SPP) (Berant & Liang, 2014; Shin et al., 2021). In this framework, the first step is to use a language model to paraphrase a natural language utterance into a canonical form, i.e., a text-to-text translation where the target paraphrase of the input is trivially converted into its formal form, in the second step, using a formal grammar or set of hand-authored rules. Recent advances in this approach have sought to eliminate the need for fine-tuning large language models in the paraphrase step via clever sequences of engineered prompts (Yang et al., 2022) and improving compositional generalization by incorporating syntactical decomposition in prompt sequences (Drozдов et al., 2022).

Our present work differs from these previous efforts in the degree to which the target formalisms are known ahead of time. Aside from the basic structure of first-order literals and definite clauses, the representational vocabulary used in our translations are fully generated during translation, consisting of predicates and argument structure that are entirely LLM-devised. Our approach expands on our previous investigation of LLM-devised representational vocabulary (literal forms and domain constants), as demonstrated in the automated formalization of William Shakespeare’s poem *Venus and Adonis* (Gordon, 2025).

3. Background

3.1 The Triangle-COPA Benchmark

In this paper, we evaluate our approach to automated formalization of problems and knowledge bases using Triangle-COPA (Maslan et al., 2015), an early commonsense reasoning benchmark that pairs 100 questions with their formal representation in first-order logic and a short animated movie in the style of the classic Heider-Simmel experimental film (Heider & Simmel, 1944) authored using a tablet-based multi-touch animation tool (Gordon & Roemmele, 2014). Each Triangle-COPA question uses the three-part format of the earlier COPA evaluation (Roemmele et al., 2011), where a *given* part describes a narrative situation involving a Big Triangle, a Little Triangle, and Circle, and asks a *why*-question that requires an interpretation of the situation to answer correctly. This is followed by exactly two alternative answers, *option A* and *option B*, where the correct answer is the one unanimously preferred by human raters that were consulted during the creation of the benchmark.

In the original work, these three parts of each question were formalized, by hand, as one or more literals in first-order logic, using predicate-argument forms invented specifically for this evaluation. To allow for the representation of higher-order relationships between literals, including temporal ordering, these representations were authored using *eventuality notation*, a convention where a prime character at the end of a predicate name signifies that the first argument of the literal is the reification of the literal itself (Hobbs, 1985). In each *given* part of the formalism, these *eventualities* are represented as domain constants (e.g., E1, E2), which might be further described using un-primed

predicates for sequential (`seq`) and parallel (`par`) occurrence in the given narrative. Eventuality arguments represented in each option are left as universally quantified variables, facilitating their logical unification with Skolem constants that may appear in candidate interpretations identified via automated reasoning. In this research, we used the Triangle-COPA materials provided in its public repository¹. An example question of the Triangle-COPA evaluation is as follows:

83. *A small triangle and big triangle are next to each other. A circle runs by and pushes the small triangle. The big triangle chases the circle. Why does the big triangle chase the circle?*

```
(and (approach' E1 C LT) (push' E2 C LT) (chase' E3 BT C) (seq E1 E2 E3))
```

a. *The big triangle is angry that the circle pushed the small triangle, so it tries to catch the circle.*

```
(angryAt' e4 BT C)
```

b. *The big triangle and circle are friends. The big triangle wants to say hello to the circle.*

```
(and (friend' e5 BT C) (goal' e6 e7 BT) (greet' e7 BT C))
```

3.2 Etcetera Abduction

Abduction, distinct from deduction or induction, is a logical reasoning method that accepts as input a knowledge base of axioms and a set of logical literals (the *observables*), and answers the question: What sets of assumptions could be made such that, if they were true, they would logically entail the observables?

Early work on automated abductive reasoning by Hobbs et al. (1993) made three important contributions. First, they proposed that logical abduction was well suited to model the broad cognitive processes of interpretation, particularly for natural language understanding where the words of an utterance serve as the observables and its meaning identified as the entailing assumptions. Second, they introduced a mechanism for defeasibility of first-order axioms in abductive reasoning, where any uncertainties had to be explicitly assumed in special “etcetera literals” added to antecedents in definite clauses. Third, they offered an exhaustive abductive reasoning algorithm (Weighted Abduction) that ranked solutions by transferring “costs” from consequences to antecedents while back-chaining through knowledge base axioms.

