Investigating Transformer guided Chaining for Interpretable Natural Logic Reasoning

Rajaraman Kanagasabai, Saravanan Rajamanickam and Shi Wei

Agency for Science, Technology and Research (A*STAR)
Institute for Infocomm Research
Singapore 138632
{kanagasa, saravanan_rajamanickam, shi_wei}@i2r.a-star.edu.sg

Abstract

Natural logic reasoning has received increasing attention lately, with several datasets and neural models proposed, though with limited success. More recently, a new class of works have emerged adopting a Neuro-Symbolic approach, called *transformer guided chaining*, whereby the idea is to iteratively perform 1-step neural inferences and chain together the results to generate a multi-step reasoning trace. Several works have adapted variants of this central idea and reported significantly high accuracies compared to vanilla LLM's. In this paper, we perform a critical empirical investigation of the chaining approach on a multi-hop First-Order Logic (FOL) reasoning benchmark. In particular, we develop a reference implementation, called *Chainformer*, and conduct several experiments to analyze the accuracy, generalization, interpretability, and performance over FOLs. Our findings highlight key strengths and possible current limitations and suggest potential areas for future research in logic reasoning.

1 Introduction

We consider deductive reasoning over Natural Logic (MacCartney and Manning, 2014; Moss, 2010), i.e., reasoning over statements expressed in language. Natural logic reasoning has received increasing attention lately, with several datasets (Yu et al., 2020; Liu et al., 2020; Clark et al., 2020; Dalvi et.al. 2021) and neural models proposed (Huang et.al. 2021, Jiao et.al. 2022, Clark et al., 2020; Saha et.al. 2020; Wang et.al. 2021; Xu et.al. 2022; Pi et.al. 2022).

Most of the traditional neural approaches tackled multi-step reasoning as a single pass 'all-at-once' inference. For reasoning problems that are inherently multi-step, it is more natural to consider a symbolic machinery in tandem with the neural model. Taking inspiration from this philosophy, a new class of works have emerged recently by combining neural models (popularly using transformers (Vaswani et.al. 2017)) with symbolic chaining. The central idea is to iteratively perform 1-step neural inferences and chain together the results to generate a multi-step reasoning trace. ProofWriter (Tafjord et.al. 2021) was one of the first to explore this idea, and demonstrate >95% multi-hop reasoning accuracy on several synthetic datasets. (Picco et.al. 2021) and (Bostrom 2022) reported similar results. Recently, several works (Qu et.al. 2022; Yang et.al. 2022; Tafjord et.al. 2022; Ghosal et.al. 2022; Ribeiro et.al. 2022; Hong et.al. 2022) applied variants of this approach on EntailmentBank (Dalvi et.al. 2021) and showed superior performance. The iterative approach is attractive because i) it is *faithful* in that it naturally reflects the internal reasoning process, and is inherently interpretable, ii) it has been shown to be easily adapted for multiple choice Q&A (Shi et.al. 2021) and open-ended Q&A (Tafjord 2022), besides Natural Language Inference (NLI), iii) it enables teachable reasoning (Dalvi et.al. 2022).

While the above results are promising, we argue that an unbiased third-party investigation is important to facilitate a better understanding of the strengths and weaknesses. This is the main goal of this paper. Towards this, we develop a reference implementation, called *Chainformer* that captures

Findings of the Association for Computational Linguistics: ACL 2023, pages 9240–9253

the core idea behind the chaining approach, and benchmark on a multi-hop FOL reasoning task using a recently proposed diagnostic dataset, called LogicNLI (Tian et.al. 2021). The dataset is composed of a rich class of FOLs that go beyond conjunctive implications and is non-trivial with a reported human reasoning accuracy of 77.5% (Tian et.al. 2021).

Entailment	$P \vdash \text{h} \wedge P \nvdash \neg\text{h}$
Contradiction	$P \nvdash \text{h} \wedge P \vdash \neg\text{h}$
Neutral	$P \nvdash \text{h} \wedge P \nvdash \neg\text{h}$
Paradox	$P \vdash \text{h} \wedge P \vdash \neg\text{h}$

Table 1. Inference Relations between P and h.

