Dynamic Generation of Interpretable Inference Rules
in a Neuro-Symbolic Expert System

Nathaniel Weir
Johns Hopkins University
nweir@jhu.edu

Benjamin Van Durme
Johns Hopkins University
vandurme@jhu.edu

Abstract

We present an approach for systematic reasoning that produces human interpretable proof trees grounded in a factbase. Our solution evokes classic Prolog-based inference engines, where we replace handcrafted rules through a combination of neural language modeling, guided generation, and semiparametric dense retrieval. This novel reasoning engine, NELLIE, dynamically instantiates interpretable inference rules that capture and score entailment (de)compositions over natural language statements. NELLIE shows competitive performance on scientific QA datasets requiring structured explanations over multiple facts.

1 Introduction

We explore whether neural NLP models can emulate the behavior of expert systems (Jackson, 1986; Metaxiotis et al., 2002), i.e., symbolic inference engines that reason over a knowledge base (KB) of facts and axioms expressed in a logical formalism. Expert systems are appealing in part for their explainable behavior: decisions are made through a process of systematically constructing a well-formed symbolic proof predicated upon explicit knowledge carefully enumerated by an engineer under consultation with a domain expert.

As expert systems are brittle, can be prohibitively expensive, and are time-consuming to curate (Musen and Van der Lei, 1988), the AI community has turned to neuro-symbolic mechanisms that incorporate large language models (LLMs)—which might reason over natural language (NL)—into otherwise symbolic reasoning procedures. Reasoning over NL representations eschews the need to adhere to a single symbolic formalism and introduces the possibility of using the inferential power of pretrained models. It has been suggested that LLMs can perform something akin to logical reasoning (Clark et al., 2021) and can implicitly capture the world knowledge reflected in the language corpora on which they are trained (Petroni et al., 2019), and that such world knowledge can be queried, retrieved, and reasoned about under careful supervision (Shwartz et al., 2020; Talmor et al., 2020). However, such reasoning is hardly interpretable, as LLMs are notoriously black box systems. We are only in the early stages of probing models for what they might “believe” about the world, let alone rely upon such belief during systematic reasoning (Tafjord et al., 2022).

This motivates our pursuit of an inference procedure that uses LLMs and their reasoning capacity, but which constructs explicit knowledge-grounded proofs that are interpretable to humans and grounded in verifiable facts. This work introduces NELLIE, the Neurosymbolic Large LM Inference Engine, a QA system which leverages
recent LLMs as proposal functions in the search of proofs grounded in an externally provided NL factbase. The “symbols” that our neuro-symbolic inference engine reasons over are mostly free-form NL sentences. Rather than require a knowledge engineer to carefully write hundreds of inference rules as in the classical expert system setting, NELLIE employs neural generation and dense retrieval as dynamic rule generators (DRGs; Kalyanpur et al., 2021), producing conjunctions of premises that, if themselves proved through provenance retrieval or further decomposition, will prove an input NL query via compositional entailment. An example of this procedure is depicted in Figure 1.

Our DRGs make use of the semi-structured content in NELLIE’s NL factbase using a pair of methods for knowledge-infused text generation: forced decoding of sentences acquired via learnt dense retrieval, and guided generation conditioned on infilling templates that reflect inference-supporting syntactic types drawn from the tables underlying the KB (Jansen et al., 2018). These techniques bias the proof search towards trees that are likely provable given the available facts, reflecting the capacity for LMs to infuse high level domain structure into text-based reasoning wherever it is available.

The system is built upon a backward chaining symbolic theorem prover written in Prolog, but ultimately requires only a few handwritten meta-rules specifying the high-level structure of NL-based inference rules. The meta-rules pertain to checking for entailment of a hypothesis against the provided factbase, or decomposing it into a conjunction of support facts to recursively prove. In order to treat NL sentences as if they were symbols in a purely symbolic proof search algorithm, we use a form of weak unification (Sessa, 2002; Weber et al., 2019) between a queried fact and one in the factbase.

NELLIE is appealingly modular. We use a heterogeneous mixture of neural models for generating, retrieving, and verifying premises for entailment hops, which are formalized structurally as conjunctive Prolog-based search steps. We strive to compartmentalize and separately optimize the generative and discriminative decision points of the reasoning process: a seq2seq model proposes statements via stochastic sampling, which are then verified as logically coherent using a mixture of models trained to recognize textual entailment (RTE).

The resulting inference engine produces coherent and interpretable proofs akin to the compositional entailment trees of EntailmentBank (Dalvi et al., 2021), while performing competitively even while required to “explain its work” by grounding a logical decision proof in an external NL factbase.

