Verifying the Steps of Deductive Reasoning Chains

Zacchary Sadeddine
Télécom Paris,
Institut Polytechnique de Paris
zacchary.sadeddine@telecom-paris.fr

Fabian M. Suchanek
Télécom Paris,
Institut Polytechnique de Paris
suchanek@telecom-paris.fr

Abstract

As Large Language Models penetrate everyday life more and more, it becomes essential to measure the correctness of their output. In this paper, we propose a novel task: the automatic verification of individual reasoning steps in a logical deductive Chain-of-Thought. This task addresses two well-known problems of LLMs, hallucination and incorrect reasoning. We propose a new dataset of logical reasoning chains, in which the individual deduction steps have been manually annotated for soundness, and benchmark several methods on it. We find that LLMs can detect unsound reasoning steps fairly well, but argue that verification has to be performed by transparent methods instead. We test symbolic methods, but find that they under-perform. We develop a neuro-symbolic baseline called VANESSA that comes closer to the performance of LLMs.

The question is how you arrive at your opinions and not what your opinions are.
– Bertrand Russell

1 Introduction

Large Language Models (LLMs) have worked miracles in natural language processing. Yet they still have difficulties with logical reasoning (Creswell et al., 2023; Huang and Chang, 2023; Helwe et al., 2021): given a text and a question that requires logical reasoning on that text, they may arrive at the wrong answer. The now commonly used technique of Chain-of-Thought (CoT) prompting (Wei et al., 2022) has been shown to improve the reasoning capabilities of the models. With this technique, the LLM does not produce a simple answer and possibly explanations, but a reasoning chain, where each step in the chain consists of premises and a conclusion. Even though these chains make the path from question to answer more transparent, they still suffer from the same issues: each individual reasoning step can rest on invalid (hallucinated) premises, or be simply logically incorrect. Previous work has focused nearly exclusively on verifying the final answer of the LLM – not the reasoning chain. This is problematic because a model may arrive at the correct answer for the wrong reasons (as in Figure 1). Indeed, Golovneva et al. (2023) have observed that even strong LLMs, such as GPT-3, produce a large amount of unsound reasoning steps. Examples are in Appendix D.

1. Context and question (given)

If a game is selected in the Top List, it was acclaimed by the critics or sold 1m units. The game "Legend of Zelda" is in the Top List and was not acclaimed by the critics. Did "Legend of Zelda" sell 1m units?

2. Reasoning Chain (produced by a LLM)

Premise 1.1: The game"Legend of Zelda" is in the Top List.
Premise 1.2: If a game is selected in the Top List, it was acclaimed by the critics or sold 1m units
Conclusion 1: The game "Legend of Zelda" was acclaimed by the critics.

Premise 2.1: The game "Legend of Zelda" was acclaimed by the critics.
Premise 1.2: If a game is acclaimed by the critics, it sold 1m units
Conclusion 2: 1m copies of Legend of Zelda were sold.

Figure 1: Illustration (adapted from Han et al., 2022) of an incorrect reasoning chain that delivers the correct answer (first red line: invalid conclusion; second red line: hallucinated premise). Real examples in Appendix D.

The question thus arises **how we can automatically detect unsound reasoning steps**. Previous work (Ling et al., 2024) has verified reasoning chains with GPT-3, with mixed results. In this paper, we propose a benchmark of logical deductive reasoning chains to evaluate such methods systematically. We have gathered several major publicly available logical reasoning datasets: ProofWriter (Tafjord et al., 2021), ProntoQA (Saparov and He, 2023), FOLIO (Han et al., 2022) and EntailmentBank (Dalvi et al., 2021). Where reasoning chains did not exist, we have produced them ourselves with different models, and we have manually evaluated each step in each chain by two criteria: *validity* (whether the conclusion follows indeed from the

Findings of the Association for Computational Linguistics: ACL 2025, pages 456–475
July 27 - August 1, 2025

premises) and *goundedness* (whether each premise is either mentioned in the text or in the conclusion of a previous step). This gives us a dataset of 1400 reasoning chains, for a total of more than 5000 annotated steps.

We then proceed to **benchmark the performance of a variety of state-of-the-art approaches on the task verifying deductive reasoning chains**. Unsurprisingly, we find that LLM-based methods perform fairly well. This, however, is of little comfort: it can only marginally increase trust in the validity of a reasoning chain if one black-box model tells us that the output of another black-box model is correct. Hence, we set out to study to what degree symbolic approaches can approximate the performance of purely neural methods. We find that existing symbolic methods do not offer the performance of black-box models.

To explore how far symbolic methods can go, we develop **VANESSA**[1]**, a symbolic baseline that combines syntactic analysis of sentences with a deductive reasoner**. VANESSA has two modes: In the fully symbolic mode, it achieves low recall but very high precision (i.e., it rarely validates an invalid reasoning step). In the neuro-symbolic mode, it uses a neural model for Natural Language Inference (NLI), i.e., to check whether one basic sentence logically entails another basic sentence. In contrast to more advanced tasks such as verifying logical steps entirely, or translating text to first order logic, NLI is a well-studied task, and modern approaches can solve it rather well on basic sentences. Using such a module allows VANESSA to perform on par with black-box methods. That said, the task of verifying deductive reasoning chains cannot be solved in perfection by any of the methods that we have studied, and it thus remains an open and challenging task.

All code and data is available at `https://github.com/dig-team/VANESSA`. Our paper is structured as follows: Section 2 discusses related work and Section 3 preliminaries; Section 4 presents our benchmark; Section 5 introduces the approaches we evaluate; Section 6 presents our neuro-symbolic VANESSA; and Section 7 showcases experiments, before Section 8 concludes.

[1]Verification of Answers by Natural deduction, Entailments, and Syntactic Sentence Analysis

2 Related Work

Deductive Reasoning with Language Models. RuleTaker (Clark et al., 2021) was the first work to show that Language Models can perform deductive reasoning on natural language text. After that, many reasoning-based question answering datasets have been proposed, each with different settings and different types of reasoning (Clark et al., 2018; Yu et al., 2020; Liu et al., 2020; Yang et al., 2018; Boratko et al., 2020; Hendrycks et al., 2021). Recently, the use of LLMs has greatly improved results on these tasks, notably through the use of Chain-of-Thought prompting (Wei et al., 2022). Several works aim to further improve their reasoning through the use of self-reflection and external tools such as theorem provers (Creswell et al., 2022; Olausson et al., 2023; Pan et al., 2023; Lyu et al., 2023). However, all of these works evaluate the final answer of the LLM, and not the steps of the reasoning chain. But LLMs may arrive at the right answer for the wrong reasons.

Benchmarks of Deductive Reasoning Chains. ProofWriter (Tafjord et al., 2021) was one of the earliest datasets of deductive reasoning chains. However, it contains only sound reasoning chains, which means that it cannot serve as a benchmark for models that verify reasoning chains. The same holds for ProntoQA (Saparov and He, 2023; Saparov et al., 2023) and EntailmentBank (Dalvi et al., 2021). In our work, we corrupt some of the reasoning steps from these datasets to obtain a balanced benchmark. More recently, ROSCOE (Golovneva et al., 2023) proposed a benchmark for reasoning chain evaluation, with step-level human annotations, which Jacovi et al. (2024) also did for open-domain question answering, with a focus on the factuality of statements. However, their datasets focus on arithmetic, discrete and commonsense reasoning, while we aim at purely logical deductive reasoning.

Verifying Deductive Reasoning Chains. ProofWriter (Tafjord et al., 2021) proposed a system that can validate each reasoning step. However, this method is restricted to a precise vocabulary and a limited set of reasoning patterns. ProntoQA (Saparov and He, 2023; Saparov et al., 2023) comes with a symbolic method to verify reasoning steps for their dataset, which parses the steps to first-order logic and then uses a formal solver. We use this method in our benchmark. ReCEval (Prasad et al., 2023), the work of Ling et al. (2024), and ROSCOE (Golovneva et al., 2023) also propose methods to evaluate reasoning chains. However, these works evaluate the reasoning chain as a whole, while our work evaluates every single step individually. Furthermore, they explore only the direct use of LLMs for this task, and not symbolic and neuro-symbolic methods, as we do. Most similar to ours, Jacovi et al. (2024) uses LLMs to verify individual steps. However, they primarily evaluate each statement against real-world knowledge, rather than the logical soundness of the steps.

3 Preliminaries

We adopt the formalization of Ling et al. (2024) for deductive reasoning-based question answering:

Definition. *A deductive reasoning-based question answering task is a tuple (C, Q, O, A) where C is the context, Q the question, O the set of answer options and $A \in O$ the ground-truth answer. All information needed to answer the question Q is present in the context C, and can be used to arrive at the correct answer A through deductive reasoning. Q, C and O are submitted to a model, which outputs an answer A'.*

We consider only the answer options Yes/No/Uncertain, considering that any multiple-choice question can formulated this way.

Definition. *A reasoning chain for a reasoning-based question answering task is a sequence of intermediate reasoning steps $S = (s_1, s_2, \ldots, s_m)$, where each reasoning step consists of one or more premises and a conclusion ($s_i = (p_i^1, \ldots, p_i^n, c_i)$) and the final conclusion c_m is the answer A' of the model.*

We are interested in verifying (1) the validity and (2) the groundedness of each step in a reasoning chain. A reasoning step is deductively *valid* if the conclusion follows logically from the premises (Ling et al., 2024). More formally, step s_i is valid if

$$p_i^1, \ldots, p_i^n \vdash c_i$$

A premise of a reasoning step is *grounded* if it comes from the context or from a previous conclusion. A reasoning step is grounded if all of its premises are grounded, i.e., step s_i is grounded if

$$\forall j : p_i^j \in (C \cup \{c_k\}_{k<i})$$

When a step is not grounded, it means that the model hallucinated and "invented" premises. A step that is both grounded and valid is *sound*. A deductive reasoning chain is sound (*correct* in the sense of Golovneva et al., 2023; Prasad et al., 2023; Ling et al., 2024) if all of its steps are sound.

4 Benchmark Creation

Our goal is to create a dataset of deductive reasoning chains whose steps are annotated for groundedness and validity. We build it from the following reasoning-based question answering sources:

ProofWriter (Tafjord et al., 2021) is a question answering dataset that contains proofs with intermediate steps. This dataset was generated synthetically using small ontologies, and hence contains short and simple sentences with a limited vocabulary, and reasoning patterns needed to deduce the answer rely mainly on Modus Ponens ($A,\ A \Rightarrow B \vdash B$). We used 150 instances from the Depth 5, Open World Assumption development set for our dataset. Since all reasoning chains are sound in the original dataset, we introduce three possible perturbations corresponding to errors actually made by LLMs: The first, *Negate*, negates a premise or conclusion by adding a "not" (simulating misinformation and incorrect deductions). The second, *Hallucination*, replaces a premise or the conclusion by a sentence that is irrelevant to the problem (simulating invention). The third, *Remove*, deletes a premise (simulating a conclusion without enough support). All permutations affect validity, and the first two also affect groundedness when applied to a premise.

ProntoQA (Saparov and He, 2023) is also a logical question answering dataset of contexts with intermediate reasoning steps. It was also generated using hierarchical ontologies, but uses more diverse and complex reasoning patterns than ProofWriter. We used the 50 first instances of the 4-hop Composed Random set from ProntoQA-OOD (Saparov et al., 2023), which has the particularity of using fictional words (e.g. "zumpuses"). Similarly to the original paper, we generated chains-of-thought for this dataset using several Large Language Models (Mixtral 8x7B, LLaMa2-70B, LLaMa3-80B). We then manually annotated the validity and groundedness of each step in these chains.

FOLIO (Han et al., 2022) is a reasoning-based question answering dataset containing a wide array of problems and reasoning patterns, based on first-order-logic. While the previous two datasets are restricted in terms of lexical and syntactic variety, FOLIO contains sentences with a large variation

Dataset	Chains-of-thought	Instances	Steps	Average steps	Average premises	Correct answers	Valid steps	Grounded steps
ProofWriter	Given+Negate	149	754	5.06	2.49	100%	53%	,68%
ProofWriter	Given+Remove	149	754	5.06	1.99	100%	49%	100%
ProofWriter	Given+Hallucination	149	754	5.06	2.49	81%	48%	76%
ProntoQA	By Mixtral	50	93	1.86	3.75	58%	69%	59%
ProntoQA	By LLaMa2	50	204	4.08	2.74	80%	41%	89%
ProntoQA	By LLaMa3	50	141	2.82	2.03	78%	70%	96%
FOLIO	By Mixtral	204	455	2.23	2.79	55%	60%	73%
FOLIO	By LLaMa2	204	653	3.20	1.91	64%	51%	84%
FOLIO	By LLaMa3	204	606	2.97	1.98	72%	67%	88%
EntailmentBank	Given+Negation	100	393	3.93	2.02	100%	47%	83%
EntailmentBank	Given+Hallucination	100	387	3.87	2.06	100%	52%	81%

Table 1: Our reasoning chain verification benchmark. There are 1409 instances with 5194 steps in total.

in formulations, words, and entities. It is based on real-life instances and examples. FOLIO does not provide any reasoning chains, and hence we generated chains for its development set with the same models we used for ProntoQA, and annotated these manually.