We conduct several experiments to analyze the performance in terms of accuracy, generalization, interpretability, and expressiveness over FOLs. Our key findings are: 1) human level multi-step reasoning performance is achieved (84.5% machine vs 77.5% human), with a minimalist transformer guided chaining implementation, and even with a base model (80.4% base vs 84.5% large). However, this requires the 1-step inferences be carefully trained for high accuracy; 2) the inferred reasoning chains are correct 78% of the time but could be more than twice longer than the optimal chains; 3) FOLs with simple conjunctions and existential quantifiers are easier to handle, whereas FOLs with equivalence are harder especially with universal quantifiers and disjunctions. Our results highlight the key strengths of the transformer-guided chaining approach and faithful reasoning in general, and suggest possible weaknesses that could motivate future research in multi-hop reasoning.

In related work, (Yu et al., 2020; Liu et al., 2020; Dalvi et.al. 2021; Tian et.al. 2021) have performed diagnostic studies on popular language models and pointed out limitations in logic reasoning capabilities. (Li et.al. 2022) investigated NLU datasets to measure correlation with logic reasoning as a key skill. Our focus is different, and we aim to specifically analyze the iterative reasoning strategy for multi-hop logic reasoning, and hence is novel.

2 Problem Definition

We consider the NLI setting (Bowman 2015; Storks et.al. 2019). Let F = $\{f_1, f_2, \cdots, f_n\}$, be n simple sentences, called *Facts*; R = $\{r_1, r_2, \cdots, r_m\}$, a set of m compound sentences, called *Rules*. Then, given the tuple P = (F, R), called the *Premise*, and a statement *h*, called the *Hypothesis*; the inference problem is to determine i) the inference relation of h, and ii) a reasoning chain X, where $X = \{X_1, X_2, \ldots, X_i, \ldots, X_k\}$ is a sequence such that $X_i = (r_i, F_i)$, where $r_i \in R$ and F_i is a set of *intermediate facts*, with members not necessarily from F.

The inference relations can be *entailment*, *contradiction*, *neutral*, or *paradox*, as defined in Table 1, where $\vdash$ is the entailment operator.

It is easy to see that the complexity of the problem varies based on the constraints imposed on F, R, X and the target inference labels of h. For example, RuleTaker (Clark et.al. 2020.) considers h to be 'true or 'false'. Additionally, R is restricted to be implication rules with conjunctions and negations. ProofWriter (Tafjord et.al. 2021) adopts a similar setting but allows h also to be un-determined ('neutral'),

In this paper we consider a more general NLI problem following (Tian et.al 2021), where i) R is expressed using a rich class of FOLs with universal $\forall$ and existential $\exists$ quantifiers, logic connectives such as disjunctions $\vee$, implications $\rightarrow$, equivalence $\equiv$ and negations $\neg$..ii) h can take any of the 4 inference labels (Table 1). Figure A-1 in the Appendix presents a sample problem instance.

3 Logic Reasoning Method

Logic reasoning using chaining strategy can be implemented in several ways, e.g. with fact selection (Bostrom et.al. 2022), rule selection (Sanyal et.al. 2022), inference verification (Tafjord et.al. 2022), etc. We aim to adopt a minimalist implementation, as we believe it facilitates better examination of the strengths and weaknesses of the central methodology,.

We consider the Forward Chaining algorithm from Sec 9.3.2 of (Russell et.al. 2010), which is known to be sound and complete for a rich class of FOLs. Basically, the algorithm starts with the known facts and applies rules whose preconditions are satisfied, to infer new facts repeatedly until the hypothesis can be verified. To extend to natural language, our idea is to employ a transformer

model to do fact unification and rule inference, and a second transformer to verify the given hypothesis against the currently known facts.

Rule Inference: In this step, given the current known facts and a rule, the rule preconditions are matched through unification to check for a rule match. If the latter succeeds, new facts are inferred (intermediate facts); otherwise, no facts are generated and the control moves to the next rule. We model this as an abstractive Q&A task, with the current facts as the 'context', the chosen rule as the 'question' and the inferred facts as the desired 'answer'. A T5 transformer model (Raffel et.al. 2020) is employed for this purpose. In particular, the processed input to the model is 'question: <rule> context: <known facts>' and the output is 'inferred facts' if the rule can be triggered and 'none' otherwise.

