NELLIE: A Neuro-Symbolic Inference Engine for Grounded, Compositional, and Explainable Reasoning

Nathaniel Weir[1], **Peter Clark**[2], and **Benjamin Van Durme**[1]
[1]Johns Hopkins University, Baltimore, MD, USA
[2]Allen Institute for AI, Seattle, WA, USA
{nweir, vandurme}@jhu.edu, peterc@allenai.org

Abstract

Our goal is to develop a modern approach to answering questions via systematic reasoning where answers are supported by human interpretable proof trees grounded in an NL corpus of facts. Such a system would help alleviate the challenges of interpretability and hallucination with modern LMs, and the lack of grounding of current explanation methods (e.g., Chain-of-Thought). This paper proposes a new take on Prolog-based inference engines, where we replace handcrafted rules with a combination of neural language modeling, guided generation, and semiparametric dense retrieval. Our implementation, NELLIE, is the first system to demonstrate fully interpretable, end-to-end grounded QA as entailment tree proof search, going beyond earlier work explaining known-to-be-true facts from text. In experiments, NELLIE outperforms a similar-sized state-of-the-art reasoner while producing knowledge-grounded explanations. We also find NELLIE can exploit both semi-structured and NL text corpora to guide reasoning. Together these suggest a new way to jointly reap the benefits of both modern neural methods and traditional symbolic reasoning.

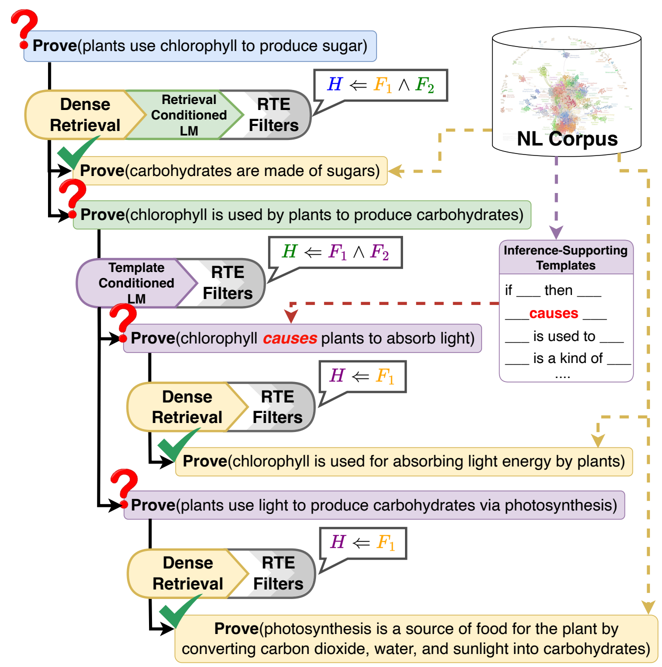

Figure 1: Given a query, NELLIE performs a neuro-symbolic backward chaining search for proof trees whose leaves are grounded in a corpus of facts. It generates candidate decomposition rules guided by retrieved facts or templates. Then, it recursively tries to prove rule conditions via entailment from the corpus or further decomposition.

1 Introduction

Large language models (LLMs) are impressive at question-answering (QA), but it remains challenging to understand how answers systematically follow from authoritative information. This general opacity is a growing impediment to widespread use of LLMs, e.g., in critical applications such as medicine, law, and hiring decisions, where interpretability and trust are paramount. While there has been rapid progress in having LLMs generate systematic explanations for their answers, e.g., Chain-of-Thought [Wei *et al.*, 2022], Entailer [Tafjord *et al.*, 2022], or Maieutic Prompting [Jung *et al.*, 2022], such explanations are not grounded in external facts and may include hallucinations [Ji *et al.*, 2022]. Rather, what would be desirable - and what this work pursues - is a system that systematically reasons over authoritative text: to support answers with human interpretable proof trees grounded in the text, while not requiring translation to an entirely symbolic formalism.

Our approach is to revisit the behavior of *expert systems* [Jackson, 1986; Metaxiotis *et al.*, 2002]. Expert systems are appealing for their explainable behavior: decisions are made by constructing a well-formed symbolic proof from explicit, formally represented knowledge authored by a knowledge engineer in consultation with a domain expert. However, as expert systems are known to be both expensive and brittle [Musen and Van der Lei, 1988], the AI community has turned to *neuro-symbolic* mechanisms that use large language models to reason over natural language (NL). Reasoning over NL does not require a symbolic formalism and allows the use of inferentially powerful LLMs, but also loses the systematic, verifiable proofs that expert systems produced. This motivates our pursuit of a new way to jointly reap the benefits of both modern neural methods and traditional reasoning.

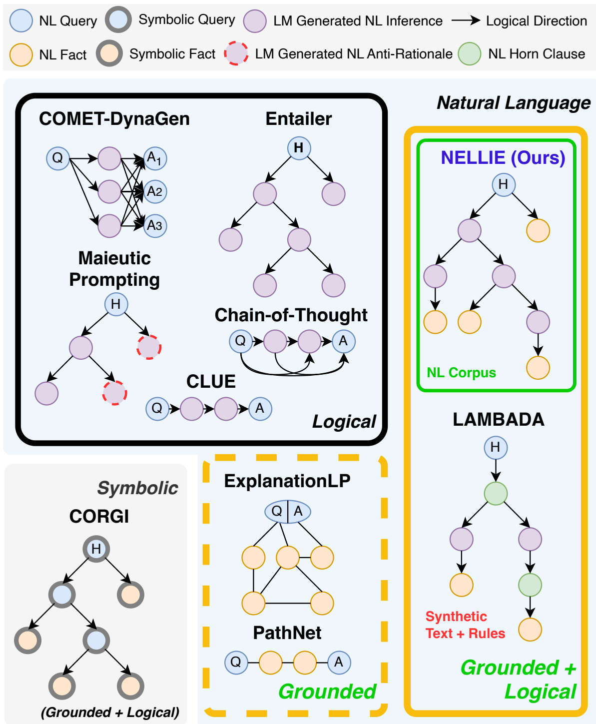

Figure 2: Comparison of approaches to neural XQA. Each approach leads to NL graphs in support of a **query**. Our proposal is to produce logically directed explanations containing **model-generated intermediate steps** while grounding a tree in **verified facts** without relying on **handwritten horn clauses**.

We desire the following expert system-inspired desiderata:

1. **Grounding** inferences fully and scalably in a corpus of knowledge from an authoritative human source.
2. **Logical direction**, showing how a given hypothesis (and all intermediate inferences leading up to it) follows as the logical consequence of the underlying knowledge source
3. **Competitive end-to-end QA performance** in a complex domain requiring diverse forms of reasoning.

As illustrated in Figure 2, various eXplainable QA (XQA) methods accomplish 2 of these criteria, but not all 3. Some methods, like Chain-of-Thought, generate inference chains or logically structured graphs from an LLM without grounding in verified knowledge. Others, like ExplanationLP [Thayaparan *et al.*, 2021], compose graphs of grounded facts, but do not show logical direction. LAMBADA [Kazemi *et al.*, 2023] achieves both direction and grounding but is limited to simple domains with provided NL Horn rules and sets of 1-2 dozen facts. In contrast, we desire a system that handles a larger corpus (10K statements) and doesn't need handcrafted rules.