Our analysis explores the impact of NELLIE’s knowledge-guided generation techniques on proof search and QA performance. We identify avenues for future improvement of modular components, elucidated by granular consideration of error cases.

**Contribution** This work introduces NELLIE, a neoclassical backward-chaining inference engine that reasons over NL statements using finetuned LMs, guided generation, and dense retrieval from explicit knowledge. While recent work on explanation tree generation has considered the task of proving known-to-be-true statements given small sets of gold support facts (Dalvi et al., 2021; Bostrom et al., 2022), or generating proofs grounded in model beliefs (Tafjord et al., 2022), we instead focus specifically on fully interpretable, KB-grounded question answering as proof search. We impose the requirement that all our QA model’s decisions must be entailment-based proof trees fully grounded in an external factbase; in this challenging scenario, NELLIE achieves competitive, fully interpretable performance on QA datasets for which such proof trees are not always guaranteed.

## 2 Related Work

**Neural Theorem Proving over Language** A long-standing approach to reasoning over language is to reason over the projection of language into a symbolic form, such as for question answering (Green et al., 1961; Zelle and Mooney, 1996) or verifying entailments (Bos and Markert, 2005). Provided the translation from language to symbolic representation can be done at high accuracy, then one can leverage very fast and scalable solutions for discrete symbolic inference (Riazanov and Voronkov, 2002; Kautz et al., 1992; Kautz and Selman, 1999). Unfortunately accurately translating broad-domain natural language into an adequately expressive representation for reasoning is a challenge (Schubert, 2015).

Recent work has explored methods of working with language in ways that do not require accurately mapping to other discrete representations. Some use a LM to generate proof steps in mathematical theorem proving (Polo and Sutkewer, 2020; Welleck et al., 2022). Variants of neural theorem provers (NTPs; Rocktäschel and Riedel, 2017) such
as NLProlog (Weber et al., 2019) attempt to shoe-horn vector representations of NL into symbolic reasoning by learning continuous representations of symbols in a theorem prover. Such an approach yields interpretable usage of facts and symbols, but the learnt symbols do not have a verbalizable meaning. Kalyanpur et al. (2021) inject neural reasoning into a Boxer/Prolog-based symbolic reasoner via a special predicate to make external calls to an LM.

Another line of work has explored the capacity for transformers to emulate stepwise (Tafjord et al., 2021) or end-to-end (Clark et al., 2021; Picco et al., 2021; Saha et al., 2021) logical reasoning over small rulebases converted into NL. This “rule compiling” (Kautz, 2022) type of approach, which has thusfar considered only simple conjunctive rules, faces scalability and reliability issues.

**Modular Reasoning over NL**  NELLIE’s systematic reasoning is related to recent 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 using separate modules to perform span extraction and arithmetic operations. Khot et al. (2021) introduce another NMN variant that decomposes a question into simpler ones answerable by existing models. Systematic question decomposition is also explored in Talmor and Berant (2018); Min et al. (2019); Wolfson et al. (2020).

**Systematic Explanation Generation**  Recent work has tackled structured explanation generation, in which a high-level hypothesis is supported by a reasoning chain of inference operations over natural language. One strategy, coined “chain of thought” prompting (Wei et al., 2022; Kojima et al., 2022), elicits intermediate inference hops from LMs before they generate an answer. The recent 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 previous works on explainable reasoning that treat a KB-retrieved set of support statements as an explanation, stopping short of showing their role in logical entailment (Pan et al., 2019; Jansen and Ustalov, 2019; Yadav et al., 2019; Valentino et al., 2022; Thayaparan et al., 2021).

Components of NELLIE are related to concurrent approaches for explanation tree construction (Bostrom et al., 2022; Hong et al., 2022). Ribeiro et al. (2022) also use iterative retrieval-conditioned generation, and Yang et al. (2022); Tafjord et al. (2022) also use entailment classifiers to filter proof steps. Few tree generation approaches consider the harder scenario of multiple-choice QA as NELLIE does, opting instead to focus on the reconstruction task from support facts. Tafjord et al. (2022) do consider the harder task; their work is most similar to ours. They propose a backward chaining QA system that generates entailment trees (not grounded in human-verified facts) using a search driven by internal model belief of factuality, complementary to NELLIE’s use of guided and retrieval-conditioned generation.

**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 used in Prolog solvers will, upon performing variable substitution, recursively attempt to prove the terms in the rule’s *antecedent*. For example, a disease classification system might prove query **CONTAGIOUS**(flu) via facts **CONTAGIOUS**(influenza) and **OTHERNAME**(flu, influenza), and conjunctive rule **CONTAGIOUS**(X) ⇔ **OTHERNAME**(X, Y) ∧ **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); Weber et al. (2019) for a broader overview.