Finally, **EntailmentBank** (Dalvi et al., 2021) is a deductive reasoning dataset where the deductions rely on Natural Language Inference. It provides reasoning trees, which we convert to linear reasoning chains. Similarly to ProofWriter, all original steps are grounded and valid. Prasad et al. (2023) proposed a method to generate perturbations for EntailmentBank, but their perturbations affect only one intermediate conclusion in the whole chain (i.e., only validity). To evaluate both groundedness and validity, we apply Negation and Hallucination perturbations in the same way as for ProofWriter.

Annotations were performed by the authors, with the instruction of annotating whether the conclusion can be logically deduced from the information contained in the premises (for validity), and whether each premise appeared previously (for groundedness). Inter-annotator agreement was 98.6%. Table 1 shows the statistics of our benchmark. The Correct Answers column corresponds to the accuracy of the final answer on the QA task, while Valid and Grounded Steps indicate the proportion of steps that were annotated as valid and grounded, respectively. In general, LLaMa models generate deeper reasoning chains (containing more steps) than Mixtral, indicating a better ability to decompose their reasoning. This is also shown by the fact that Mixtral-generated reasoning steps contain more premises on average, which often means that some premises are not useful or that the step is very complex and could be further decomposed. The same goes for groundedness, where Mixtral lags behind the LLaMa models. Both Mixtral and LLaMa3 generate more valid reasoning steps than LLaMa2, suggesting that LLMs get better at reasoning over time. Overall, our benchmark contains a wide array of both synthetic and real data, and of grounded and ungrounded, valid and invalid, and complex and simple reasoning chains.

5 Methods

We evaluate different methods for the detection of unsound reasoning chains, which rely on LLMs to different extents.

NLI. As proposed by ReCEval (Prasad et al., 2023), the validity of a reasoning step can be verified by NLI: if the premises entail the conclusion, then the step is considered valid. We adapt this technique to verify also groundedness: if each premise is entailed by a sentence from the context or from a previous conclusion, the step is grounded. A variant of this scheme (which we call *Full Context*) asks whether the premise is entailed from the entire context, concatenated with all previous conclusions. We used three models for NLI: a state-of-the-art DeBERTa model fine-tuned for NLI (Laurer et al., 2024), a LLaMa3-8B-Instruct model instructed and few-shot prompted for the task, and GPT 3.5-Turbo. The latter generally performs worse than LLaMa3, and the results are in Appendix F. The prompt for the NLI task is in Appendix C.2.

First-Order Logic Transformation. Another way to verify a reasoning chain is to transform each step to first-order logic by help of a LLM, and then verify the conclusion using a formal theorem prover (Olausson et al., 2023; Pan et al., 2023). This method makes the reasoning part of the verification

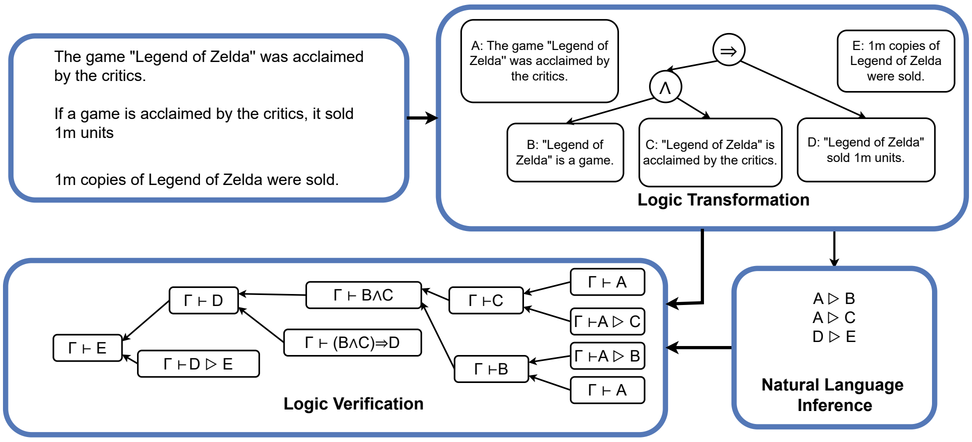

Figure 2: The VANESSA baseline for the (valid) step from Figure 1. We write $A \triangleright B$ for A entails B.

trustworthy and reliable, and reduces the task of the LLM from complete verification to first-order logic transformation. Groundedness can also be verified with a similar adaptation to that of NLI: when we want to verify the groundedness of a premise, we create a step with the examined premise as the conclusion, and a sentence from the previous context as premise. We use the LINC framework (Olausson et al., 2023) for our experiments. The original paper used GPT 3.5-turbo in its experiments, which we replaced with LLaMa3-8B-Instruct. The latter performs not only equivalently (see our ablation study in Appendix G), but is also lighter and easier to reproduce (and finance).

Fully Symbolic. ProntoQA (Saparov and He, 2023; Saparov et al., 2023) is a fully symbolic method to verify the soundness of a reasoning step, which parses the steps to first-order logic and then uses a formal solver. We also evaluate a naive baseline that assumes that every step in the chain is sound if the final answer is correct. We add another symbolic baseline, which says that a premise is grounded if it appears verbatim in the context or a previous conclusion.

VANESSA. To test how far symbolic or neuro-symbolic approaches can go, we develop a neuro-symbolic baseline, VANESSA. It also transforms the input to first-order logic and applies a reasoner. However, instead of using a LLM to transform sentences to first-order logic, it relies on a pattern-based transformation, and performs NLI between pairs of basic sentences.

6 VANESSA

Natural language exhibits variations: a step can be valid even if the conclusion is phrased differently from how it appears in the premises. In our example in Figure 1, Premise 1.2' states "If a game *is selected into* the Top List", while the conclusion is "LoZ *is* in the Top List". Purely symbolic methods have difficulties dealing with such variations. Purely neural methods, in contrast, are limited in their transparency. Hence, we set out to design VANESSA, a neuro-symbolic baseline that is more resilient to phrase variations. VANESSA translates the context and the reasoning steps to a logical form, and then uses a deductive solver. Our key idea is that, different from other solver-based methods, VANESSA does not convert a statement to a predicate with arguments. Rather, it keeps the statements as atomic literals, and reasons on them by NLI. In this way, VANESSA is more robust to phrase variations.

6.1 Transformation to Logic

We want to transform a given text (the context or a reasoning step) into a logical formula. To this end, we first apply co-reference resolution (with LingMess Otmazgin et al., 2023) to eliminate pronouns. We then transform every sentence into a logical form. We target the logical operators $\vee, \wedge, \Rightarrow, \forall$, as well as the exclusive disjunction $\oplus$. We first parse the sentence with a constituency parser (Kitaev et al., 2019), a well-established task with near-perfect performance. Then, we apply manually designed tree regular expressions (adapted from Graphene by Niklaus et al., 2016) recursively to transform the tree. For example, "Alex plays football and eats pasta" is transformed into `Alex_plays_football` $\wedge$ `Alex_eats_pasta`. Quantifiers such as "everyone" give rise to variables, i.e., "Everyone likes pasta" becomes $\forall x : x$_likes_pasta. We give the full list of tree patterns in Appendix A. In the case of "neither"/"nor", we negate the sentence parts with the rule-based tool of Anschütz et al. (2023). Finally, we eliminate all variables by instantiating them with all definite noun phrases from the input (i.e., the context or the reasoning step), creating one copy of the step per instantiation.

Note that our literals are atomic, and have no deeper syntactic structure: We produce `Alex_plays_football` and not `plays(Alex, football)`. The meaning of these atomic literals is exploited by the help of natural language inference, as follows: we apply an NLI model to pairs of sentences from the premises and the conclusion (chosen as described in Appendix B). If the NLI outputs an entailment between A and B, we add $A \Rightarrow B$ to our set of formulas (and $A \Rightarrow \neg B$ if it outputs a contradiction). For example, if our input step contains also `Alex_does_sports`, we would add `Alex_plays_football` $\Rightarrow$ `Alex_does_sports`. In this way, we completely bypass the need for a deep syntactic analysis of individual statements.

6.2 Natural Deduction

For a given reasoning step, we want to verify if the conclusion follows logically from the formulas given by the context, the premises, and the entailments. This is no easy feat. Two phenomena from classical logic in particular can be problematic in our context: material implication and the principle of explosion. *Material implication* is the equivalence $(A \Rightarrow B) \Leftrightarrow (\neg A \vee B)$. Such an equivalence does not correspond well to the "if... then..." statements in natural language. For instance, for any true fact B, $A \Rightarrow B$ will hold regardless of A, allowing deductions such as "If all pigs can fly, then the Earth revolves around the Sun". The *principle of explosion* is the idea that anything can be deduced from a contradiction: $A, \neg A \vdash B$ (for any A, B) – which does not correspond well to natural language semantics.

We avoid these phenomena by using Natural Deduction (Gentzen, 1935). This framework allows us to selectively modify the rules of reasoning. The complete set of rules we use is shown in Figure 3. We take inspiration from Church's weak deduction theorem (Batens, 1987): in order to prove $\Gamma \vdash A \Rightarrow B$, we shall prove that $\Gamma, A \vdash B$ and $\Gamma \nvdash B$. We implement this idea by adding "non-optional" premises in two rules (indicated in red in the figure). In these rules, the presence of non-optional premises means that at least one of these must be essential to the proof. The first of these rules can be read as: $\Gamma \vdash A \Rightarrow B$ can be deduced if $\Gamma, A \vdash B$ and $\Gamma \nvdash B$. This condition avoids the deduction of arbitrary implications from true facts. If it is not present, then from any $\Gamma \nvdash B$, one could write $\Gamma, A \vdash B$ and thus deduce $\Gamma \vdash A \Rightarrow B$. The second rule with non-optional premises can be read as: $\Gamma \vdash \neg A$ can be deduced if $\Gamma, A \vdash B$ and $\Gamma, A \vdash \neg B$, and $\Gamma \nvdash B$ or $\Gamma \nvdash \neg B$. This condition prevents the principle of explosion: in its absence, from a contradiction $\Gamma \vdash B$ and $\Gamma \vdash \neg B$, one can immediately write $\Gamma, A \vdash B$ and $\Gamma, A \vdash \neg B$ regardless of A, and thus deduce $\Gamma \vdash \neg A$.

These rules are a restriction of standard natural deduction, and since standard natural deduction is sound and complete, we have:

Proposition. *Our natural deduction with the rules from Figure 3 is not complete, but sound.*

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \text{ (F.)}$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash A \vee B} \text{ (B.)}$$

$$\frac{\Gamma \vdash A \qquad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \text{ (B.)}$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma, A \vdash C \qquad \Gamma, B \vdash C}{\Gamma \vdash C} \text{ (B.F.)}$$

$$\frac{\Gamma \vdash A \vee B \qquad \Gamma \vdash \neg A}{\Gamma \vdash B} \text{ (F.)}$$

$$\frac{\Gamma \vdash A \Rightarrow B \qquad \Gamma \vdash A}{\Gamma \vdash B} \text{ (F + B.F.)}$$

$$\frac{\Gamma, {\color{red}A} \vdash B}{\Gamma \vdash A \Rightarrow B} \text{ (B.F.)}$$

$$\frac{\Gamma, {\color{red}A} \vdash B \qquad \Gamma, {\color{red}A} \vdash \neg B}{\Gamma \vdash \neg A} \text{ (F.B.)}$$

Figure 3: Set of rules used for Natural Deduction, with "F" indicating forward rules and "B" indicating backward rules. Non-optional premises are shown in red.

To apply the rules efficiently, we built a Natural Deduction Solver (Gentzen, 1935) that performs a bidirectional search (Pollock, 1999) (Algorithm 1). It starts from known facts (the premises) and aims to reach an objective (the conclusion). At every step, rules can be applied to deduce new facts from known facts (forward), or from one objective to induce other objectives (backward). More complex rules instantiate new objectives based on known facts (backward-forward) and objectives (forward-backward).