Gordon (2016) expanded on these contributions by making a connection between the etcetera literals in Hobbs et al. (1993) and basic probability theory. In this formulation, the probability of an etcetera literal in the antecedent of a first-order definite clause is defined as the conditional probability of the consequent given the remaining antecedents, or as the prior probability of the consequent in the case that the etcetera literal is the only antecedent. For example, the following axioms encode prior and conditional probabilities for interpreting situations where someone might be angry at someone else (as in Question 83, above).

Axiom 1. *On rare occasions, someone might be angry at someone else. (prior probability)*

```
(if (etc0_angryAt 0.1 e x y)
    (angryAt' e x y))
```

Axiom 2. *If someone is trying to annoy you, you’ll likely be angry. (conditional probability)*

1. <https://github.com/asgordon/TriangleCOPA>

```
(if (and (annoy' e1 x y) (goal' e2 e1 x) (etc1_angryAt 0.8 e1 e2 e x y))
    (angryAt' e y x))
```

Axiom 3. *If someone attacks someone you like, you'll likely be angry. (conditional probability)*

```
(if (and (attack' e1 y z) (like' e2 x z) (etc3_angryAt 0.9 e1 e2 e x y z))
    (angryAt' e x y))
```

Gordon (2016) described Etcetera Abduction as an algorithm for probability-ordered logical abduction. The algorithm exhaustively back-chains from observable literals through knowledge base axioms to a given *depth*, collecting all solutions consisting of *only* etcetera literals (representing prior and conditional probabilities), and ranks these solutions by their joint probability, naïvely computed as the product of their probabilities. Much of the computational costs of the algorithm are consumed in attempting to unify etcetera literals in the cartesian product of each literal's antecedents. As such, substantial performance gains can be garnered by maintaining an *n-best* list of solutions during the combinatorial search, and avoiding the expensive unification work if the best result would fail to beat the least probable solution on the list.

3.3 Solving Triangle-COPA Questions with Etcetera Abduction

Gordon (2016) demonstrated that Triangle-COPA questions could be solved using Etcetera Abduction and a hand-authored knowledge base of 252 axioms encoded as first-order definite clauses. In total, 91 of the 100 questions were solved in this work, although some questions required special tuning of the algorithm's *depth* and *n-best* parameters to identify a correct solution.

There are at least two established methods to solve a Triangle-COPA question using Etcetera Abduction. Gordon (2016) used the first method, *n-best answer*, where an *n-best* list of interpretations are generated by providing the question's *given* part as observables to be explained using the knowledge base. Then each probability-ordered solution is matched against the question's two options using logical unification, and the first option to successfully unify with a solution is selected as the more plausible alternative. This method can fail to make a choice if neither alternative unifies with solutions in the *n-best* list, which is possible even when large values for *n* and *depth* are used. A second option-selection method, *more-probable answer*, avoids this problem by splitting the question into two separate interpretation problems. Here, two new sets of observables are generated as a conjunction of the question's *given* literals with those in each option. Then, the most probable (1-best) solution is identified for each conjunction, with the one with the highest joint-probability selected as the more plausible alternative. Intuitively, this method is successful when half of the observables (the *given*) can be explained by back-chaining through knowledge base axioms to the observables in the other half (the *option* literals). This method can fail to select an alternative only in the case when these joint probabilities are equal, which can easily occur when using very coarse probability estimates for prior and conditional probabilities in knowledge base axioms.

In the experiments described in this paper, we report results using both of these Triangle-COPA answer-selection methods, as implemented in the Python reference implementation of Etcetera Abduction provided in its public repository².

2. <https://github.com/asgordon/EtcAbductionPy>

4. Methodology

In this paper, we explore methods for using LLMs to automatically convert English expressions of Triangle-COPA questions and relevant domain knowledge into logical form, such that they can be solved using Etcetera Abduction. Our methodology consists of two main phases, question formalization and knowledge formalization, which are described in the following sections. An open-weight LLM was used in this research, namely the Qwen3.6-27B model with Multi-Token Prediction and 4-bit weights using dynamic K-XL quantization by Unsloth.ai³, and hosted locally using `llama.cpp` server⁴. The context window was set to 256K tokens, with each phase executed in its own context over multiple LLM requests. Algorithms used in both phases were written in Python and communicated to the LLM via the server API using the standard OpenAI-compatible endpoints.