Facts Checking: This step verifies the given hypothesis against the currently known facts based on Table 1. In our implementation, we accomplish this by formulating a 2-class NLI task, for inferring $F' \vdash h$ and $F' \vdash \neg h$, where F' is the currently known facts.

Assemble Chain Additionally, for interpretability, we store the rule and the intermediate facts, every time a rule is satisfied. If the hypothesis is successfully verified, then the stored rules and facts are assembled to form a reasoning chain and returned.

An outline of the complete algorithm (Figure A-2), and an illustration (Figure A-4) are presented in in the Appendix, along with the training details.

4 Experiments and Results

We perform several experiments using the multi-hop FOL reasoning dataset LogicNLI, (Tian et.al 2021). The dataset includes 16K/2K/2K train/dev/test instances, with each instance consisting of over 12 facts and 12 rules, along with labeled statements and reasoning chains (called proof paths). A sample instance is in Figure A-4.

The results are presented in the following subsections. In all the tabulated results, the performance metrics are averaged over 10 runs and quoted in % for easier interpretation, unless stated otherwise. Details about the implementation, hyper-parameter settings and machine configuration are provided in Appendix A.

4.1 Comparison with Baselines (Table 2)

Firstly, we compare the performance in terms of accuracy against the baseline language models BERT (Delvin et.al. 2019), RoBERTa (Liu et.al. 2019), and XLNet (Yang et.al. 2019). Additionally, we considered a naïve algorithm, called *NaiveFactsChecker*, that does facts checking as in Sec 3 but without rule inference.

Models	**Accuracy (%)**	
	Dev	**Test**
Random	25.0	
Human	77.5	
BERT-base	30.1	29.5
RoBERTa-base	59.5	58.0
BERT-large+MLP	57.0	55.9
RoBERTa-large+MLP	65.0	68.3
XLNet + MLP layer	64.0	65.4
NaiveFactsChecker	50.1	51.2
Chainformer + t5-base	78.1	80.4
Chainformer + t5-large	**80.2**	**84.5**

Table 2: Comparison of Accuracy against Baseline models on Dev/Test

We observe that *NaiveFactsChecker* achieved ~50% (2x more than *Random*), suggesting that about 50% of the hypotheses in LogicNLI may be verifiable from the given facts alone. All LM baselines, barring *BERT-base*, performed better, with *RoBERTa-large+MLP* the best model. In comparison, *Chainformer* significantly outperformed all baselines and even exceeded human performance. This is surprising given that our implementation was minimalist without other functionalities often used in the published approaches. We argue that the results highlight the strength of iterative LM-guided reasoning over 'all-at-once' approach. Furthermore, the t5-base model version was comparable in performance to the t5-large version, which gives promise for low-compute possibilities in implementing logic reasoning.

4.2 Detailed Performance Analysis

Here we investigate Chainformer approach in more detail to derive further insights.

4.2.1 Generalization (Figure 1)

To analyze the generalization ability of our approach, we varied training instances from 2400 (25%), 4800 (50%), 7200 (75%) and 9600 (100%) and measured performance of i) 1-step inference, and ii) final reasoning (Figure 1).

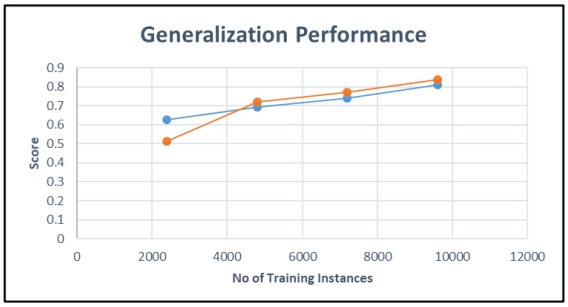


Figure 1 : Performance over various 1-step inference training set sizes, where 1-step accuracy is plotted in Blue and the final reasoning performance in Orange.

We observe an almost linear improvement, indicating good generalization.