Algorithm 1 Our Natural Deduction Algorithm

```
Input: premises, entailments, conclusion
interests ← [conclusion]
facts ← premises + entailments
prioQueue ← facts + interests
interestLinks ← []
while conclusion ∉ facts and prioQueue ≠ ∅ do
    current ← prioQueue.pop()
    if type(current) = Fact then
        for interestLink ∈ interestLinks do
            if facts.discharge(interestLink) then
                newFact ← instantiate(interestLink, facts)
                prioQueue ← newFact
            end if
        end for
        newFacts ← [f(fact) for f ∈ forwardRules]
        prioQueue += newFacts
        type(current) ← FBFact
        prioQueue += current
        if type(current.fact) = Implies then
            newInterest ← Interest(current.left)
            interests += newInterest
            prioQueue += newInterest
        end if
    else
        if type(current) = Interest then
            newInterests, newInterestLinks ← [b(interest) for b ∈ backwardRules]
            type(current) ← BFInterest
            prioQueue += current
        else if type(current) = BFInterest then
            newInterests, newInterestLinks ← [b(interest) for b ∈ backwardForwardRules]
        else if type(current) = FBFact then
            newInterests, newInterestLinks ← [b(interest) for b ∈ forwardBackwardRules]
        end if
        for newInterestLink ∈ newInterestLinks do
            if facts.discharge(interestLink) then
                newFact ← instantiate(interestLink, facts)
                prioQueue += newFact
            else
                interestLinks += newInterestLink
            end if
        end for
        for newInterest ∈ newInterests do
            prioQueue += newInterest
        end for
    end if
end while
return conclusion ∈ facts
```

Transparency. When VANESSA approves of a reasoning step, it can give a fully symbolic proof (Figure 2). It says from where each premise was deduced (to prove groundedness), and it shows the proof tree of the natural deduction (to prove validity). The only black-box remains the Natural Language Inference between two sentences. Such an entailment is, however, relatively easy to verify for a human. In our example, it amounts to verifying whether "LoZ is selected into the Top List" entails "LoZ is in the Top List". If the use of NLI is not desired, VANESSA can also run in a fully symbolic mode, where the entailment is performed through string matching, i.e. an entailment holds iff the two sentences are identical – giving a completely symbolic proof for the validity and groundedness of a reasoning step.

7 Experiments

We evaluate all methods from Section 5 on our benchmark dataset from Section 4. We evaluate VANESSA with 3 different NLI models (LLama3, DeBERTa, and Symbolic), as well as with a conventional solver ("CS") instead of our natural deduction solver. If a method gives an error during execution, we consider that the model deems the step unsound. We measure precision and recall for the tasks of identifying valid and grounded reasoning steps. Since false positives are more harmful in these tasks (as they mean greenlighting an unsound step), we also report the F0.5 metric. For a better overview of performance on the whole distribution, we also report Somers' D (Somers, 1962).

Validity Verification. Our results for step-level validity verification (Table 2) show that scores generally decrease as datasets become more complex: ProofWriter and ProntoQA use synthetic data, and all methods perform relatively well. FOLIO uses non-synthetic data and EntailmentBank even uses phrase variations, so that all methods degrade.

Overall, LINC is the best-performing method. Yet, interestingly, its error rate is by far the highest on the more complex EntailmentBank and FOLIO, which shows that translating non-synthetic sentences to first-order logic works well in synthetic context, but has limits in more real-life scenarios. While one can debate whether an LLM translation to first order logic can be trusted, LINC is conservative and achieves high precision.

On the other side of the spectrum, direct NLI methods usually have the highest recall of all methods, but are among the worst performers in the other metrics. This means that this method is too generous with considering steps as valid. However, it also outperforms all other methods on EntailmentBank, which is the dataset designed with NLI in mind (as its name indicates).

Finally, the symbolic methods generally perform worse than the neural methods. As expected, the baseline performs dismal, because a correct answer

Dataset + CoT	Verification Method	Error Rate	Precision	Recall	F0.5	D
Proof-Writer + Neg.	Answer Baseline	0%	$0.52_{\pm0.02}$	$0.75_{\pm0.02}$	$0.55_{\pm0.02}$	-0.04
	ProntoQA	0%	$0.46_{\pm0.03}$	$0.26_{\pm0.02}$	$0.4_{\pm0.03}$	-0.09
	VANESSA Symbolic	0%	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	5%	$0.99_{\pm0.0}$	$0.93_{\pm0.01}$	$0.98_{\pm0.01}$	0.93
	VANESSA DeBERTa	18%	$0.99_{\pm0.0}$	$0.75_{\pm0.02}$	$0.93_{\pm0.01}$	0.76
	VANESSA L3-CS	0%	$0.7_{\pm0.02}$	$1.0_{\pm0.0}$	$0.74_{\pm0.02}$	0.61
	VANESSA DB-CS	0%	$0.65_{\pm0.02}$	$0.98_{\pm0.01}$	$0.7_{\pm0.02}$	0.49
	LINC LLaMa3	0%	$0.99_{\pm0.0}$	$0.83_{\pm0.02}$	$0.95_{\pm0.01}$	0.84
	NLI LLaMa3	0%	$0.95_{\pm0.01}$	$0.88_{\pm0.02}$	$0.93_{\pm0.01}$	0.83
	NLI DeBERTa	0%	$0.88_{\pm0.03}$	$0.26_{\pm0.02}$	$0.6_{\pm0.03}$	0.31
Proof-Writer + Remove	Answer Baseline	0%	$0.5_{\pm0.02}$	$0.83_{\pm0.02}$	$0.55_{\pm0.02}$	0.04
	ProntoQA	0%	$0.69_{\pm0.03}$	$0.32_{\pm0.02}$	$0.56_{\pm0.03}$	0.22
	VANESSA Symbolic	0%	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	5%	$0.99_{\pm0.0}$	$0.93_{\pm0.01}$	$0.98_{\pm0.01}$	0.93
	VANESSA DeBERTa	18%	$0.99_{\pm0.0}$	$0.75_{\pm0.02}$	$0.93_{\pm0.01}$	0.76
	VANESSA L3-CS	0%	$0.62_{\pm0.02}$	$0.99_{\pm0.0}$	$0.67_{\pm0.02}$	0.51
	VANESSA DB-CS	0%	$0.66_{\pm0.02}$	$0.98_{\pm0.01}$	$0.7_{\pm0.02}$	0.56
	LINC LLaMa3	5%	$0.98_{\pm0.01}$	$0.84_{\pm0.02}$	$0.95_{\pm0.01}$	0.84
	NLI LLaMa3	0%	$0.78_{\pm0.02}$	$0.87_{\pm0.02}$	$0.79_{\pm0.02}$	0.64
	NLI DeBERTa	0%	$0.65_{\pm0.04}$	$0.26_{\pm0.02}$	$0.5_{\pm0.03}$	0.16
Proof-Writer + Hallu	Answer Baseline	0%	$0.49_{\pm0.02}$	$0.69_{\pm0.02}$	$0.52_{\pm0.02}$	0.02
	ProntoQA	0%	$0.52_{\pm0.04}$	$0.25_{\pm0.02}$	$0.43_{\pm0.03}$	0.04
	VANESSA Symbolic	0%	$0.91_{\pm0.01}$	$0.99_{\pm0.0}$	$0.92_{\pm0.01}$	0.91
	VANESSA LLaMa3	4%	$0.89_{\pm0.02}$	$0.93_{\pm0.01}$	$0.89_{\pm0.01}$	0.83
	VANESSA DeBERTa	21%	$0.89_{\pm0.02}$	$0.74_{\pm0.02}$	$0.85_{\pm0.02}$	0.67
	VANESSA L3-CS	0%	$0.64_{\pm0.02}$	$0.99_{\pm0.0}$	$0.68_{\pm0.02}$	0.54
	VANESSA DB-CS	0%	$0.64_{\pm0.02}$	$0.99_{\pm0.01}$	$0.69_{\pm0.02}$	0.54
	LINC LLaMa3	5%	$0.91_{\pm0.02}$	$0.82_{\pm0.02}$	$0.89_{\pm0.02}$	0.76
	NLI LLaMa3	0%	$0.75_{\pm0.02}$	$0.88_{\pm0.02}$	$0.77_{\pm0.02}$	0.62
	NLI DeBERTa	0%	$0.63_{\pm0.04}$	$0.29_{\pm0.02}$	$0.51_{\pm0.03}$	0.16
ProntoQA + Mixtral	Answer Baseline	0%	$0.77_{\pm0.06}$	$0.57_{\pm0.06}$	$0.71_{\pm0.06}$	0.22
	ProntoQA	0%	$0.92_{\pm0.04}$	$0.5_{\pm0.06}$	$0.77_{\pm0.06}$	0.45
	VANESSA Symbolic	1%	$0.95_{\pm0.02}$	$0.57_{\pm0.06}$	$0.82_{\pm0.05}$	0.55
	VANESSA LLaMa3	20%	$0.9_{\pm0.04}$	$0.77_{\pm0.05}$	$0.85_{\pm0.04}$	0.6
	VANESSA DeBERTa	78%	$0.69_{\pm0.11}$	$0.16_{\pm0.04}$	$0.43_{\pm0.09}$	0.05
	VANESSA L3-CS	13%	$0.75_{\pm0.05}$	$0.77_{\pm0.05}$	$0.74_{\pm0.05}$	0.27
	VANESSA DB-CS	14%	$0.68_{\pm0.05}$	$0.82_{\pm0.05}$	$0.69_{\pm0.05}$	-0.02
	LINC LLaMa3	9%	$0.94_{\pm0.03}$	$0.85_{\pm0.04}$	$0.9_{\pm0.03}$	0.77
	NLI LLaMa3	0%	$0.76_{\pm0.05}$	$0.93_{\pm0.03}$	$0.78_{\pm0.04}$	0.43
	NLI DeBERTa	0%	$0.73_{\pm0.07}$	$0.4_{\pm0.06}$	$0.62_{\pm0.07}$	0.11
ProntoQA + LLaMa2	Answer Baseline	0%	$0.46_{\pm0.04}$	$0.75_{\pm0.05}$	$0.5_{\pm0.04}$	0.15
	ProntoQA	0%	$0.91_{\pm0.04}$	$0.54_{\pm0.05}$	$0.78_{\pm0.05}$	0.6
	VANESSA Symbolic	1%	$0.93_{\pm0.03}$	$0.61_{\pm0.05}$	$0.83_{\pm0.04}$	0.67
	VANESSA LLaMa3	11%	$0.87_{\pm0.04}$	$0.74_{\pm0.05}$	$0.82_{\pm0.04}$	0.7
	VANESSA DeBERTa	76%	$0.46_{\pm0.14}$	$0.07_{\pm0.03}$	$0.25_{\pm0.07}$	0.02
	VANESSA L3-CS	9%	$0.55_{\pm0.05}$	$0.56_{\pm0.05}$	$0.55_{\pm0.05}$	0.25
	VANESSA DB-CS	9%	$0.38_{\pm0.04}$	$0.7_{\pm0.05}$	$0.42_{\pm0.04}$	-0.09
	LINC LLaMa3	3%	$0.86_{\pm0.04}$	$0.89_{\pm0.03}$	$0.85_{\pm0.03}$	0.81
	NLI LLaMa3	0%	$0.51_{\pm0.04}$	$0.97_{\pm0.02}$	$0.56_{\pm0.04}$	0.4
	NLI DeBERTa	0%	$0.55_{\pm0.06}$	$0.48_{\pm0.05}$	$0.54_{\pm0.05}$	0.22
ProntoQA + LLaMa3	Answer Baseline	0%	$0.76_{\pm0.04}$	$0.89_{\pm0.03}$	$0.78_{\pm0.03}$	0.32
	ProntoQA	0%	$0.94_{\pm0.03}$	$0.48_{\pm0.05}$	$0.77_{\pm0.05}$	0.44
	VANESSA Symbolic	1%	$0.97_{\pm0.01}$	$0.64_{\pm0.05}$	$0.87_{\pm0.03}$	0.59
	VANESSA LLaMa3	4%	$0.97_{\pm0.01}$	$0.92_{\pm0.02}$	$0.95_{\pm0.02}$	0.89
	VANESSA DeBERTa	85%	$0.72_{\pm0.12}$	$0.1_{\pm0.03}$	$0.33_{\pm0.08}$	0.06
	VANESSA L3-CS	0%	$0.82_{\pm0.04}$	$0.9_{\pm0.03}$	$0.82_{\pm0.03}$	0.52
	VANESSA DB-CS	0%	$0.71_{\pm0.04}$	$0.96_{\pm0.02}$	$0.74_{\pm0.03}$	0.13
	LINC LLaMa3	3%	$0.96_{\pm0.02}$	$0.97_{\pm0.01}$	$0.95_{\pm0.02}$	0.95
	NLI LLaMa3	0%	$0.74_{\pm0.04}$	$0.95_{\pm0.02}$	$0.77_{\pm0.03}$	0.33
	NLI DeBERTa	0%	$0.79_{\pm0.06}$	$0.36_{\pm0.05}$	$0.63_{\pm0.06}$	0.16
Entailment Bank + Negation	Baseline Answer	0%	$0.47_{\pm0.03}$	$0.87_{\pm0.02}$	$0.51_{\pm0.02}$	-0.03
	ProntoQA	0%	$0_{\pm0}$	$0.01_{\pm0.01}$	$0.04_{\pm0.02}$	0
	VANESSA Symbolic	0%	$0.78_{\pm0.11}$	$0.04_{\pm0.01}$	$0.18_{\pm0.05}$	0.17
	VANESSA LLaMa3	17%	$0.76_{\pm0.03}$	$0.69_{\pm0.03}$	$0.74_{\pm0.03}$	0.51
	VANESSA DeBERTa	20%	$0.76_{\pm0.04}$	$0.43_{\pm0.04}$	$0.65_{\pm0.04}$	0.4
	VANESSA L3-CS	1%	$0.47_{\pm0.03}$	$0.76_{\pm0.03}$	$0.51_{\pm0.03}$	-0.02
	VANESSA DB-CS	0%	$0.46_{\pm0.03}$	$0.71_{\pm0.03}$	$0.49_{\pm0.03}$	-0.02
	LINC LLaMa3	45%	$0.76_{\pm0.05}$	$0.26_{\pm0.03}$	$0.55_{\pm0.05}$	0.26
	NLI LLaMa3	0%	$0.67_{\pm0.03}$	$0.99_{\pm0.01}$	$0.71_{\pm0.03}$	0.51
	NLI DeBERTa	0%	$0.77_{\pm0.03}$	$0.87_{\pm0.02}$	$0.78_{\pm0.03}$	0.61
Entailment Bank + Hallucin.	Baseline Answer	0%	$0.53_{\pm0.03}$	$0.82_{\pm0.03}$	$0.57_{\pm0.03}$	-0.02
	ProntoQA	0%	$0_{\pm0}$	$0.01_{\pm0.0}$	$0.03_{\pm0.02}$	0
	VANESSA Symbolic	0%	$0.73_{\pm0.12}$	$0.04_{\pm0.01}$	$0.18_{\pm0.05}$	0.13
	VANESSA LLaMa3	4%	$0.65_{\pm0.03}$	$0.75_{\pm0.03}$	$0.67_{\pm0.03}$	0.29
	VANESSA DeBERTa	3%	$0.59_{\pm0.04}$	$0.46_{\pm0.03}$	$0.56_{\pm0.03}$	0.11
	VANESSA L3-CS	0%	$0.56_{\pm0.03}$	$0.82_{\pm0.03}$	$0.6_{\pm0.03}$	0.13
	VANESSA DB-CS	0%	$0.52_{\pm0.03}$	$0.71_{\pm0.03}$	$0.55_{\pm0.03}$	-0.02
	LINC LLaMa3	46%	$0.86_{\pm0.04}$	$0.27_{\pm0.03}$	$0.6_{\pm0.05}$	0.31
	NLI LLaMa3	0%	$0.6_{\pm0.03}$	$0.98_{\pm0.01}$	$0.65_{\pm0.02}$	0.3
	NLI DeBERTa	0%	$0.74_{\pm0.03}$	$0.87_{\pm0.02}$	$0.76_{\pm0.02}$	0.5
FOLIO + Mixtral	Baseline Answer	0%	$0.68_{\pm0.03}$	$0.58_{\pm0.03}$	$0.66_{\pm0.03}$	0.18
	ProntoQA	0%	$0.72_{\pm0.09}$	$0.06_{\pm0.01}$	$0.25_{\pm0.04}$	0.07
	VANESSA Symbolic	0%	$0.94_{\pm0.03}$	$0.16_{\pm0.02}$	$0.47_{\pm0.05}$	0.24
	VANESSA LLaMa3	11%	$0.79_{\pm0.03}$	$0.6_{\pm0.03}$	$0.74_{\pm0.03}$	0.37
	VANESSA DeBERTa	12%	$0.84_{\pm0.03}$	$0.52_{\pm0.03}$	$0.75_{\pm0.03}$	0.39
	VANESSA L3-CS	8%	$0.62_{\pm0.03}$	$0.7_{\pm0.03}$	$0.63_{\pm0.02}$	0.06
	VANESSA DB-CS	8%	$0.62_{\pm0.03}$	$0.66_{\pm0.03}$	$0.63_{\pm0.03}$	0.07
	LINC LLaMa3	30%	$0.83_{\pm0.03}$	$0.48_{\pm0.03}$	$0.72_{\pm0.03}$	0.35
	NLI LLaMa3	0%	$0.66_{\pm0.02}$	$0.88_{\pm0.02}$	$0.69_{\pm0.02}$	0.25
	NLI DeBERTa	0%	$0.74_{\pm0.03}$	$0.59_{\pm0.03}$	$0.7_{\pm0.03}$	0.28
FOLIO + LLaMa2	Baseline Answer	0%	$0.53_{\pm0.03}$	$0.55_{\pm0.03}$	$0.53_{\pm0.02}$	0.02
	ProntoQA	0%	$0.84_{\pm0.07}$	$0.06_{\pm0.01}$	$0.25_{\pm0.04}$	0.14
	VANESSA Symbolic	0%	$0.97_{\pm0.02}$	$0.24_{\pm0.02}$	$0.6_{\pm0.04}$	0.36
	VANESSA LLaMa3	5%	$0.74_{\pm0.02}$	$0.69_{\pm0.03}$	$0.73_{\pm0.02}$	0.44
	VANESSA DeBERTa	5%	$0.81_{\pm0.02}$	$0.6_{\pm0.03}$	$0.75_{\pm0.02}$	0.47
	VANESSA L3-CS	3%	$0.56_{\pm0.03}$	$0.59_{\pm0.03}$	$0.57_{\pm0.02}$	0.11
	VANESSA DB-CS	3%	$0.56_{\pm0.03}$	$0.51_{\pm0.03}$	$0.55_{\pm0.03}$	0.09
	LINC LLaMa3	25%	$0.85_{\pm0.03}$	$0.48_{\pm0.03}$	$0.73_{\pm0.03}$	0.43
	NLI LLaMa3	0%	$0.6_{\pm0.02}$	$0.95_{\pm0.01}$	$0.65_{\pm0.02}$	0.36
	NLI DeBERTa	0%	$0.72_{\pm0.02}$	$0.69_{\pm0.03}$	$0.71_{\pm0.02}$	0.41
FOLIO + LLaMa3	Baseline Answer	0%	$0.72_{\pm0.02}$	$0.69_{\pm0.02}$	$0.71_{\pm0.02}$	0.12
	ProntoQA	0%	$0.66_{\pm0.04}$	$0.2_{\pm0.02}$	$0.45_{\pm0.03}$	-0.01
	VANESSA Symbolic	0%	$0.97_{\pm0.01}$	$0.26_{\pm0.02}$	$0.62_{\pm0.03}$	0.31
	VANESSA LLaMa3	5%	$0.81_{\pm0.02}$	$0.64_{\pm0.02}$	$0.76_{\pm0.02}$	0.31
	VANESSA DeBERTa	9%	$0.86_{\pm0.02}$	$0.56_{\pm0.02}$	$0.77_{\pm0.02}$	0.35
	VANESSA L3-CS	1%	$0.69_{\pm0.02}$	$0.69_{\pm0.02}$	$0.69_{\pm0.02}$	0.06
	VANESSA DB-CS	1%	$0.72_{\pm0.02}$	$0.67_{\pm0.02}$	$0.71_{\pm0.02}$	0.13
	LINC LLaMa3	24%	$0.87_{\pm0.02}$	$0.55_{\pm0.02}$	$0.78_{\pm0.02}$	0.37
	NLI LLaMa3	0%	$0.74_{\pm0.02}$	$0.89_{\pm0.02}$	$0.77_{\pm0.02}$	0.31
	NLI DeBERTa	0%	$0.77_{\pm0.02}$	$0.62_{\pm0.02}$	$0.73_{\pm0.02}$	0.22