4.1 Formalizing Triangle-COPA Questions

In the first phase, the 100 questions in the Triangle-COPA benchmark were automatically translated into logical form using only their English-language representations, where the specific predicates and argument structure of the representation were devised by the LLM itself. The task was structured as a multi-turn chat with the LLM, where each individual Triangle-COPA question was formalized by the LLM in a single turn. The initial prompt provided detailed instructions on the required format of a question’s *given*, *option A*, and *option B* formalizations, along with specific guidance on how to formulate novel predicate-argument forms as needed. In order to better mirror the original Triangle-COPA formalization, the LLM was instructed to use eventuality notation (primed predicates) with numbered eventuality constants in the *given* representation, and numbered universally-quantified eventuality variables in each option representation.

To guarantee that LLM generations were correctly formatted, we adopted the technique used in Gordon (2025) of requesting generations in a custom JSON format, and enforcing valid responses by constraining generation using custom JSON schemas. Schema-constrained generations (also known as Structured Model Outputs) is a popular option for both locally-hosted and commercially-hosted LLMs that enables developers to provide strict requirements on the contents of generated JSON objects, including required fields and constraints on contents using standard regular expression. The resulting JSON objects are then parsed and algorithmically translated to the desired output format. In this step of the question formalization phase, we authored a single JSON schema requiring *given*, *option A*, and *option B* fields, with values consisting of arrays of literals consisting of predicates and arguments. In the *given* section, arguments of literals were required (via regular expression) to be represented as (uppercase) constants, while option fields would accept mixed combinations of constants and variables. A post-processing step algorithmically converted these JSON generations into the S-expression format seen in the original Triangle-COPA question set. The whole JSON schema itself was provided verbatim to the LLM in the instructions of the initial prompt to further align expectations with constraints during generation.

After formalizing a question, the predicate-argument forms were scanned for the use of previously-unseen predicates or argument counts (arity). For each novel literal form used by the LLM, a

3. <https://huggingface.co/unsloth/Qwen3.6-27B-MTP-GGUF>

4. <https://github.com/ggml-org/llama.cpp>

follow-up call was made to the LLM to define the meaning of the predicate and the roles of its ordered arguments. For this task we adapted the vocabulary solicitation prompt and JSON schema (for a lifted literal) used in Gordon (2025). Our revised prompt included further instructions on definitions of literal forms that followed eventuality notation, which was not used in the previous work. Instructions to the LLM strongly encouraged the reuse of these novel literal forms in subsequent question formalizations.

At the completion of this phase, the 100 question formalizations were converted into the original Triangle-COPA format, and the LLM-devised vocabulary was serialized for reuse during the subsequent knowledge formalization phase.

4.2 Formalizing Domain Knowledge

Answering Triangle-COPA questions using Etcetera Abduction requires a knowledge base of commonsense axioms in the domain of interpersonal relations, emotions, and psychology. In our research, we investigate whether English expressions of this domain knowledge can be automatically translated into the format of the definite clauses used by Etcetera Abduction, to include etcetera literals in the antecedents that encode the appropriate prior and conditional probabilities. There has been some notable efforts to author axiomatic knowledge of psychology and interpersonal relations as English statements, including Heider’s seminal work (1958) and Smedslund’s *Psychologic* (1997). In future work, we envision that these and other English-language knowledge stores can serve as the starting point for the automated curation of appropriate domain knowledge. In our current work, however, we opted instead to author our own English glosses for the 252 axioms that were successful in solving Triangle-COPA problems in Gordon’s (2016) original work.

The original 252 axioms did include some English-language notation (mainly Lisp-style comments), but lacked any information that could serve as the basis for an axiom’s probability. Accordingly, we hand-authored 252 new English sentences to serve as the starting point for their formalization. While there was some effort to use diverse sentence structure and vocabulary, the resulting glosses tended to follow a handful of patterns, as exemplified in the following examples of prior-probability axioms (2 and 4) and conditional-probability axioms (1, 3, 5, and 6):

1. *When someone is a parent of a child, the child often accompanies the parent.*
2. *On occasion, a person might annoy someone else.*
3. *If you are angry at someone, you might chase them.*
4. *Once in a while, someone might hug another person.*
5. *If you get pushed by someone, it is possible it is because they are a bully.*
6. *One very likely reason to be surprised is if you see someone you didn’t expect.*

Using these 252 glosses as our informal knowledge base, we again employed a schema-constrained LLM to author the corresponding first-order definite clauses. In this phase, the initial prompt included detailed instructions about the parts of a valid definite clause (single consequent as a head, array of antecedents as a body), and encouragement to reuse literal forms it had devised during the question-formalization phase. This previous vocabulary was provided verbatim in the prompt as a serialized JSON object, as was the full definite clause JSON schema used to constrain the LLM’s output.