4.2.2 Performance over FOLs (Table 3, 4 & 5)

For our next, experiment, we studied the ability in reasoning over various FOL classes. LogicNLI contains 23 FOL classes in total, and we first analyzed Chainformer to determine the respective inference accuracies. A summary of the results is presented in Table 3, 4 and 5. Details about the individual classes and the respective accuracies are provided in the Appendix.

We notice that FOLs with logical equivalence are harder than implication, rather unsurprisingly, and the easiest with neither of them (Table 4). Similarly, disjunctions are harder than conjunctions (Table 5). Universal quantifiers are harder than no quantifiers, but existential quantifiers are easier in comparison (Table 3). A possible explanation is that neural unification is easier when matching any one relevant fact is sufficient rather than requiring the same for all relevant facts. However, this depends on the modeling and implementation specifics. It might be possible to alter the behavior with approaches, e.g. using different angle than 'Abstractive Q&A' (Section 3), but this needs more research.

	Analysis of Quantifiers		
	∃	∀	None
# of FOL Classes	8	9	6
Accuracy	91.4	65.3	88.4

Table 3 Accuracy over FOLs w.r.to Quantifiers

	Analysis of Connectives I		
	→	≡	None
# of FOL lasses	15	6	2
Accuracy	81.2	74.5	92.3

Table 4 Analysis of Accuracy over FOLs w.r.to Implication → and Equivalence ≡

	Analysis of Connectives II		
	∧	∨	None
# of FOL Classes	15	5	5
Accuracy	77.2	70.3	85.7

Table 5 Analysis of Accuracy over FOLs w.r.to Conjunctions ∧ and Disjunctions ∨

4.2.3 Interpretability (Table 6 & 7)

We next analyzed interpretability of the predicted reasoning chains, by asking two questions i) *Is the chain correct?* and ii) *Is the chain optimal compared to the ground truth*.

Towards this, we define two metrics viz. *correctness* and *minimality*. A chain is deemed correct if and only if every chain fragment corresponds to a valid entailment. Minimality is defined as the ratio of the length of the target chain over the length of the predicted chain. Note that a chain may be incorrect even if one step corresponds to an invalid entailment. Thus, we may have situations where the hypothesis is successfully inferred but the chain is incorrect. Such chains are called *partially correct*.

As an exhaustive analysis of all chains is arduous, we sampled 200 'entailment' and 200 'contradiction' instances from the predicted chains, as a preliminary evaluation, and tasked a student (not part of the project) to manually label the validity of every chain fragment. The labels were later verified via a random check by two project members to remove incorrect entries. The results are presented in Table 6.

On average, we observed that 78.8% of the chains were fully correct (Table 6), providing

support for chaining as a faithful reasoning approach. In fact, about 10% of the chains were partially correct and only 11.2% were incorrect.

To analyze minimality, we extracted the correct chains and computed the minimality score against the gold standard chains. An overall average score of 0.42 was observed (Table 7), implying that the correctly predicted chains could be 2.3 times longer than the optimal ones.

Label		Correctness
Entailment	Correct	73.5
	Incorrect	12.8
	Partially correct	13.7
Contradiction	Correct	84.1
	Incorrect	9.8
	Partially correct	6.1

Table 6: Correctness of Predicted Chains

Label	Minimality Score
Entailment	0.44
Contradiction	0.40

Table 7 Minimality of Verified Chains

5 Discussion and Conclusions

We considered the recently emerging neuro-symbolic approach for addressing multi-step natural logic reasoning, called the *transformer guided chaining*. The approach adopts an iterative reasoning strategy in contrast to the traditional neural approaches that tackle multi-step reasoning as a single pass 'all-at-once' inference. The iterative approach is attractive as it offers several advantages such as i) it is *faithful* in that it naturally reflects the internal reasoning process, ii) it is inherently interpretable, iii) it can be applied to multiple choice Q&A and open-ended Q&A, besides Natural Language Inference.