VANESSA-DB-CS: VANESSA with DeBERTa and conventional solver
VANESSA-L3-CS: VANESSA with LLaMa3 and conventional solver.

Table 2: Performance for validity verification with different methods. Best performer on F0.5 and Somers' D is highlighted in green, second best in blue, and third best in red. Horizontal lines separate symbolic, neuro-symbolic and neural methods.

does not mean a correct reasoning. ProntoQA performs well on ProntoQA, but shows weak results on the other datasets, with low recall, exemplifying the general problem of symbolic approaches. The symbolic VANESSA has a better performance on all metrics than ProntoQA, even on the ProntoQA dataset, making it the best symbolic method. However, recall is still limited.

When VANESSA is combined with NLI models, in contrast, the performance increases so that the method joins the ranks of the best-performing black-box models. VANESSA combined with LLaMa3 is consistently among the three best methods in F0.5 and Somers' D over the benchmark, and its performance does not decrease as much as the other methods on EntailmentBank. Our Natural Deduction Solver systematically has better scores than the conventional one, indicating that it overcomes issues with material implication and the principle of explosion that the conventional solver faces.

Groundedness Verification. Overall, all methods perform better at groundedness (Table 3) than at validity. As before, the direct NLI methods usually have high recall but low precision, while LINC and symbolic methods have lower recall but better precision. For the direct NLI methods, there is no clear winner between the sentence-by-sentence and full context strategies. Overall, the best performing