Importantly, we did not ask the LLM to include the required etcetera literal in its output, as it could be easily algorithmically derived and inserted into the antecedent during the final conversion of the JSON formalisms into S-expressions. Needed only was a numeric value for its first argument to represent the probability of the etcetera literal. For this, we included an additional field in the definite clause JSON schema for this numeric value, and added instructions in the prompt to translate linguistic certainty into probability with this guidance: “almost certain” ≈ 0.9 , “likely” ≈ 0.7 , “possible/might” ≈ 0.4 , “rarely/on occasion” $\approx 0.1\text{--}0.2$.

Although strongly encouraged to reuse earlier vocabulary, the LLM was instructed to devise new literal forms if needed. As in the previous phase, the output of the LLM was scanned for novel predicates or arities after each turn. When identified, the LLM was prompted to define them using the same prompts and schemas as before, and encouraged to reuse them in subsequent formalizations.

As a final step, we scanned all devised vocabulary and axioms for literal forms that did not have a prior probability axiom, which is required by the Etcetera Abduction software. For each of these forms, we prompted the LLM to define a new definite clause with an empty body (no antecedents), estimate its prior probability using the previous numerical guidance, and author an English gloss of the axiom in the same style as seen previously.

At the completion of this step, the JSON formulations of each definite clause were algorithmically converted into the S-expression format used by the Etcetera Abduction software, to include an etcetera literal in each antecedent with arguments for its probability and all universally quantified variables in the order they appear among the axiom’s other literals.

5. Results

We describe the results of applying our two-phase methodology to the Triangle-COPA question set and glosses of knowledge base axioms. We then evaluate the performance of the resulting knowledge base in solving the formalized Triangle-COPA questions with Etcetera Abduction using two methods of answer selection and two settings for back-chaining depth. We compare these results to those obtained using the original Triangle-COPA formulation (Maslan et al., 2015) and hand-authored axioms (Gordon, 2016).

Applying phase 1 of our methodology to the English portions of the Triangle-COPA questions yielded a new version of this evaluation that shared many of the qualities of the original. Each of the three question parts (*given*, *option A*, *option B*) were represented with largely the same number of literal forms, but different patterns of representational choices were evident. For example, consider the differences in literal forms used by the LLM and in Maslan et al.’s (2015) original formulation for Question 83, as follows.

83. *A small triangle and big triangle are next to each other. A circle runs by and pushes the small triangle. The big triangle chases the circle. Why does the big triangle chase the circle?*

Maslan: (and (approach' E1 C LT) (push' E2 C LT) (chase' E3 BT C) (seq E1 E2 E3))

LLM: (and (next_to' E1 LT BT) (run_by' E2 C LT) (push' E3 C LT) (chase' E4 BT C))

a. *The big triangle is angry that the circle pushed the small triangle, so it tries to catch the circle.*

Maslan: (angryAt' e4 BT C)

LLM: (and (angry_that' e1 BT E3) (try_to_catch' e2 BT C))

b. *The big triangle and circle are friends. The big triangle wants to say hello to the circle.*

Maslan: (and (friend' e5 BT C) (goal' e6 e7 BT) (greet' e7 BT C))

LLM: (and (are_friends_with' e1 BT C) (want_to_say_hello_to' e2 BT C))

In this example, some representational choices are identical, e.g., the use of `push'` and `chase'` predicates of arity 3. Here and in other questions, we see the LLM favor predicate names that closely match the English vocabulary, e.g., `run_by'` and `try_to_catch'`. While the LLM did not choose to explicitly order the *given* eventualities (using `seq`), it successfully used eventualities to encode a higher-order relation, i.e., the literal `(angry_that' e1 BT E3)` in option A. We expect that eventuality notation of this style is relatively infrequent in the corpora used for pretraining in modern LLM development, and therefore attribute the LLM's success to its adherence to the detailed instructions in our prompt.

Similar patterns were evident in the execution of phase 2 of our methodology, where 252 English glosses of the hand-authored knowledge base were automatically formalized. The following six examples compares the representational choices made by the LLM to Gordon's (2016) axioms.