**Neural Predicates**  While most declarative predicates have meaning only in the context of user-defined inference axioms, others can call external modules that produce values for their arguments and/or determine the truth value of the predicate. NELLIE makes heavy use of this feature for defining neural model-invoking predicates. In the above example, we might train a seq2seq to produce other names for a disease Y, turning **OTHERNAME**(Y⁺, X⁻)\(^1\) into a neural predicate. This mechanism opens up the capacity to introduce externally-defined object symbols, e.g. seq2seq-

\(^1\)In Prolog syntax, `+` denotes inputs, `−` outputs.
4 NELLIE Overview

Depicted in Figure 2, NELLIE is comprised of: An external factbase (some \{f_1, \ldots, 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 use as the factbase the WorldTree corpus (Xie et al., 2020), a set of 81 relational tables whose rows concatenate into 9K inference-supporting science facts.

4.1 Question Conversion

Given a multiple-choice question, NELLIE takes a candidate answer and converts the QA pair into a single hypothesis statement \( h \) using a neural model for Question to Declarative Sentence conversion (QA2D; Demszky et al., 2018). It then searches for a proof of \( h \) against its knowledge base (See §B). 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.

4.2 Inference Rule Structure

NELLIE dynamically instantiates inference rules given a query hypothesis. The structure of such a rule is strikingly simple, emulating one of the following templates:

I. Hypothesis \( \leftarrow \) Fact

II. Hypothesis \( \leftarrow \) Fact1 \& 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 entail the hypothesis. Template I is given higher search precedence than II, yielding a rather intuitive high-level procedure: NELLIE first looks up the hypothesis against its factbase, searching for an entailing fact. If it does not find one, it decomposes the hypothesis into a pair of statements to be proved. Concretely, for an input hypothesis \( h \), we define the predicate \( \text{PROVE}(h) \) that serves as the primary goal term. We define the following core meta-rules, which use the neural predicates \( \text{RETREIVE}, \text{ENTAILS}, \text{and RULEGEN} \):

**Weak Unification** In classical backward chaining, a unification operator is used to assign equivalence to two logical atoms; this can only occur between two atoms if they are of 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. For comparing atoms with multiple symbols, Weber et al. (2019) estimate a unification score as the aggregation of pairwise similarity scores using a similarity function \( s_1 \approx_\theta s_2 \in [0, 1] \) parameterized by model \( \theta \). The score of an entire proof is taken to be the minimum of scores across all steps in the proof. In this work, we apply a similar aggregation of unification scores to the search for proof trees rooted an NL factbase. We say that a query fact \( s_1 \) “weakly unifies” with provenance fact \( s_2 \) with unification score equal to the confidence of an NLI model taking \( s_2 \) as the premise and \( s_1 \) as the hypothesis.
1. **Fact Unification**

\[ \text{PROVE}(h) \iff \text{RETRIEVE}(h^+, f^-) \land \text{ENTAILS}(f, h) \]

2. **Two Premise Rule Generation**

\[ \text{PROVE}(h) \iff \text{RULEGEN}(h^+, f_1^+, f_2^-) \land \text{ENTAILS}([f_1, f_2], h) \land \text{PROVE}(f_1) \land \text{PROVE}(f_2) \]

3. **Retrieved First Premise Rule Generation**

\[ \text{PROVE}(h) \iff \text{RETRIEVE}(h^+, f_1^-) \land \text{RULEGEN}(h^+, f_1^+, f_2^-) \land \text{ENTAILS}([f_1, f_2], h) \land \text{PROVE}(f_2) \]

At each step in the backward-chaining search, NELLIE’s Prolog engine attempts unify a query with the head of one of these three rules.

### 4.3 Unification with Retrieved Facts

For factbase fact \( f \), \( \text{PROVE}(f) \) is vacuously true. Rule 1 shows how we prove \( \text{PROVE}(h) \) using retrieval. The predicate \( \text{RETRIEVE} \) proposes candidate \( f_1 \)’s given \( h \) using a FAISS (Johnson et al., 2019)-based nearest neighbor dense retrieval index. To promote logical coherence, we apply a set of differently-trained neural RTE models as filters that iteratively whittle away \( f_1 \) candidates that are not classified as entailing \( h \).

\[ \text{ENTAILS}(f, h) \iff \bigwedge_{j=1}^{n} \text{ENTAILS}_j(f, h) \]

Implicit in rule 1 is that \( \text{PROVE}(h) \) weakly unifies with some \( \text{PROVE}(f_1) \); we assign the unification score \( \theta(h, f_1) \) equal to the confidence (normalized distance from 0.5) of one RTE model.