Dataset +CoT	Verification Method	Precision	Recall	F0.5	D
Proof-Writer + Neg.	Symbolic NLI	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA Symbolic	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA DeBERTa	$0.96_{\pm0.01}$	$1.0_{\pm0.0}$	$0.96_{\pm0.01}$	0.94
	LINC LLaMa3	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	1.00
	NLI LLaMa3	$0.93_{\pm0.01}$	$1.0_{\pm0.0}$	$0.94_{\pm0.01}$	0.88
	NLI DeBERTa	$0.85_{\pm0.01}$	$1.0_{\pm0.0}$	$0.87_{\pm0.01}$	0.72
	NLI LLaMa3 FC	$0.95_{\pm0.01}$	$1.0_{\pm0.0}$	$0.96_{\pm0.01}$	0.92
	NLI DeBERTa FC	$0.99_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
Proof-Writer + Hallu.	Symbolic NLI	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA Symbolic	$0.99_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	$0.98_{\pm0.01}$	$1.0_{\pm0.0}$	$0.98_{\pm0.0}$	0.97
	VANESSA DeBERTa	$0.98_{\pm0.01}$	$1.0_{\pm0.0}$	$0.98_{\pm0.0}$	0.97
	LINC LLaMa3	$0.97_{\pm0.01}$	$1.0_{\pm0.0}$	$0.98_{\pm0.01}$	0.95
	NLI LLaMa3	$0.96_{\pm0.01}$	$1.0_{\pm0.0}$	$0.97_{\pm0.01}$	0.92
	NLI DeBERTa	$0.97_{\pm0.01}$	$1.0_{\pm0.0}$	$0.98_{\pm0.01}$	0.95
	NLI LLaMa3 FC	$0.91_{\pm0.01}$	$1.0_{\pm0.0}$	$0.92_{\pm0.01}$	0.79
	NLI DeBERTa FC	$0.97_{\pm0.01}$	$1.0_{\pm0.0}$	$0.97_{\pm0.01}$	0.94
ProntoQA + Mixtral	Symbolic NLI	$0.91_{\pm0.04}$	$0.34_{\pm0.06}$	$0.66_{\pm0.08}$	0.41
	VANESSA Symbolic	$0.96_{\pm0.02}$	$0.7_{\pm0.06}$	$0.86_{\pm0.04}$	0.71
	VANESSA LLaMa3	$0.89_{\pm0.04}$	$0.71_{\pm0.06}$	$0.83_{\pm0.05}$	0.64
	VANESSA DeBERTa	$0.84_{\pm0.06}$	$0.44_{\pm0.06}$	$0.69_{\pm0.07}$	0.39
	LINC LLaMa3	$0.91_{\pm0.04}$	$0.34_{\pm0.06}$	$0.66_{\pm0.08}$	0.41
	NLI LLaMa3	$0.61_{\pm0.05}$	$0.93_{\pm0.03}$	$0.65_{\pm0.05}$	0.18
	NLI DeBERTa	$0.82_{\pm0.05}$	$0.78_{\pm0.05}$	$0.80_{\pm0.05}$	0.58
	NLI LLaMa3 FC	$0.68_{\pm0.05}$	$0.85_{\pm0.05}$	$0.70_{\pm0.05}$	0.33
	NLI DeBERTa FC	$0.83_{\pm0.05}$	$0.64_{\pm0.06}$	$0.76_{\pm0.05}$	0.49
ProntoQA + LLaMa2	Symbolic NLI	$0.99_{\pm0.01}$	$0.87_{\pm0.02}$	$0.95_{\pm0.01}$	0.65
	VANESSA Symbolic	$0.99_{\pm0.01}$	$0.98_{\pm0.01}$	$0.98_{\pm0.01}$	0.95
	VANESSA LLaMa3	$0.98_{\pm0.01}$	$0.97_{\pm0.01}$	$0.97_{\pm0.01}$	0.90
	VANESSA DeBERTa	$0.95_{\pm0.01}$	$0.9_{\pm0.02}$	$0.94_{\pm0.02}$	0.53
	LINC LLaMa3	$0.99_{\pm0.01}$	$0.87_{\pm0.02}$	$0.95_{\pm0.01}$	0.65
	NLI LLaMa3	$0.9_{\pm0.02}$	$0.98_{\pm0.01}$	$0.91_{\pm0.02}$	0.29
	NLI DeBERTa	$0.94_{\pm0.02}$	$0.98_{\pm0.01}$	$0.94_{\pm0.01}$	0.72
	NLI LLaMa3 FC	$0.93_{\pm0.02}$	$0.96_{\pm0.01}$	$0.93_{\pm0.02}$	0.55
	NLI DeBERTa FC	$0.99_{\pm0.01}$	$0.94_{\pm0.02}$	$0.97_{\pm0.01}$	0.81
ProntoQA + LLaMa3	Symbolic NLI	$0.98_{\pm0.01}$	$0.76_{\pm0.04}$	$0.92_{\pm0.02}$	0.35
	VANESSA Symbolic	$0.97_{\pm0.01}$	$0.93_{\pm0.02}$	$0.95_{\pm0.02}$	0.44
	VANESSA LLaMa3	$0.99_{\pm0.01}$	$0.91_{\pm0.02}$	$0.96_{\pm0.01}$	0.59
	VANESSA DeBERTa	$0.97_{\pm0.01}$	$0.83_{\pm0.03}$	$0.92_{\pm0.02}$	0.26
	LINC LLaMa3	$0.98_{\pm0.01}$	$0.76_{\pm0.04}$	$0.92_{\pm0.02}$	0.35
	NLI LLaMa3	$0.94_{\pm0.02}$	$0.98_{\pm0.01}$	$0.94_{\pm0.02}$	-0.02
	NLI DeBERTa	$0.98_{\pm0.01}$	$0.94_{\pm0.02}$	$0.96_{\pm0.01}$	0.57
	NLI LLaMa3 FC	$0.95_{\pm0.02}$	$0.96_{\pm0.01}$	$0.94_{\pm0.02}$	0.18
	NLI DeBERTa FC	$0.98_{\pm0.01}$	$0.88_{\pm0.03}$	$0.95_{\pm0.02}$	0.50
FOLIO + Mixtral	Symbolic NLI	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA Symbolic	$1.0_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	$0.96_{\pm0.01}$	$1.0_{\pm0.0}$	$0.97_{\pm0.01}$	0.95
	VANESSA DeBERTa	$0.86_{\pm0.01}$	$1.0_{\pm0.0}$	$0.88_{\pm0.01}$	0.75
	LINC LLaMa3	$0.99_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	0.99
	NLI LLaMa3	$0.93_{\pm0.01}$	$1.0_{\pm0.0}$	$0.94_{\pm0.01}$	0.88
	NLI DeBERTa	$0.85_{\pm0.01}$	$1.0_{\pm0.0}$	$0.87_{\pm0.01}$	0.72
	NLI LLaMa3 FC	$0.95_{\pm0.01}$	$1.0_{\pm0.0}$	$0.96_{\pm0.01}$	0.92
	NLI DeBERTa FC	$0.99_{\pm0.0}$	$1.0_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
FOLIO + LLaMa2	Symbolic NLI	$0.99_{\pm0.0}$	$0.77_{\pm0.02}$	$0.93_{\pm0.01}$	0.58
	VANESSA Symbolic	$1.00_{\pm0.0}$	$0.77_{\pm0.02}$	$0.94_{\pm0.01}$	0.59
	VANESSA LLaMa3	$0.94_{\pm0.01}$	$0.95_{\pm0.01}$	$0.94_{\pm0.01}$	0.66
	VANESSA DeBERTa	$0.92_{\pm0.01}$	$0.97_{\pm0.01}$	$0.93_{\pm0.01}$	0.74
	LINC LLaMa3	$0.99_{\pm0.0}$	$0.77_{\pm0.02}$	$0.93_{\pm0.01}$	0.58
	NLI LLaMa3	$0.89_{\pm0.01}$	$0.99_{\pm0.0}$	$0.91_{\pm0.01}$	0.57
	NLI DeBERTa	$0.96_{\pm0.01}$	$0.96_{\pm0.01}$	$0.96_{\pm0.01}$	0.78
	NLI LLaMa3 FC	$0.9_{\pm0.01}$	$0.95_{\pm0.01}$	$0.91_{\pm0.01}$	0.46
	NLI DeBERTa FC	$0.92_{\pm0.01}$	$0.97_{\pm0.01}$	$0.93_{\pm0.01}$	0.64
FOLIO + LLaMa3	Symbolic NLI	$1.00_{\pm0.0}$	$0.88_{\pm0.01}$	$0.97_{\pm0.01}$	0.61
	VANESSA Symbolic	$1.00_{\pm0.0}$	$0.9_{\pm0.01}$	$0.97_{\pm0.01}$	0.72
	VANESSA LLaMa3	$0.98_{\pm0.01}$	$0.97_{\pm0.01}$	$0.98_{\pm0.01}$	0.84
	VANESSA DeBERTa	$0.99_{\pm0.0}$	$0.98_{\pm0.01}$	$0.98_{\pm0.0}$	0.91
	LINC LLaMa3	$1.00_{\pm0.0}$	$0.83_{\pm0.02}$	$0.95_{\pm0.01}$	0.61
	NLI LLaMa3	$0.92_{\pm0.01}$	$1.00_{\pm0.0}$	$0.93_{\pm0.01}$	0.62
	NLI DeBERTa	$0.98_{\pm0.01}$	$0.98_{\pm0.01}$	$0.97_{\pm0.01}$	0.83
	NLI LLaMa3 FC	$0.92_{\pm0.01}$	$0.88_{\pm0.01}$	$0.91_{\pm0.01}$	0.31
	NLI DeBERTa FC	$0.95_{\pm0.01}$	$0.97_{\pm0.01}$	$0.95_{\pm0.01}$	0.66
Entail-ment Bank + Neg.	Symbolic NLI	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA Symbolic	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	0.99
	VANESSA LLaMa3	$0.98_{\pm0.01}$	$0.99_{\pm0.0}$	$0.97_{\pm0.01}$	0.94
	VANESSA DeBERTa	$0.99_{\pm0.01}$	$0.99_{\pm0.0}$	$0.98_{\pm0.01}$	0.96
	LINC LLaMa3	$0.95_{\pm0.01}$	$0.99_{\pm0.0}$	$0.96_{\pm0.01}$	0.87
	NLI LLaMa3	$0.97_{\pm0.01}$	$0.99_{\pm0.0}$	$0.97_{\pm0.01}$	0.92
	NLI DeBERTa	$0.98_{\pm0.01}$	$0.99_{\pm0.0}$	$0.98_{\pm0.01}$	0.95
	NLI LLaMa3 FC	$0.99_{\pm0.01}$	$0.95_{\pm0.01}$	$0.97_{\pm0.01}$	0.86
	NLI DeBERTa FC	$0.99_{\pm0.0}$	$0.93_{\pm0.01}$	$0.97_{\pm0.01}$	0.83
Entail-ment Bank + Hallu	Symbolic NLI	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA Symbolic	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	$0.99_{\pm0.0}$	1.0
	VANESSA LLaMa3	$0.91_{\pm0.01}$	$0.99_{\pm0.0}$	$0.92_{\pm0.01}$	0.75
	VANESSA DeBERTa	$0.91_{\pm0.02}$	$0.99_{\pm0.0}$	$0.92_{\pm0.01}$	0.72
	LINC LLaMa3	$0.97_{\pm0.01}$	$0.99_{\pm0.0}$	$0.97_{\pm0.01}$	0.94
	NLI LLaMa3	$0.85_{\pm0.02}$	$0.99_{\pm0.0}$	$0.87_{\pm0.02}$	0.47
	NLI DeBERTa	$0.89_{\pm0.02}$	$0.99_{\pm0.0}$	$0.9_{\pm0.01}$	0.65
	NLI LLaMa3 FC	$0.87_{\pm0.02}$	$0.99_{\pm0.0}$	$0.89_{\pm0.01}$	0.55
	NLI DeBERTa FC	$0.83_{\pm0.02}$	$0.99_{\pm0.0}$	$0.86_{\pm0.02}$	0.35

Table 3: Performance for groundedness verifications with different methods. FC = Full Context. Best performer is highlighted in green, second best in blue, and third best in red.

method is VANESSA with LLaMa3.

Error Analysis. We manually examined the false positives for validity on the FOLIO reasoning chains (Table 8 in Appendix E). For VANESSA, the most important source of error are faulty entailments, which cause up to 100% of false positives. Indeed, an incorrect entailment can easily lead to an invalid conclusion. As an example, the step "Premise 1: All professional tennis players are athletes. Premise 2: Djokovic is an athlete. Conclusion: Djokovic is a professional tennis player." was deemed valid by VANESSA because of an entailment between Premise 2 and the Conclusion. This entailment (which doesn't stand) is likely caused by the fact that, in the real world, Djokovic is a professional tennis player, which the LLM might have learned during training. More examples and statistics are in Appendix E.

8 Conclusion

We have presented a benchmark for the verification of individual reasoning steps in a deductive chain-of-thought. We find that neural methods perform best, but lack transparency, while symbolic methods struggle with recall.

To mitigate these shortcomings, we have introduced VANESSA, a neuro-symbolic method that uses a fully symbolic natural deduction solver, and relies on natural language inference to exploit sentence semantics. Our experiments show that VANESSA can rival the performance of neural methods while being more transparent.

However, in general, verifying reasoning chains cannot yet be done in perfection, by any approach, and it thus remains an exciting avenue of research.

9 Limitations

While all methods can verify validity and groundedness with varying precision, none of the methods has a precision of 100% on real-world datasets. Our work can thus be seen only as a benchmarking of existing methods, and as a call to further improve these methods, and not as an approach that could certify the soundness of reasoning chains with the quality that would be needed in high-stake applications.

Acknowledgements

This work was partially funded by the NoRDF project (ANR-20-CHIA-0012-01).

References

Miriam Anschütz, Diego Miguel Lozano, and Georg Groh. 2023. This is not correct! negation-aware evaluation of language generation systems. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 163–175, Prague, Czechia. Association for Computational Linguistics.

Diderik Batens. 1987. Relevant implication and the weak deduction theorem. *Studia Logica: An International Journal for Symbolic Logic*, 46(3):239–245.

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136, Online. Association for Computational Linguistics.

Matthias Cetto, Christina Niklaus, André Freitas, and Siegfried Handschuh. 2018. Graphene: Semantically-linked propositions in open information extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2300–2311. Association for Computational Linguistics.

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.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2023. Selection-inference: Exploiting large language models for interpretable logical reasoning. In *The Eleventh International Conference on Learning Representations*.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021. Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Gerhard Gentzen. 1935. Untersuchungen Über das Logische Schließen. I. *Mathematische Zeitschrift*, 35:176–210.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. 2022. Folio: Natural language reasoning with first-order logic.

Chadi Helwe, Chloé Clavel, and Fabian M. Suchanek. 2021. Reasoning with Transformer-based Models: Deep Learning, but Shallow Reasoning. In *AKBC*.

Chadi Helwe, Simon Coumes, Chloé Clavel, and Fabian M. Suchanek. 2022. TINA: Textual Inference with Negation Augmentation. In *EMNLP Find*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards reasoning in large language models: A survey.

Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roee Aharoni, and Mor Geva. 2024. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4615–4634, Bangkok, Thailand. Association for Computational Linguistics.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Moritz Laurer, Wouter van Atteveldt, Andreu Casas, and Kasper Welbers. 2024. Less Annotating, More Classifying: Addressing the Data Scarcity Issue of

Supervised Machine Learning with Deep Transfer Learning and BERT-NLI. *Political Analysis*, 32(1).

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2024. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3622–3628. International Joint Conferences on Artificial Intelligence Organization. Main track.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329, Nusa Dua, Bali. Association for Computational Linguistics.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Theo Olausson, Alex Gu, Ben Lipkin, Cedegao Zhang, Armando Solar-Lezama, Joshua Tenenbaum, and Roger Levy. 2023. 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*, pages 5153–5176, Singapore. Association for Computational Linguistics.

Shon Otmazgin, Arie Cattan, and Yoav Goldberg. 2023. LingMess: Linguistically informed multi expert scorers for coreference resolution. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2752–2760, Dubrovnik, Croatia. Association for Computational Linguistics.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

John Pollock. 1999. Natural deduction. *an unpublished manuscript is available at http://johnpollock.us/ftp/OSCAR-web-page/PAPERS/Natural-Deduction. pdf.*

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Robert H. Somers. 1962. A new asymmetric measure of association for ordinal variables. *American Sociological Review*, 27(6):799–811.

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.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Weihao Yu, Zihang Jiang, Yanfei Dong, and Jiashi Feng. 2020. Reclor: A reading comprehension dataset requiring logical reasoning. In *International Conference on Learning Representations*.

A Transformation Patterns

Following Cetto et al. (2018), we detect tree regular expression patterns in constituency trees and apply transformations to the tree. The full list of patterns is shown in Table 4.