We performed a detailed empirical investigation of this approach, using a challenging FOL reasoning dataset. Our key findings are: 1) human level performance is achieved on multi-hop FOL reasoning task with a minimalist implementation (80.4% machine vs 77.5% human), and even with a base model (80.4% base vs 84.5% large). This provides support for the potential of chaining strategy and encourages possible applications on real life texts; 2) FOLs with simple conjunctions and existential quantifiers are easier to handle, whereas FOLs with equivalence are harder especially with universal quantifiers and disjunctions, suggesting scope for further research; 3) the predicted reasoning chains are correct 78% of the time, but could be more than twice longer than the optimal chains. The latter implies that two or more *correct* reasoning chains are possible, and iterative reasoning strategy might return one of them (though sub-optimal). This underscores the importance of human validation in interpretability evaluation, as automating it, say by scoring exact match, is likely to underestimate the true performance,

A key observation is that the approach hinges on how accurately the 1-step inferences can be performed, as small errors can propagate over multiple iterations and get magnified. For example, if the rule inference step results in false positives/negatives, it is unclear how the chaining performance will be impacted. In addition, if facts are incomplete or even inconsistent, how effective will the reasoning be? These are interesting research questions for further investigation. (Ghosal et.al. 2022; Dalvi et.al. 2022) are steps along this direction.

On another direction, most of the chaining-based works have considered mainly 'entailment' as the inference relation. To handle real-life texts, it is important to go beyond simple entailment relations, and consider more sophisticated ones, e.g. necessity, possibility and rebuttal (MacCartney and Manning, 2014; Huang et.al. 2022). To, cover such relations, new models and approaches are required, and they could facilitate enhancing the scope of current faithful reasoning approaches towards addressing advanced multi-hop reasoning scenarios.

5.1 Limitations

Our work is one of the first to perform a detailed empirical investigation of transformer guided chaining but is clearly preliminary. The following are some key limitations:

- Evaluation of Interpretability: A fair evaluation of interpretability is not straightforward. In this paper, we reported results from a preliminary study with limited human labor.
- Analysis of negations: LogicNLI dataset uses negations in the facts, rules and statements but it is difficult to disentangle them for a fair investigation. Hence, we were unable to rigorously analyze the ability in handling negations.

- Evaluation on Real-life data: Our reported work focused on a synthetic dataset. For a more rigorous evaluation, it is imperative to consider more datasets including real-life ones.

References

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

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2020. Transformers as soft reasoners over language. *In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3882–3890. International Joint Conferences on Artificial Intelligence Organization. Main track

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

Bhavana Dalvi, , Oyvind Tafjord, and Peter Clark. 2022. "Towards teachable reasoning systems." arXiv preprint arXiv:2204.13074.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In NAACL-HLT.

Deepanway Ghosal, Aditya, S. and Choudhury, M., 2022. Generating Intermediate Steps for NLI with Next-Step Supervision. *arXiv preprint arXiv:2208.14641*.

Chadi Helwe, Chloé Clavel, and Fabian M Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In 3rd Conference on Automated Knowledge Base Construction.

Chadi Helwe, Chloé Clavel, and Fabian Suchanek. 2022. LogiTorch: A PyTorch-based library for logical reasoning on natural language. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Ruixin Hong, Hongming Zhang, Xintong Yu, and Changshui Zhang. 2022. METGEN: A Module-Based Entailment Tree Generation Framework for Answer Explanation. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 1887–1905, Seattle, United States. Association for Computational Linguistics

Yinya Huang, Meng Fang, Yu Cao, Liwei Wang, Xiaodan Liang. 2021. DAGN: Discourse-aware graph network for logical reasoning. *In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5848-5855. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.467.

Yinya Huang, Zhang Hongming, Hong Ruixin, Liang Xiaodan, Zhang Changshui, and Yu Dong. 2022. MetaLogic: Logical Reasoning Explanations with Fine-Grained Structure. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing,

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 1930–1940, New Orleans, Louisiana*. Association for Computational Linguistics

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Fangkai Jiao, Yangyang Guo, Xuemeng Song, and Liqiang Nie. 2022. MERIt: Meta-Path Guided Contrastive Learning for Logical Reasoning. *In Findings of the Association for Computational Linguistics: ACL 2022, pages 3496–3509, Dublin, Ireland.* Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.276.