1. *When someone is a parent of a child, the child often accompanies the parent.*

Gordon: (if (and (parent' e1 BT LT)
 (etc2_accompany 0.8 e1 e))
 (accompany' e LT BT))

LLM: (if (and (is_parent_of' e1 y x)
 (etc2_want_to_be_followed_by 0.7 e x y e1))
 (want_to_be_followed_by' e x y))

2. *On occasion, a person might annoy someone else.*

Gordon: (if (etc0_annoy 0.1 e x y)
 (annoy' e x y))

LLM: (if (etc0_bother 0.2 e x y)
 (bother' e x y))

3. *If you are angry at someone, you might chase them.*

Gordon: (if (and (angryAt' e1 x y)
 (etc2_chase 0.2 e1 e x y))
 (chase' e x y))

LLM: (if (and (angry_at' e1 x y)
 (etc2_chase 0.6 e x y e1))
 (chase' e x y))

4. *Once in a while, someone might hug another person.*

Gordon: (if (etc0_hug 0.01 e x y)
 (hug' e x y))

LLM: (if (etc0_hug 0.3 e x y)
 (hug' e x y))

5. *If you get pushed by someone, it is possible it is because they are a bully.*

Gordon: (if (and (bully' e1 x y)
 (etc3_push 0.1 e1 e x y))
 (push' e x y))
 LLM: (if (and (push' e1 y x)
 (etc1_bully 0.6 e y x e1))
 (bully' e y x))

6. One very likely reason to be surprised is if you see someone you didn't expect.

Gordon: (if (and (see' e1 x e2)
 (unexpect' e3 e2 x)
 (etc1_surprise 0.9 e1 e2 e3 e x))
 (surprise' e e2 x))
 LLM: (if (and (see' e1 x y)
 (not_expect_to_see' e2 x y)
 (etc1_surprised_by 0.9 e x y e1 e2))
 (surprised_by' e x y))

In examples 1 through 4, we see nearly identical logic between hand-authored and LLM-authored axioms, with two notable differences. First, the predicate names devised by the LLM are, in general, more verbose than the hand-authored predicate names, e.g., `is_parent_of'` compared to simply `parent'`. Second, the probabilities assigned to etcetera literals show considerable differences, owing perhaps to the LLM's strict adherence to the probability ranges suggested in the instructions. Example 5 shows a failure case of our methodology, where the consequent and antecedent are swapped in the axiom. This is likely due to the variation in phrasing that we added to the English glosses precisely to test the LLM's performance. In this case, being bullied is meant to be a causal antecedent to the pushing event, indicated by the phrase "it is possible that it is because". The order of antecedent and consequent is the opposite of the majority of other English glosses of axioms, and this subtlety was not noticed by the LLM. We expect that further engineering of the prompt to include a rich set of examples would suffice to overcome errors of this type. Example 6 demonstrates that the LLM can successfully compose complex antecedents of multiple literals, and is arguably more coherent in its use of predicates and variables than seen in the hand-authored version.

While our methodology succeeds in using an LLM to devise expressive literal forms for use in questions and axioms, it is also necessary to correctly reuse these invented forms across question and axiom formalizations. Without sufficient reuse of vocabulary, the formalized axioms will fail to unify with question literals, preventing the Etcetera Abduction algorithm from back-chaining any further than the prior-probability axioms.

To quantify the degree of reuse of literal forms throughout the two phases of our methodology, we tabulated cumulative total literals and cumulative unique literals after each formalization. Figure 1 (a) visualizes the reuse of LLM-devised predicate-argument forms while formalizing the 100 Triangle-COPA questions. The linear growth of cumulative total literal forms indicates consistent expressivity in the representation of Triangle-COPA questions over the full set of 100 questions. The linear growth of cumulative unique literal forms indicates that the LLM continued to invent new forms at a consistent rate across the phase, with approximately one new invention for each occurrence of reuse. A 10-question rolling average of reuse-rate indicates variability over the phase, but generally trends upward.