B Pairing Strategies for NLI

After logic transformation and instantiation, VANESSA uses Natural Language Inference (NLI) to find relations between atomic statements. As in Helwe et al. (2022), we write $A \triangleright B$ for A entails B, and $A \blacktriangleright B$ for A contradicts B.

Testing all combinations of atomic statements in a reasoning step would both be computationally expensive and increase the error chance (as NLI is not 100% reliable). Another problem that appears when doing so is that implications can be "shortcut" by entailments: if we have two premises $A \Rightarrow B$ and C, and if the NLI model predicts $C \triangleright B$, then the solver can prove B without first proving A (which should not be possible in a rigorous reasoning). For these reasons, we designed a strategy for choosing the statements to pair that avoids these situations but still covers all interesting cases. To do so, we divide the statements in one reasoning step into two sets: left and right, depending on which side they are of an implication. Statements that appear in formulas that don't contain any implication are considered to be in both left and right sets. We also make a difference between statements from the premises and those from the conclusion. We then run NLI on all elements from specific pairs of categories and keep the predictions that are interesting based on the pairing. To illustrate this, let us take a case where the premises are $A \Rightarrow B$ and $C \Rightarrow D$, and the conclusion is $E \Rightarrow F$. Several sets of entailments / contradictions could make the conclusion derive from the premises (we'll not consider symmetries):

- $E \triangleright A$ and $B \triangleright C$ and $D \triangleright F$
- $E \triangleright A$ and $B \blacktriangleright D$ and $\neg C \triangleright F$ (Contraposition)
- $E \blacktriangleright B$ and $\neg A \triangleright C$ and $D \triangleright F$
- $E \blacktriangleright B$ and $\neg A \blacktriangleright D$ (i.e. $D \triangleright A$) and $\neg C \triangleright F$
- $\neg F \triangleright A$ and $B \triangleright C$ and $D \triangleright F$
- $\neg F \triangleright A$ and $B \blacktriangleright D$ and $\neg C \triangleright F$
- $\neg F \blacktriangleright B$ (i.e. $B \triangleright F$) and $\neg A \triangleright C$ and $D \triangleright F$
- $\neg F \blacktriangleright B$ and $B \blacktriangleright D$ and $\neg C \triangleright F$

Based on this, we can determine all the pairs of statements that we have to perform NLI on:

- Right Premise $\triangleright$ Left Premise
- Left Conclusion $\triangleright$ Left Premise
- Right Premise $\triangleright$ Right Conclusion
- Right Premise $\blacktriangleright$ Right Premise
- Left Conclusion $\blacktriangleright$ Right Premise
- $\neg$ Left Premise $\triangleright$ Left Premise
- $\neg$ Right Conclusion $\triangleright$ Left Premise

In the last two cases, we need to negate the NLI premise, in which case we add "It is not true that" before the sentence.

Coordinate Clauses

`ROOT «: (S < (S ?$.. CC & $.. S))`

Create a new node whose label depends on CC ("and", "or", neitherNor", "nor"), with children the original ROOT node where S is replaced by one of its **S** children

Non-restrictive relative clauses commencing with a preposition followed by a relative pronoun

`ROOT «: (S « (NP <, NP & < (/,/ $+ (SBAR <, (WHPP $+ S & <, IN & <- WHNP) & ?$+ /,/))))`

Create a new "AND" node with two children. One is the ROOT node where NP's right siblings are removed. The other is ROOT(**S** + PP(IN, NP))

Non-restrictive Relative Clauses commencing with the relative pronoun "where"

`ROOT «: (S « (/.*/ < (NP|PP $+ (/,/ $+ (SBAR <, (WHADVP $+ S & «: WRB) & ?$+ /,/)))))`

Create a new "AND" node with two children. One is the ROOT node where NP|PP's right siblings are removed. The other is ROOT(**S** + PP(IN, NP|PP))

Non-restrictive Relative Clauses commencing with the relative pronoun "whom"

`ROOT «: (S « (NP <, NP & < (/,/ $+ (SBAR <, (WHNP $+ (S <, NP & <- VP) & «: (WP <: whom)) & ?$+ /,/))))`

Create a new "AND" node with two children. One is the ROOT node where NP's right siblings are removed. The other is ROOT(**S**) where NP has been inserted as subject

Non-restrictive Relative Clauses commencing with the relative pronoun "whose"

`ROOT «: (S « (NP < (NP $+ (/,/ $+ (SBAR <, (WHNP $+ S & <, (WP$)) & ?$+ /,/)))))`

Create a new "AND" node with two children. One is the ROOT node where NP's right siblings are removed. The other is ROOT(**S**) where NP+"'s" has been inserted in the subject

Non-restrictive Relative Clauses commencing with the relative pronoun "who/which"

`ROOT «: (S « (NP <, NP & < (/,/ $+ (SBAR <, (WHNP $+ S & «: WP|WDT) & ?$+ /,/))))`

Create a new "AND" node with two children. One is the ROOT node where NP|PP's right siblings are removed. The other is ROOT(**S**) where NP has been inserted as subject

Preposed Adverbial Clauses

`ROOT «: (S < (SBAR < (S < (NP $.. VP)) $.. (NP $.. VP)))`

Create a new node whose label depends on SBAR (Useful for implications indicated by "if", most often "AND") with two children. The first one is is ROOT(**S**). The second is the ROOT node where SBAR and its left siblings are removed

Explicit Universal Quantification

`ROOT «: (S « (NP « marker !$,, RB)) & !» If & !» Universal`

markers = {every, everyone, everything, everybody, everywhere, all, each, any, anyone, anything, anybody, anywhere, people}

Create a new node Universal with a unique If child, which has two children itself. The first corresponds to (ROOT("X is" + NP)). The second is ROOT, where NP is replaced by NNP(/X/). If NP was plural, it gets singularized.

Negative Explicit Universal Quantification

`ROOT «: (S « (NP « marker)) & !» If & !» Universal`

markers = {no, nobody, nothing, nowhere}

Create a new node Universal with a unique If child, which has two children itself. The first corresponds to (ROOT("X is" + NP)). The second is Not(ROOT), where NP is replaced by NNP(/X/). If NP was plural, it gets singularized.

Implicit Universal Quantification with plural subject

`ROOT «: (S < (NP « (NNP|NNPS !» VP) & $.. VP)) & !» If & !» Universal`

NP is an indefinite plural NP. VP's tense is simple present

Create a new node Universal with a unique If child, which has two children itself. The first corresponds to (ROOT("X is" + NP)). The second is ROOT, where NP is replaced by NNP(/X/). If NP was plural, it gets singularized.

Implicit Universal Quantification with indefinites and be

ROOT «: (S < ($\boxed{\text{NP}}$ $.. (VP « /is/ & < $\boxed{\text{NP}}$))) & !» If & !» Universal

$\boxed{\text{NP}}$ are indefinite singular NP

Create a new node Universal with a unique **If** child, which has two children itself. The first corresponds to (ROOT("X is" + $\boxed{\text{NP}}$)). The second is Not(ROOT), where $\boxed{\text{NP}}$ is replaced by NNP(/X/).

Explicit Universal Quantification embedded in an implication

ROOT «: (S « ($\boxed{\text{NP}}$ « marker !$,, RB)) & »,1 **If** & !» Universal

Create a new node Universal whose child is **If**. Transform its left child to an "AND" node, whose left child corresponds to (ROOT("X is" + $\boxed{\text{NP}}$)) and right child is ROOT, where $\boxed{\text{NP}}$ is replaced by NNP(/X/). Replace occurences of $\boxed{\text{NP}}$ in **If**'s descendants by NNP(/X/).

Implicit Universal Quantification embedded in an implication

ROOT «: (S « ($\boxed{\text{NP}}$!» VP)) & »,1 **If** & !» Universal

$\boxed{\text{NP}}$ is an indefinite singulat NP

Create a new node Universal whose child is **If**. Transform its left child to an "AND" node, whose left child corresponds to (ROOT("X is" + $\boxed{\text{NP}}$)) and right child is ROOT, where $\boxed{\text{NP}}$ is replaced by NNP(/X/). Replace occurences of $\boxed{\text{NP}}$ in **If**'s descendants by NNP(/X/).

Non-restrictive Preposed Participial Phrases

ROOT «: (S [< ($\boxed{\text{VP}}$ «, (VBG|VBN)) | < ($\boxed{\text{PP|ADVP}}$ <+PP|ADVP (S <: (VP «, VBG|VBN)))] & $.. (**NP** $.. **VP**))

Create a new "AND" node with two children. The first one is ROOT with $\boxed{\text{VP}}$'s parent or $\boxed{\text{PP|ADVP}}$ is removed. The second is a node corresponding to ROOT(**NP** + "is" + **VP**)

Coordinate Verb Phrases

ROOT «: (S < (NP $.. (VP <+(VP) ($\boxed{\text{VP}}$ > (**VP** ?$.. $\boxed{\text{CC}}$ & $.. **VP**)))))

Create a new node whose label depends on $\boxed{\text{CC}}$ ("and", "or", neitherNor", "nor"), with children the original ROOT node where $\boxed{\text{VP}}$ is replaced by one of its **VP** children

Coordinate Noun Phrase lists in Subject position

ROOT «: (S < ($\boxed{\text{NP}}$ < (**NP** ?$.. $\boxed{\text{CC}}$ & $.. **NP**) $.. VP))

Create a new node whose label depends on $\boxed{\text{CC}}$ ("and", "or", neitherNor", "nor"), with children the original ROOT node where $\boxed{\text{NP}}$ is replaced by one of its **NP** children

Coordinate Noun Phrase lists in Object position

ROOT «: (S < (NP $.. (VP « ($\boxed{\text{NP}}$ < (**NP** ?$.. $\boxed{\text{CC}}$ & $.. **NP2**)))))

Create a new node whose label depends on $\boxed{\text{CC}}$ ("and", "or", neitherNor", "nor"), with children the original ROOT node where $\boxed{\text{NP}}$ is replaced by one of its **NP** children

Table 4: List of our Transformation Patterns

C Prompts

C.1 Generation Prompt

We generated reasoning chains for the ProntoQA and FOLIO datasets using three different models: Mixtral 8x7B, LLaMa2-70B, and LLaMa3-70B. We used the following instruction prompt to obtain the chains.

> You will be presented with a passage and a question about that passage. There are options to be chosen from, only one of which is the correct answer. You need to write your step-by-step reasoning and conclude with the answer. One reasoning step should consist of a set of premises and one conclusion. The conclusion should be deduced immediately from the premises. Premises can only be sentences from the passage or previous conclusions. Follow the format of the example.
>
> Example Passage: Last night, Mark either went to play in the gym or visited his teacher Tony. If Mark drove last night, he didn't go to play in the gym. Mark would go visit his teacher Tony only if he and his teacher had an appointment. In fact, Mark had no appointment with his teacher Tony in advance.
> Example Question: Is it true that Mark drove last night?
> Example Options: A. Yes. B. No. C. Uncertain.
>
> Example Reasoning:
> Premise 1.1: Mark would go visit his teacher Tony only if he had an appointment.
> Premise 1.2: Mark had no appointment with his teacher Tony.
> Conclusion 1: Mark didn't visit his teacher, Tony.
> Premise 2.1: Mark either went to the gym or visited his teacher Tony.
> Premise 2.2: Mark didn't visit his teacher Tony.
> Conclusion 2: Mark went to the gym.
> Premise 3.1: If Mark drove last night, he didn't go to play in the gym.
> Premise 3.2: Mark went to the gym.
> Conclusion 3: Mark didn't drive last night.
> Premise 4.1: Mark didn't drive last night.
> Answer: B. No.
>
> Example Passage: David knows Mr. Zhang's friend Jack, and Jack knows David's friend Ms. Lin. Everyone of them who knows Jack has a master's degree, and everyone of them who knows Ms. Lin is from Shanghai.
> Example Question: Is it true that David is from Shanghai?
> Example Options: A. Yes. B. No. C. Uncertain.
>
> Example Reasoning:
> Premise 1.1: David knows Mr. Zhang's friend Jack, and Jack knows David's friend Ms. Lin.
> Conclusion 1: David is friend with Ms. Lin.
> Premise 2.1: David is friend with Ms. Lin.
> Premise 2.2: Everyone of them who knows Jack has a master's degree, and everyone of them who knows Ms. Lin is from Shanghai.
> Conclusion 2: David is from Shanghai.
> Premise 3.1: David is from Shanghai.
> Answer: A. Yes.