Nora Kassner, Benno Krojer, and Hinrich Schütze. 2020. Are Pretrained Language Models Symbolic Reasoners over Knowledge? *In Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 552–564, Online. Association for Computational Linguistics.

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, pages 3622-3628. https://doi.org/10.24963/ijcai.2020/501.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR.

Yitian Li, Jidong Tian, Wenqing Chen, Caoyun Fan, Hao He, and Yaohui Jin. 2022. To What Extent Do Natural Language Understanding Datasets Correlate to Logical Reasoning? A Method for Diagnosing Logical Reasoning. In Proceedings of the 29th International Conference on Computational Linguistics, pages 1708–1717, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Qing Lyu, Marianna Apidianaki, Chris Callison-Burch. 2022. Towards Faithful Model Explanation in NLP: A Survey. *arXiv preprint arXiv:2209.11326*.

Bill MacCartney and Chris Manning. 2014. Natural logic and natural language inference. Computing Meaning, 47:129–147, 2014.

Bill MacCartney and Christopher D Manning. 2009. An extended model of natural logic. *In Proceedings of the eighth international conference on computational semantics, pp. 140–156.* Association for Computational Linguistics.

Lawrence S Moss. 2010. Natural logic and semantics. *In Logic, Language and Meaning,* pages 84–93. Springer.

Siru Ouyang, Zhuosheng Zhang, Hai Zhao. 2021. Fact-driven logical reasoning. *Computing Research Repository,* arXiv:2105.10334.

Gabriele Picco, Thanh Lam Hoang, Marco Luca Sbodio, and Vanessa Lopez. 2021. Neural Unification for Logic Reasoning over Natural Language. In Findings of the Association for Computational Linguistics: EMNLP 2021, pages 3939–3950, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xinyu Pi, Wanjun Zhong, Yan Gao, Nan Duan, Jian-Guang Lou. 2022. LogiGAN: Learning logical reasoning via adversarial pre-training. *In Proceedings of the 36th Conference on Neural Information Processing Systems.*

Hanhao Qu, Yu Cao, Jun Gao, Liang Ding, and Ruifeng Xu. 2022. Interpretable Proof Generation via Iterative Backward Reasoning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2968–2981, Seattle, United States. Association for Computational Linguistics.

Danilo Neves Ribeiro, Shen Wang, Xiaofei Ma, Rui Dong, Xiaokai Wei, Henghui Zhu, Xinchi Chen, Peng Xu, Zhiheng Huang, Andrew Arnold, and Dan Roth. 2022. Entailment Tree Explanations via Iterative Retrieval-Generation Reasoner. In Findings of the Association for Computational Linguistics: NAACL 2022, pages 465–475, Seattle, United States. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *In Journal of Machine Learning Research.*

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. *In Proceedings of the 2016 Conference on Empirical Methods in Natu- ral Language Processing, pages 2383–2392, Austin, Texas*. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.*

Sanjay Subramanian, Ben Bogin, Nitish Gupta, Tomer Wolfson, Sameer Singh, Jonathan Berant, and Matt Gardner. 2020. Obtaining faithful interpretations from compositional neural networks. In ACL.

Jihao Shi, Xiao Ding, Li Du, Ting Liu, and Bing Qin. 2021. Neural Natural Logic Inference for Interpretable Question Answering. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3673–3684, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022. Fairr: Faithful and robust deductive reasoning over natural language. In Annual Meeting ofthe Association for Computational Linguistics (ACL).

Soumya Sanyal, Zeyi Liao, Xiang Ren. 2022. ROBUSTLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. *Computing Research Repository*, arXiv:2205.12598.

Shane Storks, Qiaozi Gao and Joyce Yue Chai. 2019. Recent advances in natural language inference: A survey of benchmarks, resources, and approaches. arXiv preprint arXiv:1904.01172.

Asher Stern, Shachar Mirkin, Eyal Shnarch, Lili Kotlerman, Ido Dagan, Amnon Lotan and Jonathan Berant. 2011. Knowledge and Tree-Edits in Learnable Entailment Proofs. In TAC. 2011.