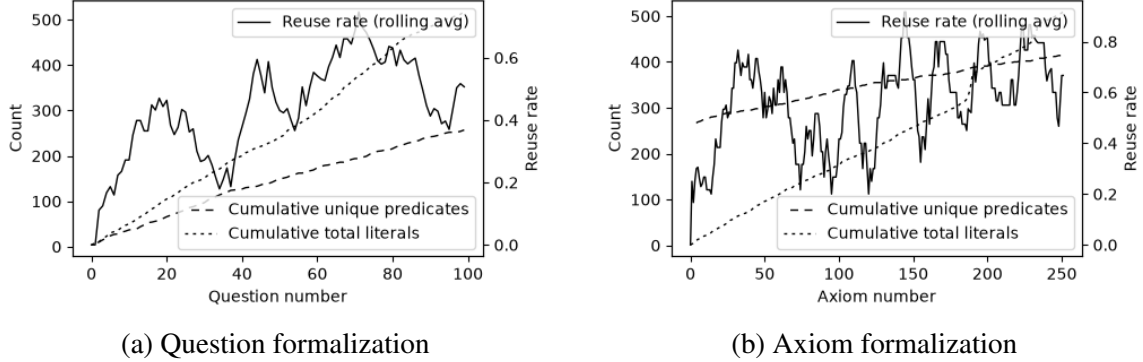


Figure 1. Reuse of literal forms during (a) formalization of 100 Triangle-COPA questions, and (b) formalization of 252 axiom glosses.

Figure 1 (b) visualizes the degree to which this amount of reuse continued when formalizing the 252 knowledge base sentences. The original knowledge base of Gordon (2016) was custom-built to solve these specific Triangle-COPA questions, such that the majority of predicate forms used in the axioms were drawn from the question formalizations. Likewise, we expect to see a higher rate of LLM reuse during this phase than in the previous phase. Indeed, the upward trend in the rolling average reuse rates continues in this phase, although newly-invented literal forms continue to be invented throughout.

To quantitatively assess the competency of the axiom formalization, we ran the Triangle-COPA test suite using the LLM-formalized question set and knowledge base axioms, and compared the results to the original formalizations with the same search parameter settings. Gordon (2016) had reported that 91 of the original 100 Triangle-COPA questions could be correctly answered using Etcetera Abduction and the hand-authored knowledge base using the *n*-best answer selection method, although some parameter tuning was required (depth and *n*-best size). For a fair comparison, we used fixed parameter settings across all of our evaluation runs, with $n = 100$ in our *n*-best answer evaluations, and back-chaining depths of both 3 and 4 across both answer selection methods. We furthermore added a per-question timeout of 60 seconds across all runs, effectively curtailing combinatorial search explosions.

Table 1 shows results when using the *n*-best answer selection method, with timeout set to 60 seconds per question, $n = 100$, and two settings for back-chaining depth (3 and 4). Results indicate poor performance of both the hand-authored and LLM-derived knowledge bases when using fixed depth, *n*-best size, and timeouts uniformly across all 100 questions. To achieve a score of 91 with the *n*-best answer selection method as reported in Gordon’s original paper (2016), the parameter settings for back-chaining depth, *n*-best size, and compute time would need to be specific to individual questions. Still, the performance of the LLM-derived axioms on LLM-formalized Triangle-COPA questions is markedly worse. The strict requirement for full unification between solutions and options in the *n*-best answer selection method leads to very high failure rates, where neither option can be unified with any of the top 100 most-probable solutions. Both the hand-authored and LLM-

Table 1. Results using *n-best answer* selection method across search depths.

	Correct	Ties	Wrong	Timeouts	Fails
Gordon 2016 (d=3)	61	2	1	1	35
Gordon 2016 (d=4)	46	1	1	30	22
Ours (d=3)	20	0	1	1	78
Ours (d=4)	17	0	1	23	59

Table 2. Results using *more-probable answer* selection method across search depths.

	Correct	Ties	Wrong	Timeouts	Fails
Gordon 2016 (d=3)	77	8	9	6	0
Gordon 2016 (d=4)	70	8	8	14	0
Ours (d=3)	54	17	28	1	0
Ours (d=4)	54	17	28	1	0

devised knowledge bases see high timeout numbers with back-chaining depth set to 4, indicating an unwieldy combinatorial search space. We find this somewhat encouraging, however, as it strongly indicates back-chaining beyond the prior-probability axioms when using the LLM-devised knowledge base.