### 4.4 Dynamic Rule Generator (DRG)

If NELLIE fails to retrieve an entailing fact for hypothesis \( h \) as decided the RTE filters, it attempts to decompose \( h \) into a pair of entailing premises using nucleus sampling (Holtzman et al., 2019) from a neural seq2seq model to propose candidates. The predicate \( \text{RULEGEN}(h, f_1, f_2) \) prompts a model trained on \( h \to f_1, f_2 \) pairs. The space of potential decompositions of a given hypothesis is extremely large; as we want to bias the proof search towards statements that can be grounded in the facts of NELLIE’s factbase, we adopt a two-pronged approach to dynamic rule generation, illustrated in inference rules 2 and 3 and described below.

**Structurally Guided Generation** In rule 2, part of the set of candidate pairs results from sampling \( f_1, f_2 \) from the DRG conditioned on \( h \). The rest is sampled using template-guided generation, which leverages the semi-structure of NELLIE’s factbase, WorldTree. WorldTree tables correspond to types of facts with similar syntax and semantics that are likely to support systematic reasoning. Each table can be viewed as an \( n \)-ary relation of columns whose values are text spans. For example, the Taxonomic relation has the columns \(<A>, [HYPERNYM], <is a kind of>, [HYPERNYM], <for>, [PURPOSE]\). Rows include ‘a bird is an animal’ and ‘a seed is a kind of food for birds.’

The DRG model is a seq2seq model finetuned to accept an optional template specifying this sort of semi-structured language. It accepts any masked infilling template (e.g. \(<\text{mask}>\) is a kind of \(<\text{mask}>\)), akin to those used to pretrain recent LMs (Lewis et al., 2019; Raffel et al., 2020), and it proposes decompositions that reflect such syntax. We only have the DRG model condition on templates for the \( f_1 \) fact, as \( f_2 \) should follow from \( f_1 \) and \( h \) without the need for guidance. We thus create a \( h, t_1 \to f_1, f_2 \) model. We feed the model templates drawn from WorldTree’s tables, guiding it towards proof steps more likely to be grounded in the factbase. 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, a second samples \( n \) candidates for each of 50 templates (see §C).

\[ \text{RULEGEN}(h^+, f_1^+, f_2^-) \iff \text{MEMBER}(t, \text{WTTEMPLATES} \cup \{\text{BLANK}\}) \land \text{TCRULEGEN}(h^+, t^+, f_1^+, f_2^-) \]

**Retrieval Conditioned Generation** In rule 3, rather than rely upon the model to generate a pair of decomposition statements, we immediately ground half of the antecedent by choosing \( f_1 \) to be a fact retrieved directly from the factbase using the \text{RETRIEVE} predicate. We have the generator force decode \( f_1 \) before producing \( f_2 \) as normal via nucleus sampling. When NELLIE uses rule 2a, the unification score equals the minimum of the scores for \( \text{PROVE}(f_1) \) and \( \text{PROVE}(f_2) \). For 2b, since \( f_1 \) is retrieved, the score equals that of \( \text{PROVE}(f_2) \).

**Filters** As stochastic sampling from language models is a particularly noisy process, a large fraction of generated candidates are invalid, meaning that the generated facts would not, even if themselves proved, compositionally entail \( h \). Accord-

---

This intuition follows from the observation that over 85% of the non-leaf nodes in the trees of EntailmentBank (Dalvi et al., 2021), a dataset of gold proof trees rooted in WorldTree facts, also reflect the syntax of a WorldTree table.
ingly, we introduce a suite of compositional entailment verifiers (Khot et al., 2020; Jhamtani and Clark, 2020) that function analogously to the retrieval module’s; the models are trained specifically to validate two-premise compositional entailment.

4.5 Proof Search

Given a query, we allow NELLIE to search breadth-first for $t$ seconds to find up to $p$ proofs of depth $d$ or less. We follow the approach by Weber et al. (2019) of pruning search branches whose unification score is guaranteed to fall below the current best, given our monotonic aggregation function $\min(\cdot)$.

5 Model Details

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.

5.1 Data Sources

WorldTree Explanations (Xie et al., 2020) is a subset of the A12 Reasoning Challenge (ARC) QA dataset (Clark et al., 2018) annotated with structured explanations, comprising. Explanations are undirected graphs whose nodes are facts from the WorldTree tablestore.