To achieve all three desiderata, we reuse the general inference framework of an expert system, but replace handcrafted rules with a combination of neural language modeling, guided generation, and semiparametric dense retrieval. We demonstrate this in a system called NELLIE, the **Ne**uro-Symbolic **L**arge **L**M **I**nference **E**ngine. NELLIE leverages LLMs as *proposal functions* in the search of proof trees showing how a conclusion follows via entailment from an external corpus of NL statements. The "symbols" that our neuro-symbolic engine reasons over are free-form NL sentences. Rather than require a knowledge engineer to carefully write hundreds of inference rules as in the classical setting, NELLIE employs an LLM as a *dynamic rule generator* [DRG; Kalyanpur *et al.*, 2021] to *generate* (rather than retrieve) candidate rules that decompose a hypothesis into subqueries that, if themselves proved recursively, will prove that hypothesis via composition (Figure 1). In this way, NELLIE can answer the question posed by the classical expert system: "Does this statement systematically follow from my knowledge, or not?".

NELLIE is built upon a backward chaining symbolic theorem prover written in Prolog, implemented using a few *meta-rules* specifying how inference should proceed. These include checking for entailment of a hypothesis against a retrieved fact or decomposing it into a conjunction of subqueries to recursively prove. To treat NL sentences as if they were symbols in a purely symbolic search algorithm, we use a natural language inference-based form of *weak unification* [Sessa, 2002; Weber *et al.*, 2019] between the hypothesis and a corpus fact. This produces interpretable proofs similar to the compositional entailment trees of prior work (e.g., EntailmentBank [Dalvi *et al.*, 2021]) while tackling the additional challenge of QA.

NELLIE is designed to search across hundreds of trees whose leaves come from one of two types of knowledge: (a) a corpus of semi-structured text statements, such as NL renditions of database entries, or (b) a corpus of free-form NL sentences. Many *correct* proofs might exist for a given hypothesis, but much fewer will be *fully groundable* in the provided corpus, making the search harder than for ungrounded alternatives like Entailer [Tafjord *et al.*, 2022] or COMET-Dynagen [Bosselut *et al.*, 2021]. To improve grounding, we introduce two guiding methods to boost the likelihood of generating rules whose conditions match against the available text: (1) For applications where a semi-structured corpus is available, NELLIE leverages this structure via templates to help steer rule generation towards syntax that is more likely to match corpus entries. (2) For both free-form and semi-structured corpora, NELLIE retrieves and conditions on statements to help steer generation towards trees grounded in them. Ablation experiments show these together and individually improve NELLIE's reasoning.

NELLIE expands upon methods that ground known-to-be-true hypotheses to provided facts [SCSEARCH; Bostrom *et al.*, 2022], extending the paradigm to perform QA. This involves searching for and comparing trees for conflicting answer options. Experiments find NELLIE outperforms a similar-sized state-of-the-art reasoner, Entailer, while producing compositional trees showing how decisions are grounded in corpus facts. Our contributions are thus:

1. An architecture for systematic reasoning from a corpus of textual knowledge to answer questions. This can be viewed as a modern approach to expert system inference, but without requiring a formal knowledge representation.
2. An implementation, NELLIE,[1] that outperforms a similar-sized SOTA reasoner (Entailer-3B) while producing

[1]Code and appendix at https://github.com/JHU-CLSP/NELLIE.

grounded trees. To our knowledge, this is the first system to perform grounded XQA as NL entailment tree search.

2 Related Work

Theorem Proving over Language A long-standing approach to reasoning over NL is to project it into a symbolic form, such as for QA [Green *et al.*, 1961; Zelle and Mooney, 1996] or entailment [Bos and Markert, 2005]. Provided the translation from NL to symbolic representation is accurate, one can leverage fast and scalable solutions for discrete symbolic inference [Riazanov and Voronkov, 2002; Kautz *et al.*, 1992; Kautz and Selman, 1999]. However, reliably translating broad-domain NL into an adequately expressive formalism for reasoning is a challenge [Schubert, 2015], though some have found success using LLMs to perform this semantic parsing task [Wong *et al.*, 2023; Lyu *et al.*, 2023; Olausson *et al.*, 2023; Pan *et al.*, 2023; Ye *et al.*, 2023].

Recent work explores methods of handling NL without parsing it to another formalism. Some use LMs to generate proof steps in mathematical theorem proving [Polu and Sutskever, 2020; Welleck *et al.*, 2022]. Variants of neural theorem provers [NTPs; Rocktäschel and Riedel, 2017] such as NLProlog [Weber *et al.*, 2019] reconcile NL facts with symbolic reasoning by learning embeddings for the facts and symbols in a theory, then using weak unification to backward chain. Kalyanpur *et al.*, 2021 inject neural reasoning into a Boxer/Prolog-based symbolic reasoner via a special LM-calling predicate. Arabshahi *et al.*, 2021b combine handwritten symbolic rules with neural symbol embeddings to classify conversational intents. Other work explores using LMs to emulate stepwise [Tafjord *et al.*, 2021; Kazemi *et al.*, 2023] or end-to-end [Clark *et al.*, 2021; Picco *et al.*, 2021] logical reasoning over small rulebases converted to NL. These approaches require both user-provided if/then rules to operate, while NELLIE requires only facts and is thus applicable to domains in which rules are not available.

Modular Reasoning over NL NELLIE's systematic reasoning is related to approaches that decompose problems into sequences of modular operations. Gupta *et al.*, 2020 introduce a neural module network [NMN; Andreas *et al.*, 2016] for QA with modules for span extraction and arithmetic operations. Khot *et al.*, 2021 introduce another NMN that decomposes questions into simpler ones answerable by existing models. These are part of a broader class of work decomposing multi-step reasoning using reasoning modules [Khot *et al.*, 2022; Saha *et al.*, 2023a; Saha *et al.*, 2023b].

Systematic Explanation Generation Recent works have used LMs to generate NL reasoning graphs in support of an answer. "Chain-of-Thought" prompting [Wei *et al.*, 2022; Kojima *et al.*, 2022], elicits free-form inference hops from the LM before it generates an answer. Other text graph generation methods connect model-generated statements via common sense relations [Bosselut *et al.*, 2021; Arabshahi *et al.*, 2021a] or for/against influence [Madaan *et al.*, 2021; Jung *et al.*, 2022], though these are not knowledge-grounded and don't address entailment (see Figure 2).

The EntailmentBank dataset [Dalvi *et al.*, 2021] has driven research towards the construction of *explanation trees*, challenging models to produce stepwise entailment proofs of a statement using a set of provided support facts. This direction builds upon works on explainable reasoning that build graphs from KB-retrieved sets of support statements, but stop short of showing their role in logical entailment [Pan *et al.*, 2019; Jansen and Ustalov, 2019; Yadav *et al.*, 2019; Valentino *et al.*, 2022]. Components of our framework are related to concurrent approaches for entailment tree construction [Bostrom *et al.*, 2022; Hong *et al.*, 2022; Sprague *et al.*, 2022]. Ribeiro *et al.*, 2022 also use iterative retrieval-conditioned generation, and Yang *et al.*, 2022 also use entailment classifiers to filter proof steps. None of the above tree generation approaches consider this harder scenario of multiple-choice QA, opting instead to focus on the reconstruction task from support facts. Tafjord *et al.*, 2022 do consider the harder task. They propose a backward chaining QA system, Entailer, that generates entailment trees (not grounded in human-verified facts) using a search grounded to the model's internal beliefs, complementary to NELLIE's use of guided and retrieval-conditioned generation to obtain knowledge grounding. As with NELLIE (§5.2), Entailer benefits from humans adding more useful statements to its available knowledge [Dalvi Mishra *et al.*, 2022]. The difference between systems is highlighted by the number of generations (i.e. search nodes) considered by the two algorithms: while Entailer considers at most a couple dozen hypothesis decompositions, NELLIE must consider hundreds or thousands to find one that is fully grounded.