Stuart Russell and Peter Norvig. 2010. Artificial intelligence: a modern approach, 3 edition. Prentice Hall.

Swarnadeep Saha, Prateek Yadav, and Mohit Bansal. 2021. multiPRover: Generating multiple proofs for improved interpretability in rule reasoning. In NAACL.

Swarnadeep Saha, Sayan Ghosh, Shashank Srivastava, and Mohit Bansal. 2020. PRover: Proof generation for interpretable reasoning over rules. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 122–136, Online. Association for Computational Linguistics.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Is Graph Structure Necessary for Multi-hop Question Answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages. 7187–7192, Online. Association for Computational Linguistics.

Changzhi Sun, Xinbo Zhang, Jiangjie Chen, Chun Gan, Yuanbin Wu, Jiaze Chen, Hao Zhou, and Lei Li. 2021. Probabilistic graph reasoning for natural proof generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3140–3151, Online. Association for Computational Linguistics.

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

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2022. Entailer: Answering Questions with Faithful and Truthful Chains of Reasoning. *arXiv preprint arXiv:2210.12217*.

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

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through logicNLI. *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738-3747, Online and Punta Cana, Dominican Republic. https://doi.org/10.18653/v1/2021.emnlp-main.303.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In NeurIPS.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers),* pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Siyuan Wang, Wanjun Zhong, Duyu Tang, Zhongyu Wei, Zhihao Fan, Daxin Jiang, Ming Zhou, and Nan Duan. 2022. Logic-driven context extension and data augmentation for logical reasoning of text. *In Findings of the Association for Computational Linguistics: ACL 2022*, pages 1619-1629, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.findings-acl.127.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, and Nan Duan. 2021a. From LSAT: The progress and challenges of complex reasoning. arXiv preprint arXiv:2108.00648.

Siyuan Wang, Zhongkun Liu, Wanjun Zhong, Ming Zhou, Zhongyu Wei, Zhumin Chen, Nan Duan. 2022. From LAST: The progress and challenges of complex reasoning. *In IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pages 2201-2216. https://doi.org/10.1109/TASLP.2022.3164218.

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

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

Fangzhi Xu, Jun Liu, Qika Lin, Yudai Pan, Lingling Zhang. 2022. Logiformer: A two-branch grph transformer network for interpretable logical reasoning. *In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1055-1065. https://doi.org/10.1145/3477495.3532016.

Kaiyu Yang, Jia Deng, Danqi Chen. 2022. Generating Natural Language Proofs with Verifier-Guided Search. *In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.*

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le.

2019. Xlnet: Generalized autoregressive pretraining for language understanding. In NeurIPS.

Weihao Yu, Zihang Jiang, Yanfei Dong, Jiashi Feng. 2020. ReClor: A reading comprehension dataset requiring logical reasoning. *In Proceedings of 8th International Conference on Learning Representations*, Addis Ababa, Ethiopis.

A Model Training and Parameters

Baseline Models

Initially, we performed evaluation on LogicNLI dataset (Tian, J. 2021). LogicNLI dataset contains different section: facts, rules statements and labels. We have used train/dev/test 16000/2000/2000 examples for our models. For baseline experiments, we have re-implemented the fine-tuned BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) base version and used [CLS] facts rules [SEP] statement [SEP] as input to the transformers to predict the logical relation. BERT uses 12-layer, 768-hidden, 12-heads, 110M parameters for base version and RoBERTa uses 123M parameters.

Our models are trained end-to-end using AdamW optimizer with the decay rate of 0.9. In addition, we have experimented with different learning rates to understand if there is any change in performance. However, learning rate of 5e-6 shows a steady linear increase with the specified decay rate for RoBERTa model. Hence, we have retained the similar hyper-parameters as mentioned in the LogicNLI dataset (Tian, J. 2021), for our BERT and RoBERTa base version. RoBERTa performs better than BERT base and shows 59% on the validation set and 57% on the test set.

The hyper-parameters are listed in the Table A-1.