EntailmentBank (Dalvi et al., 2021) is a dataset of handcrafted explanation trees for declarativized answers to 1840 ARC questions. Trees show how a given statement can be derived via compositional entailment hops starting from WorldTree facts.

eQASC (Jhamtani and Clark, 2020) is a crowdsourced dataset containing single-hop, two-fact compositional entailment questions. An expansion of QASC (Khot et al., 2020), eQASC comprises 21K positive and 59K negative-labeled $\{(f_1, f_2, h)\}$ triplets. $f_1$ and $f_2$ are drawn from 17M sentences scraped from science text.

SciTail (Khot et al., 2018) is a textual entailment dataset pulled from a combination of science exam question text and retrieved web sentences.

5.2 Architecture and Training

Question Converter NELLIE’s QA2D model is a T5-Large model (Raffel et al., 2020) trained on the original data from Demszky et al. (2018),\(^3\) plus the QA/H pairs from EntailmentBank.

Retrieval Module The dense retrieval model is a SentenceTransformers Siamese BERT-Network encoder (Reimers and Gurevych, 2019) pretrained on the MS MARCO IR corpus (Nguyen et al., 2016), which we fine-tune via ranking loss to maximize the cosine similarity between a hypothesis and its support facts.\(^4\) We gather $h, f_1$ from the WorldTree, EntailmentBank, and eQASC training sets, then sample random negatives during training.

Dynamic Rule Generator The DRG is a T5-Large model (Raffel et al., 2020) fine-tuned on $h \rightarrow [f_1, f_2]$ pairs drawn from the positive-labeled triplets from the eQASC training set, which are numerous (26K) but noisy, and the binary composition steps from the EntailmentBank training set, which are few (3.8K) but ultimately rooted in facts from WorldTree. To make the DRG able to condition on infilling templates, we adopt the span masking strategy from Raffel et al. (2020) to create a randomly masked version $\hat{f}_1$ of each $f_1$ in the training set and train the model on $h, \hat{f}_1 \rightarrow f_1, f_2$ pairs. For each original triplet, we create one training example with a masked template, and one in which $\hat{f}_1$ is an empty <mask>, giving the model the dual capacity for non-template-generation.

Entailment Filters As we desire high fidelity explainable reasoning, we want high precision filtering of generated inference hops. We find that a mixture-of-architectures setup approaches this goal. For 1-premise filtering for the unification module and 2-premise filtering for the DRG module, we fine-tune a separate pair of models consisting of a SentenceTransformers pretrained Cross-Encoder classifier and a T5-large seq2seq. The former is trained via classification loss, the latter via sequence cross entropy loss. The 1-premise filters are trained on the SciTail dataset, while the 2-premise filters are trained on positive and negative compositional entailment $(h, [f_1, f_2])$ examples from eQASC and EntailmentBank.

6 Experimental Setup

We evaluate NELLIE on multiple-choice question-answering datasets constructed so that correct answers are supported by facts in the WorldTree corpus. We use the version of the corpus constructed

\(^3\)Acquired from the authors of Tafjord et al. (2022).

\(^4\)We find that a sentence encoder trained to maximize the cosine of $h, f_1$ in which $f_1$ does not necessarily fully entail $h$ can still effectively serve dual purposes, both to retrieve the first of a $f_1, f_2$ pair in rule 2b, as well as to retrieve and score the weak unification of directly entailing single facts in rule 1.
Out Scored

| QA Accuracy (%) | % Ans | % Out Scored |
|-----------------|-------|--------------|
| Easy | Chal | Explainable | Logical |

### Results

| EntailmentBank QA |
|------------------|
| NELLIE | 89.8 | .317 | .799 | 23.1 | 56.7 | 60.9 | 46.3 | Yes | Yes |
| (– Templates) | **91.4** | .308 | .829 | 25.9 | 57.0 | 58.9 | **52.1** | Yes | Yes |
| (– Retrieval) | 88.4 | .331 | .778 | 20.9 | 57.0 | 60.3 | 48.8 | Yes | Yes |

| WorldTree QA 2.0 |
|------------------|
| NELLIE | 88.4 | .311 | .763 | 23.4 | 53.0 | 54.4 | 44.6 | Yes | Yes |
| (– Templates) | **90.5** | .305 | .796 | 24.9 | **54.7** | **58.3** | **46.0** | Yes | Yes |
| (– Retrieval) | 86.5 | .318 | .732 | 21.3 | 51.9 | 54.8 | 44.7 | Yes | Yes |

| WorldTree QA 1.0 |
|------------------|
| NELLIE | 81.1 | .302 | .671 | 20.3 | 46.8 | 51.7 | 35.8 | Yes | Yes |
| (– Templates) | **82.4** | .297 | .702 | 22.6 | 47.5 | 52.7 | 36.0 | Yes | Yes |
| (– Retrieval) | 78.3 | .307 | .640 | 18.9 | 45.2 | 48.9 | 36.9 | Yes | Yes |
| PathNet (Kundu et al., 2019) | 43.5 | 47.6 | 33.5 | Partial | No |
| CB-ANLI (Valentino et al., 2022) | 55.2 | 60.4 | 43.6 | Partial | No |
| ExplanationLP (Thayaparan et al., 2021) | 61.6 | 66.2 | 50.1 | Partial | No |