Faithful Complex LLM Reasoning Of the growing body of work on LLM-based "Chain-of-X" reasoning methods [Xia *et al.*, 2024], a portion considers ways to improve faithfulness to underlying knowledge and reduce LLM hallucinations (see Lyu *et al.*, 2024 for an overview of faithful explanation methods). NELLIE contributes to this space by providing a faithful-by-design algorithm that reasons based on logical hypothesis grounding. Other recent methods include Rethinking with Retrieval [He *et al.*, 2022], which reranks reasoning chains using a faithfulness score based on retrieved knowledge, and the forward-chaining Faithful Reasoning algorithm [Creswell and Shanahan, 2022], which follows "a beam search over reasoning traces" to answer a question by iteratively adding deductive inferences to a context of starting facts. Theirs is a different style search to our backward-chaining NELLIE, not relying upon NLI for verifying logical connectedness and not scaling to larger knowledge bases.

3 Background

A logical expert system proves a propositional query against a *theory* comprised of facts and inference rules, generally given in the form of Horn clauses. Upon finding a rule whose head can *unify* with the query, a depth-first backward chaining algorithm such as those used in Prolog solvers will perform variable substitution and then recursively attempt to prove the terms in the rule's *body*. For example, a disease classification system might prove query $\textsc{Contagious}(\textit{flu})$ via facts $\textsc{Contagious}(\textit{influenza})$ and $\textsc{OtherName}(\textit{flu}, \textit{influenza})$, and conjunctive rule $\textsc{Contagious}(X) \Leftarrow \textsc{OtherName}(X, Y) \wedge$

CONTAGIOUS(Y). It does so by unifying CONTAGIOUS(*flu*) with CONTAGIOUS(X) and then recursively unifying the terms in the rule body with their matching facts. Here, *flu* is an object symbol, CONTAGIOUS is a predicate symbol, and X is a variable. See Russell and Norvig, 2010 for further details.

Neural Predicates While most declarative predicates have meaning only in the context of user-defined inference axioms, others can call external neural modules that produce values for their arguments or determine the truth value of the predicate. A popular implementation of this paradigm is DeepProbLog [Manhaeve *et al.*, 2018], which we use to define LM-invoking predicates. In the above example, we might train a sequence-to-sequence (seq2seq) model to produce other names for a disease Y, turning OTHERNAME(Y^+, X^-)[2] into a neural predicate. This mechanism creates the ability to introduce externally defined object symbols, e.g. seq2seq-generated NL, into the engine's vocabulary.

Weak Unification In classical backward chaining, a *unification* operator assigns equivalence to two logical atoms; this requires atoms to have the same arity and have no unequal ground literals in the same argument position. Issues arise when literals are NL sentences, which can be syntactically distinct but semantically equivalent. To handle this, Weber *et al.*, 2019 propose a *weak unification* operator, which allows for the unification of *any* same-arity atoms regardless of conflicting symbols.[3] They estimate a *unification score* as the aggregation of pairwise similarity scores using a similarity function $\theta(s_1, s_2) \in [0, 1]$. The score of the full proof is taken as the minimum of scores across all steps. In this work, we apply a similar aggregation; we say that a query fact s_1 "weakly unifies" with provenance fact s_2 with a unification score equal to the confidence of an NLI model taking s_2 as the premise and s_1 as the hypothesis.

4 Overview of Approach

Depicted in Figure 3, our framework is comprised of: an external corpus of facts (some $\{f_1, \dots f_n\}$); a module that converts a QA pair to a hypothesis; an off-the-shelf theorem prover; and a suite of meta-axioms that use neural fact retrieval and dynamic rule generation modules to propose, verify, and score inferences. In our experiments, we consider one implementation of this framework that uses the corpus WorldTree [Xie *et al.*, 2020], a set of 9K NL science facts that can interchangeably be considered as rows in 81 n-ary relational tables.

Question Conversion Given a multiple-choice question, the system converts each candidate answer into a hypothesis h using a Question to Declarative Sentence model [QA2D; Demszky *et al.*, 2018] (See §C). It then searches for a proof of h against its knowledge base. For each alternative, we enumerate p proofs using a time-capped backward chaining search and then take as the system's answer the candidate with the overall highest-scoring proof.

[2] In Prolog syntax, '+' denotes inputs, '−' outputs.

[3] Introducing weak unification greatly increases the search runtime, as one might try to unify any two symbols in the vocabulary at every recursive step. This poses a substantial challenge when applying the query grounding algorithm to a search space over NL.

4.1 Inference Rule Structure

Our approach uses LMs to dynamically generate inference rules given a hypothesis. The rule structure is strikingly simple, instantiating one of the following meta-level templates:

I. `Hypothesis` $\Leftarrow$ `Fact`

II. `Hypothesis` $\Leftarrow$ `Fact1` $\wedge$ `Fact2`

Via template I, the system proves the hypothesis by finding a provenance `Fact` stored in its knowledge store that entails the hypothesis. Via template II, it enumerates a pair, `Fact1` and `Fact2`, both either stored in the knowledge store or themselves recursively proved, such that the pair in conjunction entails the hypothesis.[4] Template I is given higher search precedence than II, yielding an intuitive high-level procedure: we first look up the hypothesis against the factbase, searching for an entailing fact. If we do not find one, we decomposes the hypothesis into a pair of statements to be proved. Concretely, for an input hypothesis h, we define the predicate PROVE(h) that serves as the primary goal term. We define the following core meta-rules, which use the neural predicates RETRIEVE, ENTAILS, and RULEGEN. At each step in the backward-chaining search, NELLIE's Prolog engine attempts to unify a query with the head of one of these three rules:

1. *Fact Unification*
   $\text{PROVE}(h) \Leftarrow \text{RETRIEVE}(h^+, f^-) \wedge \text{ENTAILS}(f, h)$
2. *Two Premise Rule Generation*
   $\text{PROVE}(h) \Leftarrow \text{RULEGEN}(h^+, f_1^-, f_2^-) \wedge \text{ENTAILS}([f_1, f_2], h) \wedge \text{PROVE}(f_1) \wedge \text{PROVE}(f_2)$
3. *Retrieved First Premise Rule Generation*
   $\text{PROVE}(h) \Leftarrow \text{RETRIEVE}(h^+, f_1^-) \wedge \text{RULEGEN}(h^+, f_1^+, f_2^-) \wedge \text{ENTAILS}([f_1, f_2], h) \wedge \text{PROVE}(f_2)$

4.2 Unification with Retrieved Facts

For factbase fact f, PROVE(f) is vacuously true. Rule 1 shows how we prove PROVE(h) using retrieval. The predicate RETRIEVE proposes candidate f_i's given h using a FAISS [Johnson *et al.*, 2019]-based nearest neighbor dense retrieval index. We train the retrieval encoder via ranking loss such that the embedding for a hypothesis is maximally similar to its supporting facts. To promote logical coherence and improve the precision of the system, we apply a set of **neural models for recognizing textual entailment (RTE)** as filters ENTAILS$_j(\cdot)$ that iteratively rule out f_i candidates that are not classified as entailing h.

$$\text{ENTAILS}(f_i, h) \Leftarrow \bigwedge_{j=1\dots n} \text{ENTAILS}_j(f_i, h)$$

Implicit in rule 1 is that PROVE(h) weakly unifies with some PROVE(f_i); we assign the unification score $\theta(h, f_i)$ equal to the confidence of one RTE model.