C.2 Natural Language Inference Prompt

> You are an expert linguistic annotator who performs Natural Language Inference: You will be presented with a premise and a hypothesis, and you shall answer whether the premise is equivalent to or entails the hypothesis (Entailment), contradicts it (Contradiction) or does not give enough information to conclude (Neutral). Answer "Entailment" or "Contradiction" only when you're absolutely confident, and "Neutral" the rest of the time (or when making assumptions). Answer only with "Entailment", "Contradiction" or "Neutral", nothing else.
> Premise: Ducks quack . Hypothesis: New York is a city .
> Answer: Neutral
> Premise: X love the Beatles . Hypothesis: X likes the Beatles .
> Answer: Entailment
> Premise: It is not true that X loves the Beatles . Hypothesis: X likes the Beatles .
> Answer: Neutral
> Premise: It is not true that X lives in France . Hypothesis: X lives in France .
> Answer: Contradiction
> Premise: Alice changes often . Hypothesis: Alice change often .
> Answer: Entailment
> Premise: It is not true that Jack is happy . Hypothesis: Jack is unhappy .
> Answer: Entailment
> Premise: X hates cats . Hypothesis: X is a cat person .
> Answer: Contradiction
> Premise: David has blonde hair . Hypothesis: David likes football .
> Answer: Neutral
> Premise: X are a child . Hypothesis: X is a kid.
> Answer: Entailment
> Premise: Alex loves fruits . Hypothesis: John hates pears .
> Answer: Neutral

D Examples of Reasoning Chains

Tables 5, 6 and 7 show example of reasoning chains from our benchmark. Premises in red indicate groundedness issues, and conclusions in red show that the step is invalid.

Dataset	Instance	Reasoning Chain
Entailment Bank (Hallu)	**Context**: The milky way is a kind of galaxy. A galaxy is made of stars. Stars are located light years apart from each other. Light years ly are an astronomy unit used for measuring length / distance generally used for values between 1 and 27000000000. **Question**: Is it true that light year can be used to measure the distance between the stars in milky way?	**Premise 1.1**: Stars are located light years apart from each other. **Premise 1.2**: Light is a kind of nonliving thing. **Conclusion 1**: Light year is used to measure the distance between stars. **Premise 2.1**: The milky way is a kind of galaxy. **Premise 2.2**: A galaxy is made of stars. **Conclusion 2**: The milky way is made of stars. **Premise 3.1**: Light year is used to measure the distance between stars. **Premise 3.2**: The milky way is made of stars. **Conclusion 3**: Light year can be used to measure the distance between the stars in milky way.
Entailment Bank (Neg)	**Context**: Hunting decreases animal populations. Decreases in an animal population can cause that animal to be extinct. A sloth is a kind of animal. Hunting is when a human kills an animal for food / recreation in the wild. **Question**: Is it true that that human hunting sloths can cause sloths to be extinct?	**Premise 1.1**: hunting decreases animal populations. **Premise 1.2**: decreases in an animal population can cause that animal to be extinct. **Conclusion 1**: hunting an animal can cause that animal to be extinct. **Premise 2.1**: hunting an animal can cause that animal to be extinct. **Premise 2.2**: a sloth is a kind of animal. **Conclusion 2**: hunting sloths can cause sloths to be extinct. **Premise 3.1**: hunting sloths can cause sloths to be extinct. **Premise 3.2**: a sloth is a kind of animal. **Premise 3.3**: hunting is not when a human kills an animal for food / recreation in the wild. **Conclusion 3**: human hunting sloths can cause sloths to be extinct.
ProofWriter (Neg)	**Context**: Charlie is big. Charlie is cold. Charlie is kind. Charlie is quiet. Charlie is red. Charlie is rough. Charlie is smart. Erin is kind. Fiona is quiet. Fiona is rough. Harry is kind. Harry is rough. Kind things are big. All kind, smart things are rough. If something is red and quiet then it is big. All red things are cold. All cold, quiet things are smart. If something is big and smart then it is cold. All quiet things are cold. Kind, big things are red. All cold, smart things are kind. **Question**: Is it true that Charlie is red?	**Premise 1.1**: All cold, quiet things are smart. **Premise 1.2**: Charlie is cold. **Premise 1.3**: Charlie is quiet. **Conclusion 1**: Charlie is smart. **Premise 2.1**: All cold, smart things are not kind. **Premise 2.2**: Charlie is cold. **Premise 2.3**: Charlie is smart. **Conclusion 2**: Charlie is kind. **Premise 3.1**: Kind, big things are red. **Premise 3.2**: Charlie is kind. **Premise 3.3**: Charlie is big. **Conclusion 3**: Charlie is red.
ProofWriter (Hallu)	**Context**: Bob is kind. Bob is nice. Bob is white. Charlie is kind. Charlie is white. Erin is red. Erin is rough. Harry is blue. Harry is kind. Harry is red. Blue people are nice. All white people are red. If someone is white and blue then they are nice. All rough people are red. If someone is smart then they are blue. If someone is kind then they are red. If someone is nice then they are rough. If someone is red then they are smart. If someone is red and rough then they are nice. **Question**: Is it true that Harry is smart?	**Premise 1.1**: If someone is kind then they are red. **Premise 1.2**: Harry is kind. **Conclusion 1**: Harry is red. **Premise 2.1**: If someone is red then they are smart. **Premise 2.2**: If something is blue then it chases the tiger. **Conclusion 2**: Harry is smart.

Table 5: Examples of generated reasoning chains for EntailmentBank and ProofWriter

ProofWriter (Remove)	**Context**: The bald eagle is blue. The bald eagle is kind. The bald eagle likes the cat. The bald eagle does not visit the tiger. The cat chases the mouse. The cat is green. The cat likes the bald eagle. The cat likes the mouse. The cat does not like the tiger. The mouse likes the cat. The tiger chases the cat. The tiger chases the mouse. The tiger is red. The tiger likes the cat. The tiger visits the cat. The tiger visits the mouse. If something likes the bald eagle then it is blue. If something visits the bald eagle and it visits the cat then the bald eagle is red. If something chases the mouse then it visits the cat. If something is blue then it chases the tiger. If something visits the cat and the cat chases the tiger then the tiger likes the bald eagle. If something likes the tiger then the tiger likes the bald eagle. If something chases the mouse then it visits the mouse. **Question**: Is it true that The tiger likes the bald eagle?	**Premise 1.1**: If something likes the bald eagle then it is blue. **Premise 1.2**: The cat likes the bald eagle. **Conclusion 1**: The cat is blue. **Premise 2.1**: If something is blue then it chases the tiger. **Conclusion 2**: The cat chases the tiger. **Premise 3.1**: If something visits the cat and the cat chases the tiger then the tiger likes the bald eagle. **Premise 3.2**: The tiger visits the cat. **Premise 3.3**: The cat chases the tiger. **Conclusion 3**: The tiger likes the bald eagle.
ProntoQA (Mixtral)	**Context**: Impuses are lempuses. Every zumpus is a numpus. Every gorpus is a sterpus. Every brimpus is a rompus. Everything that is a grimpus or a vumpus or a tumpus is a yumpus. Everything that is a tumpus or a wumpus or a dumpus is a sterpus. Every gorpus is a zumpus and an impus and a brimpus. Everything that is a grimpus or a vumpus or a tumpus is a wumpus. Impuses are tumpuses. Fae is a rompus. Fae is a gorpus. **Question**: Is it true that Fae is a yumpus?	**Premise 1.1**: Fae is a rompus. **Premise 1.2**: Fae is a gorpus. **Premise 1.3**: Every gorpus is a zumpus and an impus and a brimpus. **Premise 1.4**: Every brimpus is a rompus. **Premise 1.5**: Fae is a rompus. **Conclusion 1**: Fae is a zumpus and an impus and a brimpus. **Premise 2.1**: Everything that is a grimpus or a vumpus or a tumpus is a yumpus. **Premise 2.2**: Fae is not a grimpus or a vumpus or a tumpus. **Conclusion 2**: Fae is not a yumpus.
ProntoQA (LLaMa2)	**Context**: Every lempus is a numpus. Everything that is a shumpus or an impus or a dumpus is a numpus. Everything that is a shumpus or an impus or a dumpus is a brimpus and a lorpus and a tumpus. Everything that is a numpus or a tumpus or a vumpus is a sterpus. Everything that is a dumpus or a zumpus or a jompus is a wumpus. Every gorpus is a zumpus. Every brimpus is a numpus. Every lorpus is a vumpus. Max is a jompus and a zumpus and a wumpus. Max is a gorpus and a dumpus and a lempus. **Question**: Is it true that Max is a lempus, a lorpus, or a tumpus?	**Premise 1.1**: Every lempus is a numpus. **Premise 1.2**: Everything that is a shumpus or an impus or a dumpus is a numpus. **Conclusion 1**: Everything that is a shumpus or an impus or a dumpus is a lempus. **Premise 2.1**: Everything that is a numpus or a tumpus or a vumpus is a sterpus. **Premise 2.2**: Everything that is a dumpus or a zumpus or a jompus is a wumpus. **Conclusion 2**: Everything that is a dumpus or a zumpus or a jompus is a sterpus. **Premise 3.1**: Max is a jompus and a zumpus and a wumpus. **Premise 3.2**: Max is a gorpus and a dumpus and a lempus. **Conclusion 3**: Max is a sterpus.
ProntoQA (LLaMa3)	**Context**: Everything that is a shumpus or a tumpus or a brimpus is a vumpus. Dumpuses are wumpuses. Everything that is a wumpus or a brimpus or a gorpus is a tumpus. Jompuses are sterpuses. Everything that is a wumpus or a rompus or a shumpus is a brimpus. Every dumpus is a gorpus. Everything that is a wumpus or a rompus or a shumpus is a lorpus and a jompus and a dumpus. Every lorpus is a zumpus. Sam is a tumpus. Sam is a shumpus. **Question**: Is it true that Sam is an impus or a wumpus or a yumpus?	**Premise 1.1**: Sam is a tumpus. **Premise 1.2**: Everything that is a wumpus or a brimpus or a gorpus is a tumpus. **Conclusion 1**: Sam is a wumpus or a brimpus or a gorpus. **Premise 2.1**: Sam is a shumpus. **Premise 2.2**: Everything that is a wumpus or a rompus or a shumpus is a brimpus. **Conclusion 2**: Sam is a brimpus. **Premise 3.1**: Sam is a brimpus. **Premise 3.2**: Everything that is a shumpus or a tumpus or a brimpus is a vumpus. **Conclusion 3**: Sam is a vumpus. **Premise 4.1**: Sam is a wumpus or a brimpus or a gorpus. **Premise 4.2**: Dumpuses are wumpuses. **Premise 4.3**: Every dumpus is a gorpus. **Conclusion 4**: Sam is a wumpus.

Table 6: Examples of generated reasoning chains for ProofWriter and ProntoQA

FOLIO (Mixtral)	**Context**: If George likes music, he wants to compose. If George has access to a program, he can compose. If George can compose and wants to compose, he will compose. **Question**: Is it true that If George will not compose, George can not compose?	**Premise 1.1**: If George likes music, he wants to compose. **Premise 1.2**: If George has access to a program, he can compose. **Premise 1.3**: If George can compose and wants to compose, he will compose. **Conclusion 1**: George will compose if he likes music, has access to a program and wants to compose. **Premise 2.1**: George will not compose. **Conclusion 2**: George doesn't like music, doesn't have access to a program or doesn't want to compose. **Premise 3.1**: George will not compose. **Premise 3.2**: George can compose. **Conclusion 3**: George doesn't want to compose.
FOLIO (LLaMa2)	**Context**: All growth companies' stocks are volatile. If the stock price is volatile, then it is not suitable for a retirement fund. Some companieš2019 stocks are growth companies' stocks. All mature companies' stocks are suitable for a retirement fund. KO is a mature company's stock. **Question**: Is it true that KO is a company stock?	**Premise 1.1**: All growth companies' stocks are volatile. **Premise 1.2**: If the stock price is volatile, then it is not suitable for a retirement fund. **Conclusion 1**: Growth companies' stocks are not suitable for a retirement fund. **Premise 2.1**: Some companies' stocks are growth companies' stocks. **Conclusion 2**: Some companies' stocks are not suitable for a retirement fund. **Premise 3.1**: All mature companies' stocks are suitable for a retirement fund. **Premise 3.2**: KO is a mature company's stock. **Conclusion 3**: KO is suitable for a retirement fund. **Premise 4.1**: KO is a company stock. **Conclusion 4**: KO is a company stock.
FOLIO (LLaMa3)	**Context**: If people own a large desktop monitor at their home, then they work regularly and often at home. All people who have a fully stocked and impressive home office are people who own a large desktop monitor at their home. All people who work regularly and often at home spend their free time tidying and organizing the spaces in their homes. All people who work regularly and often at home visit the office supply store regularly. If people spend their free time tidying and organizing the spaces in their home, then they do not go to bed late and wake up every day after noon. Remy either both goes to bed late and wakes up every day after noon and spends their free time tidying and organizing the spaces in their home, or they do not go to bed late and wake up every day after noon nor spend their free time tidying and organizing the spaces in their home. **Question**: Is it true that Remy visits the office supply store regularly?	**Premise 1.1**: Remy either both goes to bed late and wakes up every day after noon and spends their free time tidying and organizing the spaces in their home, or they do not go to bed late and wake up every day after noon nor spend their free time tidying and organizing the spaces in their home. **Premise 1.2**: If people spend their free time tidying and organizing the spaces in their home, then they do not go to bed late and wake up every day after noon. **Conclusion 1**: Remy does not spend their free time tidying and organizing the spaces in their home. **Premise 2.1**: All people who work regularly and often at home spend their free time tidying and organizing the spaces in their home. **Premise 2.2**: Remy does not spend their free time tidying and organizing the spaces in their home. **Conclusion 2**: Remy does not work regularly and often at home. **Premise 3.1**: All people who work regularly and often at home visit the office supply store regularly. **Premise 3.2**: Remy does not work regularly and often at home. **Conclusion 3**: Remy does not visit the office supply store regularly.