Table 1: QA results on EntailmentBank and WorldTree test sets.

by Dalvi et al. (2021), which contains an additional 3000 facts added by EntailmentBank annotators. NELLIE searches for up to $p=3$ proofs of max depth $d=4$ with a timeout of $t=300$ seconds per option. §A lists further hyperparameter information.

**EntailmentBank QA** We convert the EntailmentBank test set, initially designed to test tree reconstruction, into a QA dataset by retrieving the corresponding multiple-choice ARC questions from which the EntailmentBank hypotheses were originally constructed. We filter this set down to the 278 questions that are fully grounded in the extended WorldTree corpus, excluding questions whose gold trees have fact leaves that are user-added, non-generic facts.\(^5\) We refer to this test set as EBQA.

**WorldTree QA** We also evaluate on two releases of the WorldTree QA test set: the most recent 1664-question v2.0, but also the 1247-question v1.0 for comparability with previous methods. We refer to these different splits as WT1 and WT2 for brevity. WT2 contains about 65% of the questions from WT1, but is expanded to include middle school level questions while WT1 contains only elementary level. We note that annotated explanations for WorldTree questions are not trees, nor directed, nor based on entailment relations; there is no guarantee that proof trees exist for these questions.

**Metrics** The primary performance metric is predicted answer **accuracy**: whether the model generated a proof of the correct option that scored higher than any other. (If it produces no proofs, it does not give an answer). We also measure proof **precision/recall**, where a false positive is a proof of (not the selection of) a wrong answer, the **answered** rate measuring whether the model predicted a proof for any option, and the rate at which a correct answer proof was **outscored** by another option. Results are grouped according to ARC question difficulty.

### 7 Results

Table 1 shows NELLIE’s performance on the QA datasets averaged across 3 random seeds. We compare against ablations of NELLIE’s template- and retrieval-conditioned rule generation modules. We compare accuracy on the WorldTree 1.0 test set against recent approaches that construct non-proof-tree explanations: PathNet (Kundu et al., 2019), which constructs linear inference chains by linking entities between support facts, CB-ANLI (Valentino et al., 2022), which scores an answer by retrieving a small ($n=2$) set of facts and aggregating their explanatory relevance and semantic plausibility, and ExplanationLP (Thayaparan et al., 2021), which extracts explanation subgraphs from retrieved support facts using linear program-
as the block of iron is melted the particles move more rapidly

If an object is melted then the particles will move more rapidly

Melt means that the particles in that object will move more rapidly

If an object is melted then the particles will move more rapidly

As the temperature of a substance increases the molecules in that substance will move faster

Frictional force between two sticks causes them to increase in temperature

Putting a plant in a compost pile will conserve resources

Persisting in the force between two sticks moved against each other

Putting a plant in a compost pile will conserve resources

The force is the force between two sticks moved against each other

Recycling reduces the resources needed to make something

Placing bean plants in a compost pile will help conserve resources

A compost pile is used for recycling plants

A stick is a kind of object

A compost pile is used for recycling plants

A stick is a kind of object

Figure 3: Example correct proofs generated by NELLIE. Top-level query hypotheses are decomposed into subqueries via retrieval- and template-conditioned generation. Proof tree leaves are factbase facts.

Table 2: Proof recall for EntailmentBank tree nodes.

|                | NELLIE (− Templ.) | NELLIE (− Retr.) |
|----------------|-------------------|------------------|
| All Nodes      | .898              | .918             |
| Hypothesis     | .833              | .878             |

Table 7.1 Proving Gold Facts and Hypotheses

NELLIE’s overall (Ovr) accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall (Ovr) accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impose any explicit logical structure (i.e., something akin to a compositional entailment hop) on the generated explanation.

NELLIE’s overall accuracy is substantially above chance (25%) on all datasets, even though WorldTree questions do not necessarily have fact-grounded gold proof trees; let alone impos
8 Discussion and Future Work

We have crafted NELLIE specifically with intent to create systematic, interpretable, and granular reasoning behavior. This combined with our modularizing the generation and discrimination of candidates leads to our being able to diagnose a set of explicit, promising paths forward towards improving the system’s reasoning capacity.