For some questions such as the one depicted in Figure 4, it is necessary to ground a subquery in evidence from the problem. To handle this, we add the question setup (defined as all but its last sentence) as a "fact" always proposed by RETRIEVE.

[4] We find that two-premise decomposition is sufficiently powerful and expressive for our purposes. For example, to prove a hypothesis such as those in EntailmentBank [Dalvi *et al.*, 2021] that requires a *three*-premise conjunction $h \Leftarrow f_1 \wedge f_2 \wedge f_3$, NELLIE produces instead a recursive set of decompositions $H \Leftarrow f_1 \wedge f_i; f_i \Leftarrow f_2 \wedge f_3$.

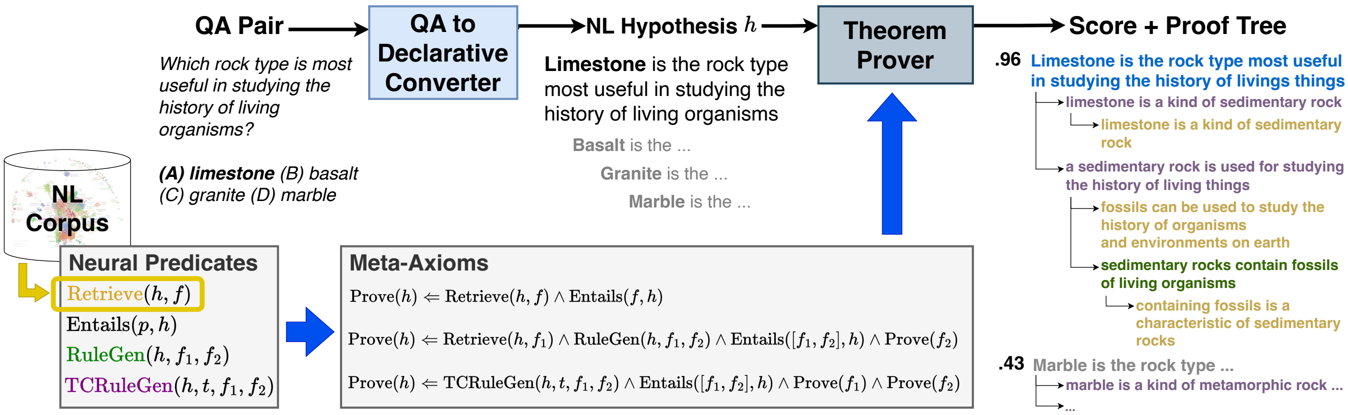

Figure 3: Proposed system framework. An off-the-shelf theorem prover searches for proofs of query PROVE(h), where symbol h is an NL hypothesis translated from a QA pair. The prover uses a set of meta-axioms invoking neural retrieval, entailment, and generation predicates to dynamically instantiate inference rules that use the NL factbase.

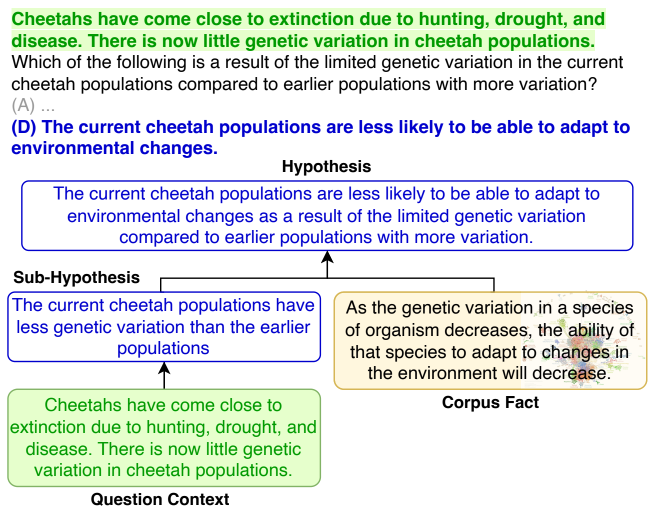

Figure 4: Example question in which a sub-hypothesis in the proof tree is grounded to the question context rather than to the factbase.

4.3 Dynamic Rule Generator (DRG)

If we do not find an entailing fact for hypothesis h, we decompose h into a pair of entailing premises; we propose candidates using nucleus sampling [Holtzman *et al.*, 2019] from a seq2seq model. The predicate RULEGEN(h, f_1, f_2) prompts a model trained to generate $h \rightarrow f_1, f_2$ pairs. The space of potential decompositions is very large; there are many deductively valid ways to prove a hypothesis in natural language, though only a fraction of them are ultimately groundable in the provided factbase. To bias the proof search towards those more likely to be grounded in the corpus, we adopt a two-pronged approach, illustrated in inference rules 2 and 3, that proposes a *heterogeneous search frontier* using different biasing strategies in addition to straightforward LM sampling.

Template Conditioned Generation (TCG) One way in which we improve our LM-based proposal function is to leverage the high-level structure that supports reasoning in a given domain: we propose *template-guided generation* to bias search towards the semi-structure of WorldTree.[5] WorldTree tables correspond to types of facts with similar syntax and semantics that support scientific reasoning. WorldTree questions are annotated with the fact rows that support their answers. Each table can be viewed as an *n*-ary relation of columns whose values are text spans. E.g., the Taxonomic relation has columns *<A>, [HYPONYM], <is a / a kind of>, <SCOPE1>, [HYPERNYM], <for>, [PURPOSE]*. Rows include *'a bird is an animal'* and *'a seed is a kind of food for birds.'*

Thus, for the RULEGEN predicate in rule 2, half of the f_1, f_2 candidates are sampled from the DRG conditioned on h plus a template that cues the model to reflect the syntax of a WorldTree table. We train the DRG to accept any masked infilling template (e.g. *<mask>is a kind of <mask>*, akin to those used to pretrain LMs [Lewis *et al.*, 2019; Raffel *et al.*, 2020]), and propose decompositions whose first fact reflects the template's syntax. We thus create a $h, t_1 \rightarrow f_1, f_2$ model. We feed the model templates drawn from WorldTree's tables, guiding it towards proof steps more likely to be grounded in the factbase. A sample of the 150 templates can be found in Figure 5 (a larger list is shown in §D). We reuse the same model for non-template-conditioned generation by feeding it an empty template. In practice, we make two generation calls: one samples m free-generated candidates, and a second samples n candidates for each of n_t templates.

$$\text{RULEGEN}(h^+, f_1^-, f_2^-) \Leftarrow \text{MEMBER}(t, \text{TEMPLATES} \cup \{\text{BLANK}\}) \wedge \text{TCRULEGEN}(h^+, t^+, f_1^-, f_2^-)$$

Template Selection WorldTree is a diverse corpus, containing tables that are specific to a particular subset of science

[5] While our approach is applicable to a wide array of reasoning problems, the use of WorldTree-specific templates illustrates one way by which we can infuse domain-specific structure into the neural search algorithm. This is a strict departure from the burdensome process of symbolic knowledge engineering. As our experiments show, this guidance method improves NELLIE's QA accuracy by a few points, though ablating it yields a similarly strong performance.