Table 7: Examples of generated reasoning chains for FOLIO

E Error Analysis

We performed a manual error analysis for the false positives of validity detection on the LLaMa3-generated Chains-of-Thought on FOLIO for all methods, with statistics in Table 8. Some of the errors originate from the Logic Transformation phase, which was not able to detect splits where one should have been performed. Yet, the main source of errors for VANESSA is the NLI module, which often detects invalid entailments, leading to incorrect predictions by the model. Even when parsing is insufficient, there needs to be such a NLI mistake for a False Positive to be produced by VANESSA. It is interesting to see that errors happen particularly when the conclusion is known to be true in the real-world (e.g. "Djokovic is a professional tennis player", or "This statement is true"), because it shows that the NLI might have absorbed too much information during training. This kind of entailment standing is reminiscent of the problems met with material implication. Table 11 shows examples of false positives.

Validation Method	NLI	Parsing
VANESSA-Symbolic	0	1
VANESSA-LLaMa3	50	11
VANESSA-Deberta	27	9
LINC-LLaMa3	0	31
NLI-LLaMa3	125	0
NLI-DeBERTa	75	0

Table 8: Main sources of False Positive error on FOLIO-LLaMa3

F GPT 3.5 for NLI

Results with GPT 3.5-Turbo for validity verification are shown in Table 9.

Dataset	Method	Precision	Recall	F0.5	D
ProofWriter-Neg	NLI LLaMa3	$0.95_{\pm 0.01}$	$0.88_{\pm 0.02}$	$0.93_{\pm 0.01}$	0.83
	NLI DeBERTa	$0.88_{\pm 0.03}$	$0.26_{\pm 0.02}$	$0.6_{\pm 0.03}$	0.31
	NLI GPT	$0.78_{\pm 0.03}$	$0.49_{\pm 0.02}$	$0.69_{\pm 0.02}$	0.35
ProofWriter-Remove	NLI LLaMa3	$0.78_{\pm 0.02}$	$0.87_{\pm 0.02}$	$0.79_{\pm 0.02}$	0.64
	NLI DeBERTa	$0.65_{\pm 0.04}$	$0.26_{\pm 0.02}$	$0.5_{\pm 0.03}$	0.16
	NLI GPT	$0.94_{\pm 0.02}$	$0.49_{\pm 0.03}$	$0.79_{\pm 0.02}$	0.53
ProofWriter-Hallu	NLI LLaMa3	$0.75_{\pm 0.02}$	$0.88_{\pm 0.02}$	$0.77_{\pm 0.02}$	0.62
	NLI DeBERTa	$0.63_{\pm 0.04}$	$0.29_{\pm 0.02}$	$0.51_{\pm 0.03}$	0.16
	NLI GPT	$0.86_{\pm 0.02}$	$0.5_{\pm 0.03}$	$0.75_{\pm 0.02}$	0.48
ProntoQA-Mixtral	NLI LLaMa3	$0.76_{\pm 0.05}$	$0.93_{\pm 0.03}$	$0.78_{\pm 0.04}$	0.43
	NLI DeBERTa	$0.73_{\pm 0.07}$	$0.4_{\pm 0.06}$	$0.62_{\pm 0.07}$	0.11
	NLI GPT	$0.76_{\pm 0.05}$	$0.82_{\pm 0.05}$	$0.76_{\pm 0.04}$	0.31
ProntoQA-LLaMa2	NLI LLaMa3	$0.51_{\pm 0.04}$	$0.97_{\pm 0.02}$	$0.56_{\pm 0.04}$	0.4
	NLI DeBERTa	$0.55_{\pm 0.06}$	$0.48_{\pm 0.05}$	$0.54_{\pm 0.05}$	0.22
	NLI GPT	$0.58_{\pm 0.04}$	$0.86_{\pm 0.04}$	$0.61_{\pm 0.04}$	0.45
ProntoQA-LLaMa3	NLI LLaMa3	$0.74_{\pm 0.04}$	$0.95_{\pm 0.02}$	$0.77_{\pm 0.03}$	0.33
	NLI DeBERTa	$0.79_{\pm 0.06}$	$0.36_{\pm 0.05}$	$0.63_{\pm 0.06}$	0.16
	NLI GPT	$0.78_{\pm 0.04}$	$0.71_{\pm 0.04}$	$0.75_{\pm 0.04}$	0.25
FOLIO-Mixtral	NLI LLaMa3	$0.66_{\pm 0.02}$	$0.88_{\pm 0.02}$	$0.69_{\pm 0.02}$	0.25
	NLI DeBERTa	$0.74_{\pm 0.03}$	$0.59_{\pm 0.03}$	$0.7_{\pm 0.03}$	0.28
	NLI GPT	$0.76_{\pm 0.03}$	$0.77_{\pm 0.03}$	$0.76_{\pm 0.02}$	0.43
FOLIO-LLaMa2	NLI LLaMa3	$0.6_{\pm 0.02}$	$0.95_{\pm 0.01}$	$0.65_{\pm 0.02}$	0.36
	NLI DeBERTa	$0.72_{\pm 0.02}$	$0.69_{\pm 0.03}$	$0.71_{\pm 0.02}$	0.41
	NLI GPT	$0.66_{\pm 0.02}$	$0.84_{\pm 0.02}$	$0.69_{\pm 0.02}$	0.4
FOLIO-LLaMa3	NLI LLaMa3	$0.74_{\pm 0.02}$	$0.89_{\pm 0.02}$	$0.77_{\pm 0.02}$	0.31
	NLI DeBERTa	$0.77_{\pm 0.02}$	$0.62_{\pm 0.02}$	$0.73_{\pm 0.02}$	0.22
	NLI GPT	$0.71_{\pm 0.03}$	$0.72_{\pm 0.03}$	$0.71_{\pm 0.02}$	0.07
EntailmentBank-Neg	NLI LLaMa3	$0.67_{\pm 0.03}$	$0.99_{\pm 0.01}$	$0.71_{\pm 0.03}$	0.51
	NLI DeBERTa	$0.77_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.78_{\pm 0.03}$	0.61
	NLI GPT	$0.74_{\pm 0.03}$	$0.86_{\pm 0.02}$	$0.76_{\pm 0.03}$	0.56
EntailmentBank-Hally	NLI LLaMa3	$0.6_{\pm 0.03}$	$0.98_{\pm 0.01}$	$0.65_{\pm 0.02}$	0.3
	NLI DeBERTa	$0.74_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.76_{\pm 0.02}$	0.5
	NLI GPT	$0.67_{\pm 0.03}$	$0.87_{\pm 0.02}$	$0.7_{\pm 0.03}$	0.37

Table 9: Validation Results for different NLI models, including GPT 3.5-Turbo

G Ablation Study - LINC

We experimented with the LINC Framework using GPT 3.5-Turbo, as in the original paper. Table 10 shows the results we obtained for validation on ProntoQA and FOLIO. These results encouraged us to use LLaMa3, which performs similarly (and is cheaper at that).

Dataset	Method	Error Rate	Precision	Recall	F0.5	D
Mixtral-ProntoQA	LINC LLaMa3	9%	$0.94_{\pm 0.03}$	$0.85_{\pm 0.04}$	$0.9_{\pm 0.03}$	0.77
	LINC GPT	2%	$0.91_{\pm 0.03}$	$0.91_{\pm 0.03}$	$0.89_{\pm 0.03}$	0.8
LLaMa-ProntoQA	LINC LLaMa3	3%	$0.86_{\pm 0.04}$	$0.89_{\pm 0.03}$	$0.85_{\pm 0.03}$	0.81
	LINC GPT	4%	$0.87_{\pm 0.03}$	$0.94_{\pm 0.02}$	$0.87_{\pm 0.03}$	0.87
LLaMa3-ProntoQA	LINC LLaMa3	3%	$0.96_{\pm 0.02}$	$0.97_{\pm 0.01}$	$0.95_{\pm 0.02}$	0.95
	LINC GPT	3%	$0.97_{\pm 0.01}$	$0.95_{\pm 0.02}$	$0.95_{\pm 0.02}$	0.93
Mixtral-FOLIO	LINC LLaMa3	30%	$0.83_{\pm 0.03}$	$0.48_{\pm 0.03}$	$0.72_{\pm 0.03}$	0.35
	LINC GPT	32%	$0.86_{\pm 0.03}$	$0.47_{\pm 0.03}$	$0.73_{\pm 0.03}$	0.38
LLaMa-FOLIO	LINC LLaMa3	25%	$0.85_{\pm 0.03}$	$0.48_{\pm 0.03}$	$0.73_{\pm 0.03}$	0.43
	LINC GPT	24%	$0.87_{\pm 0.02}$	$0.52_{\pm 0.03}$	$0.76_{\pm 0.02}$	0.48
LLaMa3-FOLIO	LINC LLaMa3	24%	$0.87_{\pm 0.02}$	$0.55_{\pm 0.02}$	$0.78_{\pm 0.02}$	0.37
	LINC GPT	25%	$0.89_{\pm 0.02}$	$0.56_{\pm 0.02}$	$0.79_{\pm 0.02}$	0.4

Table 10: Validation Results for LINC using different LLMs for the first-order logic conversion

Method	Reasoning Step	Explanation
VANESSA-Symbolic	**Premise 3.1**: If a design by Max is timeless, then a design by Max is a mass product design and evocative. **Premise 3.2**: (Not applicable, as we don't know if a design by Max is timeless) **Conclusion 3**: (Not applicable) **Premise 3.1**: (A: X is a design by Max $\wedge$ B: X is timeless) $\Rightarrow$ C: X is a mass product design and evocative **Premise 3.1**: D: . **Conclusion 3**: E: . **NLI**: E $\triangleright$ D, D $\triangleright$ E	The generated step doesn't make sense, which is why it was labeled deemed invalid. The transformation is weak, with C not being split because of the "and", and D and E being empty. Yet the NLI module recognizes that one empty sentence entails an other.
VANESSA-DeBERTa	**Premise 3.1**: An animal is either a monkey or a bird. **Premise 3.2**: Rock is not an animal. **Conclusion 3**: Rock is neither a monkey nor a bird. **Premise 3.1**: G: Rock is an animal $\Rightarrow$ (H: Rock is a monkey $\oplus$ I: Rock is a bird) **Premise 3.2**: D: Rock is not an animal **Conclusion 3**: E: Rock is not a monkey $\wedge$ F: Rock is not a bird. **NLI**: $D \blacktriangleright H$, $D \blacktriangleright I$, $\neg E \triangleright G$, $\neg F \triangleright G$	The example considers that animals can only be monkeys or birds, but does not specify that monkey and birds are necessarily animals (which is true in the real life). The NLI's output is faithful to real life, giving is for instance that "Rock is not an animal" contradicts "Rock is a monkey". Modus Tollens is applied, and VANESSA produces a False Positive.
NLI-DeBERTa	**Premise 3.1**: An animal is either a monkey or a bird. **Premise 3.2**: Rock is not an animal. **Conclusion 3**: Rock is neither a monkey nor a bird.	Using NLI directly causes the same error as previously, due to the difference between the problem setup and the real-life truth that the NLI model has absorbed during training.
VANESSA-LLaMa3	**Premise 4.1**: Djokovic is a Grand Slam champion. **Premise 4.2**: If a person is a celebrity then they are well paid. **Premise 4.3**: All Oscar-nominated actors are celebrities. **Conclusion 4**: Djokovic is well paid. **Premise 4.1**: A: Djokovic is a Grand Slam champion **Premise 4.2**: (H: Djokovic is a person $\wedge$ I: Djokovic is a celebrity) $\Rightarrow$ J: Djokovic are well paid **Premise 4.3**: K: Djokovic is an Oscar - nominated actor $\Rightarrow$ L: Djokovic is a celebrity **Conclusion 4**: G:Djokovic is well paid **NLI**: $A \triangleright H$, $A \triangleright I$, $J \triangleright G$	This error is due to the NLI model going "too far" with its conclusions. It predicts that if Djokovic is a Grand Slam champion, then he must be a celebrity, which is not necessarily the case.
NLI-LLAMa3	**Premise 4.1**: Employees will either have lunch in the company or have lunch at home. **Premise 4.2**: James does not work remotely from home. **Conclusion 4**: James has lunch in the company.	The NLI model directly entails that if James does not work remotely from home, then he has lunch in the company, while it is possible to work on-site but have lunch at home.

Table 11: Examples of false positives for different methods on FOLIO-LLaMa3