Better Entailment Filters This work has been concerned with mechanisms for improving the generative components of the hypothesis search: retrieval- and template-conditioned generation as a means to guide the search towards factbase-rooted proofs. These improvements will be better realized if the discriminative components of the system are less likely to accept improper entailment hops. A perfectly-functioning entailment filter would also improve search efficiency, as much of NELLIE’s budgeted search runtime is currently allocated to exploring the branches created by these incorrect judgments. Reallocating this time to more promising, correct branches would improve proof recall.

Faster, Smarter Search Strategies The results reported in this work are the product of a straightforward breadth-first proof search by NELLIE, making iterative calls to a neural sequence generation module that takes multiple seconds. Template-conditioned generation, which improves the quality of the search horizon, takes even longer. Imposing our timeout of 300s does not give the system adequate time to explore its entire generated search horizon. Future work should consider more informed ways to select candidates upon which to recur, so as to make better use of the time budget.

Better Declarativization Identifying a single hypothesis statement given a potentially multi-sentence question context and answer is a challenging task, and our neural seq2seq approach for QA2D conversion is far from perfect. As the entire inference procedure by NELLIE is predicated on the correctness of this decision point, it follows that performance on the fraction of the dataset for which the QA2D model errs would improve as the model improves.
9 Conclusion

We have introduced NELLIE, a reimagined version of a classical expert system that relies on the inferential power of neural LLMs rather than hand-crafted inference rules. The inference engine has the skeleton of a classical symbolic theorem prover, but the provided symbolic rules invoke neural predicates that allow for systematic reasoning over natural language statements. In order to dynamically generate inference rules, we introduce two mechanisms for knowledge-guided generation: conditioning on dense retrieval and inference-supporting fact templates. We find that even in the challenging scenario in which decisions must be fully grounded in a factbase-rooted proof tree, NELLIE performs competently on a pair of science QA dataset. Finally, we identified a number of promising avenues for future work that builds upon this paradigm, made tractable by NELLIE’s modular approach to systematic neuro-symbolic reasoning.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Op-tuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pages 2623–2631.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 39–48.

Johan Bos and Katja Markert. 2005. Recognising textual entailment with logical inference. In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pages 628–635, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Kaj Bostrom, Zayne Sprague, Swarat Chaudhuri, and Greg Durrett. 2022. Natural language deduction through search over statement compositions. arXiv preprint arXiv:2201.06028.

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

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, pages 3882–3890.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. arXiv preprint arXiv:1809.02922.

Bert F. Green, Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. 1961. Baseball: An automatic question-answerer. In Papers Presented at the May 9-11, 1961, Western Joint IRE-AIEE-ACM Computer Conference, IRE-AIEE-ACM ’61 (Western), page 219–224, New York, NY, USA. Association for Computing Machinery.

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

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In International Conference on Learning Representations.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. METGEN: A module-based entailment tree generation framework for answer explanation. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1887–1905, Seattle, United States. Association for Computational Linguistics.

P Jackson. 1986. Introduction to expert systems.
Peter Jansen, Kelly J. Smith, Dan Moreno, and Huitzilin Ortiz. 2021. On the challenges of evaluating compositional explanations in multi-hop inference: Relevance, completeness, and expert ratings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7529–7542, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Jansen and Dmitry Ustalov. 2019. 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), pages 63–77, Hong Kong. Association for Computational Linguistics.

Peter Jansen, Elizabeth Wainwright, Steven Marmorstein, and Clayton Morrison. 2018. WorldTree: A corpus of explanation graphs for elementary science questions supporting multi-hop inference. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multi-hop question-answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 137–150, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3):535–547.

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

Henry Kautz. 2022. The third ai summer: Aaaai robert s. engelmore memorial lecture. AI Magazine, 43(1):93–104.

Henry Kautz and Bart Selman. 1999. Unifying sat-based and graph-based planning. In IJCAI, volume 99, pages 318–325.

Henry A Kautz, Bart Selman, et al. 1992. Planning as satisfiability. In ECAI, volume 92, pages 359–363. Citeseer.

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

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8082–8090.

Tushar Khot, Daniel Khashabi, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2021. 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, pages 1264–1279, Online. Association for Computational Linguistics.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. Scitail: A textual entailment dataset from science question answering. In AAAI.

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

Souvik Kundu, Tushar Khot, Ashish Sabharwal, and Peter Clark. 2019. Exploiting explicit paths for multi-hop reading comprehension. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2737–2747, Florence, Italy. Association for Computational Linguistics.