WorldTree Relation	Template
KINDOF	<mask>is a kind of <mask>
IFTHEN	if <mask>then <mask>
PROP-THINGS	<mask>has <mask>
CAUSE	<mask>causes <mask>
MADEOF	<mask>made of <mask>
REQUIRES	<mask>requires <mask>
ACTION	<mask>is when <mask>
USEDFOR	<mask>used to <mask>
PARTOF	<mask>a part of <mask>
CHANGE	<mask>changes <mask>
USEDFOR	<mask>used for <mask>
AFFECT	<mask>has <mask>impact on <mask>
PROP-ANIMAL-ATTRIB	<mask>is a <mask>animal
SOURCEOF	<mask>is a source of <mask>
COMPARISON	<mask>than <mask>
EXAMPLES	an example of <mask>is <mask>
COUPLEDRELATIONSHIP	as <mask>decreases <mask>will <mask>
PROP-CONDUCTIVITY	<mask>is <mask>conductor
PROP-GENERIC	<mask>is a property of <mask>
HABITAT	<mask>live in <mask>
MEASUREMENTS	<mask>is a measure of <mask>

Figure 5: Sample of WorldTree templates used for guided generation.

problems (e.g. the "Predator-Prey" table). As conditioning on dozens of templates can be computationally expensive, we introduce a *case-based reasoning* [Schank, 1983; Das *et al.*, 2021] approach that selects relevant templates for a given hypothesis. We construct an Okapi-BM25 [Jones *et al.*, 2000] retrieval index over questions from the WorldTree QA training set to obtain the most lexically similar items to a query. At inference time, we select the top-k most similar questions to the query and take as our template subset the tables of the questions' annotated facts.

Retrieval Conditioned Generation (RCG) In rule 3, rather than generate a pair of subqueries, we *immediately ground* half of the antecedent by choosing as f_1 a fact retrieved directly from the corpus. We have the DRG force decode f_1 before generating f_2 as normal, then recur only on f_2.

Filters As stochastic sampling from LMs can be noisy, some fraction of the generated candidate set may be invalid: premises may be incoherent, or the decomposition might not properly entail h. Accordingly, we introduce a set of compositional entailment verifiers [Khot *et al.*, 2020; Jhamtani and Clark, 2020] trained on two-premise compositional entailment. We also add a "self-ask" filter, which following Tafjord *et al.*, 2022 is an LM fine-tuned to assign a statement a truth value. If the confidence is below 0.5 for either an entailment judgment or the 'self-ask' belief in f_1 or f_2, then we filter the pair. All filters condition on the question text as context. When NELLIE uses these rules, the unification score equals the lowest of the scores for $\textsc{Prove}(f_1)$, for $\textsc{Prove}(f_2)$, and the confidence of entailment filters $s_e([f_1, f_2] \Rightarrow H)$.

4.4 Proof Search

Given a query, NELLIE searches for t seconds to find up to p proofs of depth d or less. We follow Weber *et al.*, 2019 in pruning search branches whose unification score is guaranteed to fall below the current best, given our monotonic aggregation function $\min(\cdot)$. Found proofs that score under the current best do not count towards p. Full pseudocode for the algorithm, which follows a depth-first search with a breadth-first lookahead [Stern *et al.*, 2010] to check for the unification of generated subgoals, can be found in §E. It is parameterized by

1. A maximum number of proofs m at which to cut off searching. In experiments, we set this to 10 for top-level queries and 2 for recursive subqueries.
2. A number of support facts n_f to retrieve at each call to $\textsc{Retrieve}_K$, which we set to 15.
3. Candidate generation rates n_v for vanilla nucleus-sampled decompositions, n_t for template-conditioned decompositions, and n_r for retrieval-conditioned generations. We set these each to 40.[6] Upon removing exact match duplicates, a call to RULEGEN produces about 100 candidates.
4. Entailment scoring module $\textsc{Score}_e(\cdot)$, which is a separate RTE cross-encoder model for single- and double-premise entailments.

5 Experiments

We train the components of NELLIE to be able to answer questions in the Science QA domain. The different neural modules are trained on reformulations of existing datasets for scientific reasoning. Further information can be found in §A.

Our experiments illustrate how the approach exemplified by NELLIE provides grounded and logical explanations, performing comparably or better than approaches that do not satisfy these properties. We evaluate models on two multiple-choice QA datasets constructed so that correct answers are supported by facts in the WorldTree corpus:

EntailmentBank [Dalvi *et al.*, 2021] is a dataset of entailment trees for declarativized answers to the AI2 Reasoning Challenge (ARC) dataset [Clark *et al.*, 2018], showing how the hypothesis can be derived via compositional entailment hops from WorldTree facts. We recast the test set, initially designed to test tree reconstruction, into QA by retrieving the corresponding multiple-choice ARC questions from which the hypotheses were constructed.

WorldTree [Xie *et al.*, 2020] is a subset of the ARC dataset annotated with undirected explanation graphs whose nodes are facts from the WorldTree tablestore. We note that WorldTree explanations do not show how facts should combine. *There is no guarantee that fully grounded trees exist for these questions using the WorldTree corpus alone*. The difficulty of this task is akin to that of a course exam: the teacher (us) provides the student (the model) with a very large study guide of facts, but expects the student to **use** these facts by composing them to reason coherently about a problem.

Our task metric is accuracy: whether a generated proof of the correct option outscores any other.[7]

Baselines We evaluate NELLIE against **Entailer** [Tafjord *et al.*, 2022], another system that produces entailment tree proofs via backward chaining. Entailer stops recurring **not** when it finds entailing facts from a corpus, but rather when *the model*

[6]Due to batching, these correspond to 3 total calls to the HuggingFace library's `generate` function for the T5-3B model.

[7]If it produces no proofs, it gives no answer and is wrong. NELLIE searches for up to p=10 proofs of max depth d=5 with a timeout of t=180 seconds per option.