Next we ran the Triangle-COPA evaluation suite using the *more-probable answer* selection method, where each option is conjoined with the *given* literals to form two interpretation problems, and where the more-probable of the two top solutions is selected as the answer. Table 2 shows the results using the *more-probable answer* selection method. These results indicate stronger performance for both the hand-authored and LLM-derived knowledge bases than seen using the *n-best answer* selection method, but again markedly better performance for the hand-authored axioms. Notably, splitting the question into two abduction problems appears to have increased timeouts for the hand-authored axioms at depth of 3, but the use of an $n=1$ *n*-best list reduced timeouts for both systems at a depth of 4. The high number of ties and wrong answers in the LLM-authored results suggest that probability assignments are too uniform in the knowledge base. Furthermore, the lack of any change when increasing the back-chaining depth indicates a failure to make use of longer multi-step chains of inferences in the most-probable solutions.

These quantitative results show that our two-phase formalization methodology does not produce knowledge bases that approach the performance of hand-authored axioms written by an expert targeting a specific benchmark. However, when browsing the various proof graphs produced for Triangle-COPA questions using the LLM formalizations, we saw several encouraging examples. Triangle-COPA question 31, for example, is formalized by the LLM as follows:

31. *Two triangles talk to each other and then hug. Why?*

```
(and (talk_to' E1 BT LT) (hug' E2 BT LT))
```

a. *The triangles are friends.*

```
(are_friends_with' e1 BT LT)
```

b. The triangles are enemies.

(are_enemies_with' e1 BT LT)

The LLM-authored knowledge base identifies option A as the more plausible alternative, finding it the highest-probability solution using the *n-best answer* selection method, with a simple proof graph depicted in Figure 2.

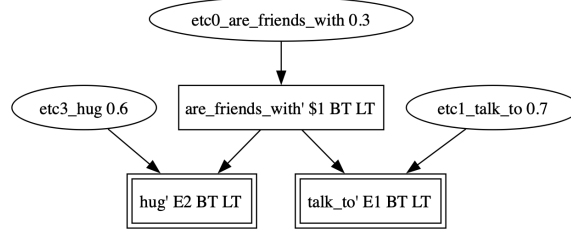


Figure 2. Proof graph of correct solution to Triangle-COPA question 31.

6. Discussion

The results presented in this paper indicate that our methodology for automatically formalizing English questions as abductive reasoning problems is not yet competitive with hand-authoring formalisms for these problems. As well, it must be remembered that neither approach is competitive with simply using the LLM to answer the questions using the English text alone. On a common-sense reasoning benchmark such as Triangle-COPA, or the earlier COPA evaluation that inspired it, contemporary LLMs of modest sizes can already achieve perfect scores. Instead, we view this line of research as a step toward solving other (non-commonsense) problems where transparency and correctness are of paramount importance. For probability-ordered abductive reasoning, these problems may include the interpretation of medical test results involving complex interactions and known false-positive rates, or when calculating relative certainties in high-stakes military operations that impact a course of action. In applications such as these, we see opportunities for LLMs to outsource the combinatorial search and probability estimations in much the same way that an LLM agent might make a tool call to a calculator or a satisfiability solver. Towards this end, the Triangle-COPA benchmark serves as a useful tool in prototyping and evaluating alternative approaches.

Our current investigation highlighted some of the challenges that remain. First, there is a need for better methods of estimating probabilities assigned to defeasible axioms, or alternatively some method for deriving these probabilities from data or simulation. As a first step, some basic probability theory may help constrain the possible range of LLM-estimated values, e.g., ensuring that the prior probability is always the most probable solution when interpreting a single literal. Second, new approaches are needed to ensure that LLM formalisms of axioms are supportive of longer, multi-step inference chains and produce well-connected proof graphs. Better guidance in our LLM prompts concerning the reuse of existing vocabulary is a reasonable first step, but we also see opportunities for multi-turn iterative refinement of the knowledge base. For example, it may be possible

to iteratively rewrite knowledge base axioms until they achieve the desired proofs for observables that are known to share common factors. Third, we see value in working toward the incorporation of existing, well-validated formal theories alongside LLM-generated axioms and problem formulations. The fluid utilization of existing theories to solve novel problems would be applicable across a broad range of scientific, engineering, and everyday human concerns.