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

Kostas S Metaxiotis, Dimitris Askounis, and John Psarras. 2002. Expert systems in production
planning and scheduling: A state-of-the-art survey. *Journal of Intelligent Manufacturing*, 13(4):253–260.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6097–6109, Florence, Italy. Association for Computational Linguistics.

Mark A Musen and Johan Van der Lei. 1988. Of brittleness and bottlenecks: Challenges in the creation of pattern-recognition and expert-system models. In *Machine Intelligence and Pattern Recognition*, volume 7, pages 335–352. Elsevier.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPs*.

Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1824–1833, Hong Kong, China. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Gabriele Picco, Thanh Lam Hoang, Marco Luca Sbodio, and Vanessa Lopez. 2021. Neural unification for logic reasoning over natural language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3939–3950, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *ArXiv*, abs/2009.03393.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Nils Reimers and Iryna Gurevych. 2019. *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Alexandre Riazanov and Andrei Voronkov. 2002. The design and implementation of vampire. *AI Communications*, 15(2-3):91–110.

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. 2022. Entailment tree explanations via iterative retrieval-generation reasoner. In *NAACL-HLT*.

Tim Rocktäschel and Sebastian Riedel. 2017. *End-to-End Differentiable Proving*. *arXiv:1705.11040 [cs]*. ArXiv: 1705.11040.

Stuart Russell and Peter Norvig. 2010. *Artificial intelligence: a modern approach*.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2021. multiPRover: Generating multiple proofs for improved interpretability in rule reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3662–3677, Online. Association for Computational Linguistics.

Lenhart Schubert. 2015. *Semantic representation*. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1).

Maria I Sessa. 2002. Approximate reasoning by similarity-based sld resolution. *Theoretical computer science*, 275(1-2):389–426.
Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4615–4629, Online. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3621–3634, Online. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. arXiv preprint arXiv:2210.12217.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 641–651, New Orleans, Louisiana. Association for Computational Linguistics.

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Leap-of-thought: Teaching pre-trained models to systematically reason over implicit knowledge. Advances in Neural Information Processing Systems, 33:20227–20237.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2021. Explainable inference over grounding-abstract chains for science questions. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1–12, Online. Association for Computational Linguistics.

Marco Valentino, Mokanarangan Thayaparan, and André Freitas. 2022. Case-based abductive natural language inference. In Proceedings of the 29th International Conference on Computational Linguistics, pages 1556–1568, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Leon Weber, Pasquale Minervini, Jannes Münchmeyer, Ulf Leser, and Tim Rocktäschel. 2019. NLProlog: Reasoning with weak unification for question answering in natural language. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 6151–6161, Florence, Italy. Association for Computational Linguistics.

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

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

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. Transactions of the Association for Computational Linguistics, 8:183–198.

Zhengnan Xie, Sebastian Thiem, Jaycie Martin, Elizabeth Wainwright, Steven Marmorstein, and Peter Jansen. 2020. WorldTree v2: A corpus of science-domain structured explanations and inference patterns supporting multi-hop inference. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 5456–5473, Marseille, France. European Language Resources Association.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. Generating natural language proofs with verifier-guided search. arXiv preprint arXiv:2205.12443.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In Proceedings of
A Hyperparameters

The DRG produces 10 (pre-filter) candidates per template and 100 free-generated ones. The retrieval-conditioned DRG generates 100 $f_2$ candidates for the 10 top-scoring retrieved $f_1$'s. All stochastic sampling is nucleus sampling ($p = .95$). RTE models filter all candidates for which the classification softmax likelihood of entailment is less than 0.7. We remove duplicates before filtering. This configuration was found using Optuna (Akiba et al., 2019). Codebase will be released upon acceptance.

B QA2D Conversion

The following is an example conversion by our QA2D model of a multiple-choice question into a set of candidate hypotheses to prove.

**Q:** Ethanol is an alternative fuel made from corn. What is one of the unfavorable effects of using ethanol as a fuel?

(A) decreased cost of fuel production
(B) decrease in farm land available for food production
(C) increase in the consumption of fossil fuels
(D) increased carbon footprint from driving automobiles

Converted to the following hypotheses:

(A) Using ethanol has an unfavorable effect on the cost of fuel.
(B) Using ethanol as fuel decreases farm land available for food production.
(C) Using ethanol as fuel increases in the consumption of fossil fuels.
(D) The use of ethanol as an alternative fuel can lead to increased carbon footprints.

C WorldTree Templates

Figure 5 depicts the infilling templates gathered from WorldTree and used for template-guided rule generation. We manually annotated this list. The initial set of templates was of size 150; this was reduced to the 50 templates matching 10 or more nodes in the EntailmentBank train split.