	Explanations		QA Accuracy (%)		
	Grounded	**Logical**	**Ovr**	**Easy**	**Chal**
EntailmentBank QA					
NELLIE (3B)	Yes	Yes	71.4	76.4	60.4
Entailer-3B					
(D = 3)	No	Yes	48.7	52.8	39.6
(D = 1)	No	No	64.9	71.2	50.9
Entailer-11B					
(D = 3)	No	Yes	73.2	77.3	64.2
(D = 1)	No	No	71.1	76.4	59.4
WorldTree QA					
NELLIE (3B)	Yes	Yes	71.4	75.7	60.9
Entailer-3B					
(D = 3)	No	Yes	45.2	50.1	33.3
(D = 1)	No	No	47.7	51.6	38.5
Entailer-11B					
(D = 3)	No	Yes	73.2	76.7	64.6
(D = 1)	No	No	74.1	78.7	63.0
PathNet	Yes	No	43.4		
TupleILP	Yes	No	49.8		
ExplanationLP	Yes	No	62.6		
Diff-Explainer	Yes	No	71.5		

Table 1: NELLIE performance vs. comparable XQA systems.

believes that a subquery is true with high confidence. Its trees are thus not grounded in verified facts. We reimplemented their algorithm using their T5-11B-based model, reporting their configuration of a max tree depth (**D**) of 3 and minimum of 1 (i.e. ≥ 1 decomposition). We also evaluate their ablated setting which recurs exactly once (D=1). This is a baseline that is neither grounded nor particularly interpretable, generating just a pair of statements. To isolate the impact of model size, we also evaluate an **Entailer-3B** model based on the same T5-3B model as NELLIE's DRG.[8] We also compare against grounded xQA methods without logical structure (see Fig 2): **PathNet** [Kundu *et al.*, 2019], which constructs 2-hop chains by linking entities between facts, and three approaches that build graphs from facts using linear programming: **TupleILP** [Khot *et al.*, 2017], **ExplanationLP** [Thayaparan *et al.*, 2021], and ***Diff*-Explainer** [Thayaparan *et al.*, 2022].[9]

5.1 Results

Table 1 shows QA performance. NELLIE's overall (**Ovr**) accuracy of over 70% matches that of the best-performing grounded baseline, *Diff*-Explainer, on WorldTree, and is within 2 points of the best logically directed baseline, Entailer-11B. This is notable given Entailer-11B is a much larger model and has no requirement to provide grounded proofs. NELLIE outperforms the same-sized Entailer-3B baseline by a margin of 22% on EntailmentBank and 26% on WorldTree under its default max D = 3 configuration, and even outperforms the less explainable D = 1 variant.

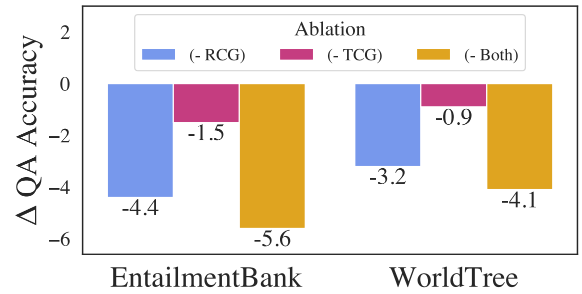

Figure 6: Effect of ablating one or both of rule-conditioned (RCG) and template-conditioned generation (TCG).

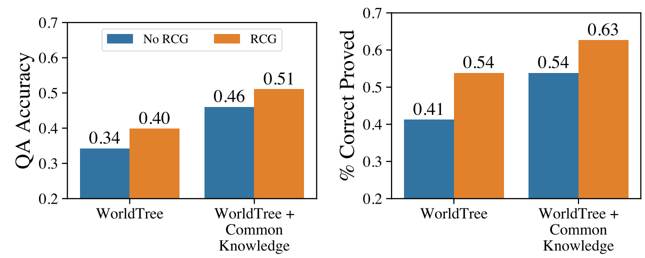

Figure 7: NELLIE QA accuracy and proof recall on OBQA with vs. without access to common knowledge statements

Figure 6 shows the results of ablating NELLIE's two conditional generation modules and replacing them with vanilla generation. Removing both template- and retrieval-conditioned generation lowers NELLIE's performance in all circumstances, highlighting the empirical benefit of structured guidance. Ablating TCG (($-$ TCG), and the drop from ($-$ RCG) to ($-$ Both)) reduces performance by 1-1.5 points. Ablating RCG (NELLIE vs ($-$ RCG)) drops it by 3-4.

5.2 Domain Generalization and Knowledge Scaling

While NELLIE was trained on datasets centered around WorldTree, we show that it can perform in another domain by altering the knowledge over which it reasons. We consider OpenBookQA [Mihaylov *et al.*, 2018], a dataset that requires reasoning over facts not included in WorldTree; each question is associated with one WorldTree science fact ("metals conduct electricity"), but also one fact from a separate pool of common knowledge ("a suit of armor is made of metal").

As with a classical expert system, we have designed NELLIE to be "improvable merely by making statements to it" [McCarthy, 1959]. This suggests that as we increase the provided knowledge store, we should expect NELLIE to reason more effectively. Because WorldTree does not cover all the knowledge required to answer OBQA questions, we test whether NELLIE performs better on them when we add the common knowledge to its retrieval index. As the common knowledge annotations are one fact per question and are not designed specifically for entailment, it is unlikely a priori that a fully grounded entailment tree can be created using the available

[8]We obtained the training data from Tafjord *et al.*, 2022 and trained our model in consultation with the authors.

[9]We show results from [Thayaparan *et al.*, 2021 & 2022].

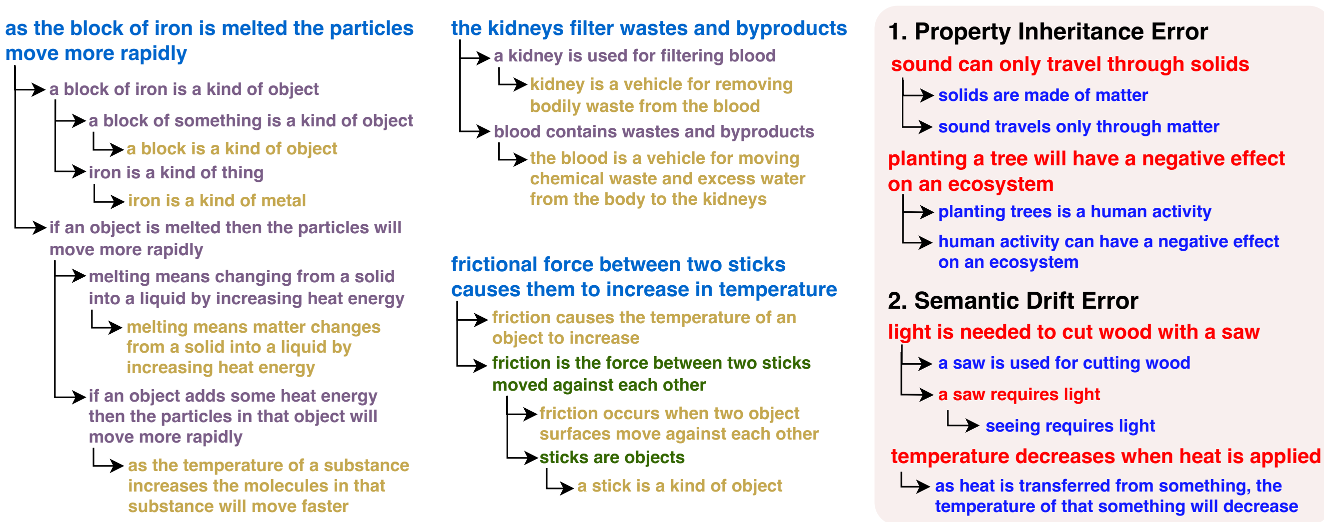

Figure 8: **(Left)** Example NELLIE proofs. **Top-level queries** are decomposed into subqueries via **retrieval-** or **template**-conditioned generation. Proof leaves are **corpus facts**. **(Right)** Common classes of error causing **false** statements to be grounded in **true** ones.