7. Conclusions

In this paper we investigated a method for translating English questions and knowledge into logical literals and axioms to be answered with an automated abductive reasoning engine (Etcetera Abduction). Our approach was to prompt an LLM with very explicit instructions regarding the desired logical forms, and to enforce adherence to these instructions using schema-constrained generation (structured output). The predicate-argument literal forms invented by the LLM were comparable to the vocabulary seen in hand-authored formalizations, and moderate vocabulary reuse was observed over the course of question and knowledge base formalization. We quantitatively evaluated the performance of our method using the Triangle-COPA benchmark for formal commonsense reasoning, starting from English glosses of 252 axioms previously authored as a solution to this benchmark. Our results showed that fully-automated formalization of questions and axioms is not yet competitive with a hand-authored approach. However, we observed encouraging multi-step inference when reasoning about Triangle-COPA problems using LLM-authored knowledge bases, pointing to several directions for future improvements. The contribution of this work is in its exploration of a fully-automated approach to logical abductive reasoning from English representations of problems and knowledge, and in demonstrating how solved benchmarks (like Triangle-COPA) can continue to inform the development of contemporary cognitive systems.

Acknowledgements

Research was sponsored by the Army Research Office and was accomplished under Cooperative Agreement Number W911NF-25-2-0040. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

- Berant, J., & Liang, P. (2014). Semantic parsing via paraphrasing. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1415–1425). Baltimore, Maryland: Association for Computational Linguistics.
- Brown, T., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems* (pp. 1877–1901).

- Chemburkar, A., Gordon, A., & Feng, A. (2024). Evaluating vision-language models on the trianglecopa benchmark. *The International FLAIRS Conference Proceedings*, 37.
- Drozdo, A., Zhou, J., Florian, R., McCallum, A., Naseem, T., Kim, Y., & Astudillo, R. (2022). Inducing and using alignments for transition-based AMR parsing. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1086–1098). Seattle, United States: Association for Computational Linguistics.
- Fellbaum, C. (2005). Wordnet and wordnets. *Encyclopedia of Language and Linguistics* (pp. 665–670). Oxford: Elsevier.
- Gordon, A. (2025). Automated formalization of william shakespeare’s venus and adonis using large language models. *Twelfth Annual Conference on Advances in Cognitive Systems*.
- Gordon, A. S. (2016). Commonsense interpretation of triangle behavior. *Thirtieth AAAI Conference on Artificial Intelligence* (pp. 3719–3725). Palo Alto, CA: AAAI Press.
- Gordon, A. S., & Roemmele, M. (2014). An authoring tool for movies in the style of heider and simmel. *Interactive Storytelling* (pp. 49–60). Cham: Springer International Publishing.
- Heider, F. (1958). *The psychology of interpersonal relations*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Heider, F., & Simmel, M. (1944). An experimental study of apparent behavior. *The American Journal of Psychology*, 57, 243–259.
- Hobbs, J. R. (1985). Ontological promiscuity. *Proceedings of the 23rd Annual Meeting on Association for Computational Linguistics* (pp. 60–69). Association for Computational Linguistics.
- Hobbs, J. R., Stickel, M. E., Appelt, D. E., & Martin, P. (1993). Interpretation as abduction. *Artificial Intelligence*, 63, 69–142.
- Magerman, D. M. (1995). Statistical decision-tree models for parsing. *33rd Annual Meeting of the Association for Computational Linguistics* (pp. 276–283). Cambridge, Massachusetts, USA: Association for Computational Linguistics.
- Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19, 313–330.
- Maslan, N., Roemmele, M., & Gordon, A. S. (2015). One hundred challenge problems for logical formalizations of commonsense psychology. *Proceedings of the Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning* (pp. 107–113). Palo Alto, CA: AAAI Press.
- Ovchinnikova, E. (2012). *Integration of world knowledge for natural language understanding*. Atlantis Thinking Machines. Atlantis Press.
- Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Roemmele, M., Bejan, C., & Gordon, A. (2011). Choice of plausible alternatives: An evaluation of commonsense causal reasoning. *Proceedings of the AAAI Spring Symposium on Logical Formal-*

- izations of Commonsense Reasoning*, Stanford University.
- Schank, R. C. (1975). The primitive ACTs of conceptual dependency. *Theoretical Issues in Natural Language Processing*.
- Shin, R., et al. (2021). Constrained language models yield few-shot semantic parsers. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 7699–7715). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.
- Smedslund, J. (1997). *The structure of psychological common sense*. Lawrence Erlbaum Associates Publishers.
- Yang, J., Jiang, H., Yin, Q., Zhang, D., Yin, B., & Yang, D. (2022). SEQZERO: Few-shot compositional semantic parsing with sequential prompts and zero-shot models. *Proceedings of NAACL’22 Findings*. Association for Computational Linguistics.