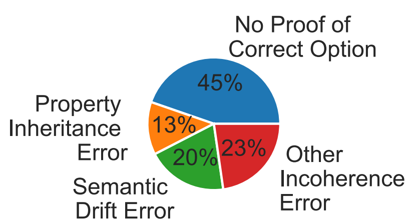

Figure 9: Distribution of NELLIE error classes among 136 incorrect answers on EntailmentBank questions

knowledge alone, making this a challenging task.

Figure 7 shows NELLIE's accuracy with and without RCG. We do not use TCG, as the targeted common knowledge is free-form. We find that QA accuracy increases 11-12% with the common knowledge added. The percent of correct statement proved also increases 9-12%. These results also highlight the importance of RCG, as it provides a 5% QA boost and 13% on correct proof rate. These results show that NELLIE can be applied out-of-the-box to a dataset requiring reasoning over a different source of free-form NL knowledge.

5.3 Tree Error Analysis

We find that NELLIE can produce high quality proofs of correct hypotheses; examples are shown in Figure 8 (left) and §F. However, the system can also generate proofs for incorrect answers that bypass its entailment filters. We can diagnose (and perhaps annotate) error patterns to address in future work. We list a few categories, depicted in Figure 8 (right). A common pattern of error is **Hypernym Property Inheritance Errors** (also known as the "fallacy of the undistributed middle"), in which the model assumes that a member of a taxonomic class has a property of their hypernym, but the property is not universal. A colloquial example is inferring *penguins can fly* from *penguins are birds* & *birds can fly*. We also observe an amount of **Errors from Semantic Drift** [Khashabi *et al.*, 2019] between inference hops, culminating in RTE model false positives. E.g., in block 2 of Fig 8 (right), which object loses temperature during heat transfer changes between hops. Figure 9 shows the distribution of these errors on a set of 136 incorrect answers from EntailmentBank. The predominant error case is a failure to find any proof of the correct option.

Correct Answer Tree Analysis To investigate whether NELLIE reaches a right solution with right vs wrong reasoning, we manually inspected for reasoning errors in 50 correct answer trees produced by NELLIE. We found that 39/50 (78%) of trees were perfectly acceptable. Most of the 11 unacceptable trees were due to incompleteness at one decomposition step.

6 Conclusion

We propose a reimagined version of a classical expert system that relies on the inferential power of LLMs rather than handcrafted rules. NELLIE has the skeleton of a symbolic theorem prover, but the provided rules invoke neural predicates that allow for systematic reasoning over NL statements. To dynamically generate inferences, we introduce two mechanisms for knowledge-guided generation: conditioning decompositions on retrieval or model-selected inference templates. Our search algorithm undergirds NELLIE, an explainable reasoning system that performs end-to-end QA by searching for grounded entailment trees. We find that NELLIE equals or exceeds the performance of comparable XQA systems, while providing explanations with the simultaneous guarantees of logical directionality and groundedness in human-provided text. This work thus suggests a new way to jointly reap the benefits of both modern neural methods and traditional reasoning.

Acknowledgements

Thanks to Nils Holzenberger, Kyle Richardson, Elias Stengel-Eskin, Marc Marone, Kate Sanders, Rachel Wicks, and Andrew Blair-Stanek for comments on earlier drafts. Thank you to Abbey Coogan for feedback on figure design.

References

[Andreas *et al.*, 2016] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[Arabshahi *et al.*, 2021a] Forough Arabshahi, Jennifer Lee, Antoine Bosselut, Yejin Choi, and Tom Mitchell. Conversational multi-hop reasoning with neural commonsense knowledge and symbolic logic rules. In *Proceedings of EMNLP*, 2021.

[Arabshahi *et al.*, 2021b] Forough Arabshahi, Jennifer Lee, Mikayla Gawarecki, Kathryn Mazaitis, Amos Azaria, and Tom Mitchell. Conversational neuro-symbolic commonsense reasoning. In *Proceedings of AAAI*, 2021.

[Bos and Markert, 2005] Johan Bos and Katja Markert. Recognising textual entailment with logical inference. In *Proceedings of HLT and EMNLP*, 2005.

[Bosselut *et al.*, 2021] Antoine Bosselut, Ronan Le Bras, and Yejin Choi. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. *AAAI*, 2021.

[Bostrom *et al.*, 2022] Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. Natural language deduction through search over statement compositions. In *Findings of ACL: EMNLP 2022*, 2022.

[Clark *et al.*, 2018] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457*, 2018.

[Clark *et al.*, 2021] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In *Proceedings of IJCAI*, 2021.

[Creswell and Shanahan, 2022] Antonia Creswell and Murray Shanahan. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*, 2022.

[Dalvi *et al.*, 2021] Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. In *Proceedings of EMNLP*, 2021.

[Dalvi Mishra *et al.*, 2022] Bhavana Dalvi Mishra, Oyvind Tafjord, and Peter Clark. Towards teachable reasoning systems: Using a dynamic memory of user feedback for continual system improvement. In *Proceedings of EMNLP*, 2022.

[Das *et al.*, 2021] Rajarshi Das, Manzil Zaheer, Dung Thai, Ameya Godbole, Ethan Perez, Jay Yoon Lee, Lizhen Tan, Lazaros Polymenakos, and Andrew McCallum. Case-based reasoning for natural language queries over knowledge bases. In *Proceedings of EMNLP*, 2021.

[Demszky *et al.*, 2018] Dorottya Demszky, Kelvin Guu, and Percy Liang. Transforming question answering datasets into natural language inference datasets. *arXiv:1809.02922*, 2018.

[Green *et al.*, 1961] Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: An automatic question-answerer. 1961.

[Gupta *et al.*, 2020] Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. Neural module networks for reasoning over text. In *International Conference on Learning Representations*, 2020.

[He *et al.*, 2022] Hangfeng He, Hongming Zhang, and Dan Roth. Rethinking with retrieval: Faithful large language model inference. *arXiv preprint arXiv:2301.00303*, 2022.

[Holtzman *et al.*, 2019] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2019.

[Hong *et al.*, 2022] Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. METGEN: A module-based entailment tree generation framework for answer explanation. In *Findings of ACL: NAACL 2022*, 2022.

[Jackson, 1986] P Jackson. Introduction to expert systems. 1986.

[Jansen and Ustalov, 2019] Peter Jansen and Dmitry Ustalov. TextGraphs 2019 shared task on multi-hop inference for explanation regeneration. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 2019.

[Jhamtani and Clark, 2020] Harsh Jhamtani and Peter Clark. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proceedings of EMNLP (EMNLP)*, 2020.

[Ji *et al.*, 2022] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 2022.

[Johnson *et al.*, 2019] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 2019.

[Jones *et al.*, 2000] Karen Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments - part 1. *Inf. Process. Manag.*, 2000.

[Jung *et al.*, 2022] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of EMNLP*, 2022.

[Kalyanpur *et al.*, 2021] Aditya Kalyanpur, Tom Breloff, David Ferrucci, Adam Lally, and John Jantos. Braid: Weaving symbolic and neural knowledge into coherent logical explanations. *AAAI*, 2021.

[Kautz and Selman, 1999] Henry Kautz and Bart Selman. Unifying sat-based and graph-based planning. In *IJCAI*, 1999.

[Kautz *et al.*, 1992] Henry A Kautz, Bart Selman, et al. Planning as satisfiability. In *ECAI*, 1992.

[Kazemi *et al.*, 2023] Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. LAMBADA: Backward chaining for automated reasoning in natural language. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023.

[Khashabi *et al.*, 2019] Daniel Khashabi, Erfan Sadeqi Azer, Tushar Khot, Ashish Sabharwal, and Dan Roth. On the possibilities and limitations of multi-hop reasoning under linguistic imperfections. *arXiv:1901.02522*, 2019.

[Khot *et al.*, 2017] Tushar Khot, Ashish Sabharwal, and Peter Clark. Answering complex questions using open information extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2017.

[Khot *et al.*, 2020] Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In *Proceedings of AAAI*, 2020.

[Khot *et al.*, 2021] Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Text modular networks: Learning to decompose tasks in the language of existing models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[Khot *et al.*, 2022] Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*, 2022.

[Kojima *et al.*, 2022] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv:2205.11916*, 2022.

[Kundu *et al.*, 2019] Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. Exploiting explicit paths for multi-hop reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[Lewis *et al.*, 2019] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv:1910.13461*, 2019.

[Lyu *et al.*, 2023] Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. In *ACL*. Association for Computational Linguistics, 2023.

[Lyu *et al.*, 2024] Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. Towards Faithful Model Explanation in NLP: A Survey. *Computational Linguistics*, 2024.

[Madaan *et al.*, 2021] Aman Madaan, Niket Tandon, Dheeraj Rajagopal, Peter Clark, Yiming Yang, and Eduard Hovy. Think about it! improving defeasible reasoning by first modeling the question scenario. In *Proceedings of EMNLP*, 2021.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumancic, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. Deepproblog: Neural probabilistic logic programming. *NIPS*, 2018.

[McCarthy, 1959] John McCarthy. Programs with common sense, 1959.

[Metaxiotis *et al.*, 2002] Kostas S Metaxiotis, Dimitris Askounis, and John Psarras. Expert systems in production planning and scheduling: A state-of-the-art survey. *Journal of Intelligent Manufacturing*, 2002.

[Mihaylov *et al.*, 2018] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[Musen and Van der Lei, 1988] Mark A Musen and Johan Van der Lei. Of brittleness and bottlenecks: Challenges in the creation of pattern-recognition and expert-system models. In *Machine Intelligence and Pattern Recognition*. 1988.

[Olausson *et al.*, 2023] Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[Pan *et al.*, 2019] Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of EMNLP-IJCNLP*, 2019.

[Pan *et al.*, 2023] Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of ACL: EMNLP 2023*, 2023.

[Picco *et al.*, 2021] Gabriele Picco, Thanh Lam Hoang, Marco Luca Sbodio, and Vanessa Lopez. Neural unification for logic reasoning over natural language. In *Findings of ACL: EMNLP 2021*, 2021.

[Polu and Sutskever, 2020] Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *ArXiv*, 2020.

[Raffel *et al.*, 2020] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020.

[Riazanov and Voronkov, 2002] Alexandre Riazanov and Andrei Voronkov. The design and implementation of vampire. *AI communications*, 2002.

[Ribeiro *et al.*, 2022] Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henry Zhu, Xinchi Chen, Zhiheng Huang, Peng Xu, Andrew O. Arnold, and Dan Roth. Entailment tree explanations via iterative retrieval-generation reasoner. In *NAACL-HLT*, 2022.

[Rocktäschel and Riedel, 2017] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. *Advances in neural information processing systems*, 2017.

[Russell and Norvig, 2010] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2010.

[Saha *et al.*, 2023a] Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. Branch-solve-merge improves large language model evaluation and generation, 2023.

[Saha *et al.*, 2023b] Swarnadeep Saha, Xinyan Yu, Mohit Bansal, Ramakanth Pasunuru, and Asli Celikyilmaz. MURMUR: Modular multi-step reasoning for semi-structured data-to-text generation. In *Findings of ACL: ACL 2023*, 2023.

[Schank, 1983] Roger C Schank. *Dynamic memory: A theory of reminding and learning in computers and people*. 1983.

[Schubert, 2015] Lenhart Schubert. Semantic representation. *Proceedings of AAAI*, 29(1), Mar. 2015.

[Sessa, 2002] Maria I Sessa. Approximate reasoning by similarity-based sld resolution. *Theoretical computer science*, 2002.

[Sprague *et al.*, 2022] Zayne Sprague, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Natural language deduction with incomplete information. In *Proceedings of EMNLP*, 2022.

[Stern *et al.*, 2010] Roni Stern, Tamar Kulberis, Ariel Felner, and Robert Holte. Using lookaheads with optimal best-first search. In *Proceedings of AAAI*, 2010.

[Tafjord *et al.*, 2021] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of ACL: ACL-IJCNLP 2021*, 2021.

[Tafjord *et al.*, 2022] Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of EMNLP*, 2022.

[Thayaparan *et al.*, 2021] Mokanarangan Thayaparan, Marco Valentino, and André Freitas. Explainable inference over grounding-abstract chains for science questions. In *Findings of ACL: ACL-IJCNLP 2021*, 2021.

[Thayaparan *et al.*, 2022] Mokanarangan Thayaparan, Marco Valentino, Deborah Ferreira, Julia Rozanova, and André Freitas. Diff-explainer: Differentiable convex optimization for explainable multi-hop inference. *TACL*, 2022.

[Valentino *et al.*, 2022] Marco Valentino, Mokanarangan Thayaparan, and André Freitas. Case-based abductive natural language inference. In *Proceedings of the 29th International Conference on Computational Linguistics*, 2022.

[Weber *et al.*, 2019] Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. NLProlog: Reasoning with weak unification for question answering in natural language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.

[Wei *et al.*, 2022] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv:2201.11903*, 2022.

[Welleck *et al.*, 2022] Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. Naturalprover: Grounded mathematical proof generation with language models. *arXiv:2205.12910*, 2022.

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

[Xia *et al.*, 2024] Yu Xia, Rui Wang, Xu Liu, Mingyan Li, Tong Yu, Xiang Chen, Julian McAuley, and Shuai Li. Beyond chain-of-thought: A survey of chain-of-x paradigms for llms, 2024.

[Xie *et al.*, 2020] Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.

[Yadav *et al.*, 2019] Vikas Yadav, Steven Bethard, and Mihai Surdeanu. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of EMNLP-IJCNLP*, 2019.

[Yang *et al.*, 2022] Kaiyu Yang, Jia Deng, and Danqi Chen. Generating natural language proofs with verifier-guided search. In *Proceedings of EMNLP*, 2022.

[Ye *et al.*, 2023] Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. SatLM: Satisfiability-aided language models using declarative prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Zelle and Mooney, 1996] John M. Zelle and Raymond J. Mooney. Learning to parse database queries using inductive logic programming. In *Proceedings of AAAI*, 1996.