Parameters	BERT	RoBERTa	XLNET
batch size	16	16	16
lr	5e−6	5e−6	5e−6
decay rate	0.9	0.9	0.9
l2 coeff.	1e−5	1e−5	1e−5
early stop	5	5	5
epochs	20	20	20
optimizer	AdamW	AdamW	AdamW

Table A-1 Hyperparemeters for Experiments

Logic Reasoning Model

Rule Inference We apply T5 (Raffel et al., 2020) as the encoder-decoder model to generate new facts given the input facts and rule. Given labeled reasoning chains in the LogicNLI dataset (Tian et al., 2021), it is not straight forward to train the model as they provide only 'positive' examples. We build our training set as follows. Given a training instance, we use the logic representation of the facts and rules and apply every rule expression on the fact expressions to generate 1-step inference with an off-the-shelf logic reasoner. The inferred facts are converted to natural language using a simple rule-based technique. The natural language versions of the source rules and facts are extracted from the dataset, and a training set is prepared using the processed input as '*question: <rule> context: <known facts>*' and the output '*inferred facts*', if the rule can be triggered and 'None' otherwise.

During training, we set number of beams as 50 and number of returned sequences as 5. We randomly split the 9600 instances into 80% training and 20% test for 5 times and report the average performance. We measure accuracy using the exact matching ratio.

As the input size depends on the facts, which may grow over multiple iterations, there is an impact on the token size limitations. We analyze the instances and find that the average size of each instance 191.758 tokens (Min: 171; Max: 240). Our current T5 model can handle sequences with up to 512 tokens. Assuming the worst case (Max size 240; 4 tokens/fact), the chaining process can go up to depth=68, before the limit is reached. We argue that this is sufficiently large for LogicNLI.dataset. For inference/real world examples, working with documents greater than 512 size, we can chunk the document (facts/rules) and use Roberta to encode each chunk accordingly.

Facts Checking We adopt RoBERTa (Liu et al., 2019) base version and used [CLS] facts [SEP] hypothesis [SEP] as input to the transformers to predict the inference relation. The hyper-parameters are as in the Table A-1.

Facts:

(1) Tim is fast.
(2) John is tall.
(3) John is not lean.

Rules:

(1) If someone is fast or tall, then he is athletic.
(2) Someone is fast if he is tall and he is lean.
(3) All those who are athletic will be not slow.
(4) If someone is lean and tall, then he is fast, and vice versa.

Hypothesis: John is not slow.

Label: Entailment

Figure A-1: Sample Instance for Illustration

Machine Configuration

For baseline models, initially we have used NVIDIA-GeForce RTX 2080 series with eight cores of GPU machines for all our experiments. Later, in order to train t5 large models, we have used NVIDIA-GeForce Tesla V100 series SXM2-32GB with 5 cores of GPU machines. Models were trained for 3-5 hours for training and reasoning.

B Supplementary Material

Sample Instance for Illustration

Figure A-1 presents a sample instance for illustration.

Algorithm pseudocode

Figure A-2 provides the full pseudocode of our algorithm outlined in Section 3.

Illustration of Output

Figure A-3 presents an illustration of the output by algorithm Chainformer.

Additional Experiments

Analysis of Inference Relations (Table A-2)

Here, we present the detailed reasoning performance for the four labels. 'Entailment' and 'Contradiction' performance were similar. 'Paradox' was the toughest (F1=74.4) among all. It had a high precision but low recall, as two reasoning chains are required for its classification. In contrast, 'neutral' had a lower precision but higher recall since most of the missed hypotheses will be labeled thus.

Algorithm Chainformer

Input: F, Facts; R, Rules; h, Hypothesis
Output: Inference ∈ {E,C,N,P}; X, the reasoning chain.
F' ← F
C ← φ, the empty set
Repeat
  R' ← *Shuffle(R)*
  For r ∈ R'
    InferredFacts ← *RuleInference*(r,F')
    NewFacts ← InferredFacts – F'
    If NewFacts is not φ
      F' ← F' ∪ NewFacts
      C ← C ∪ (r, NewFacts)
    End
  End
Until no new facts added to *F'*
Inference ← *FactsChecking*(h,F')
Reasoning Chain X ← *AssembleChain*(h,Inference,C)

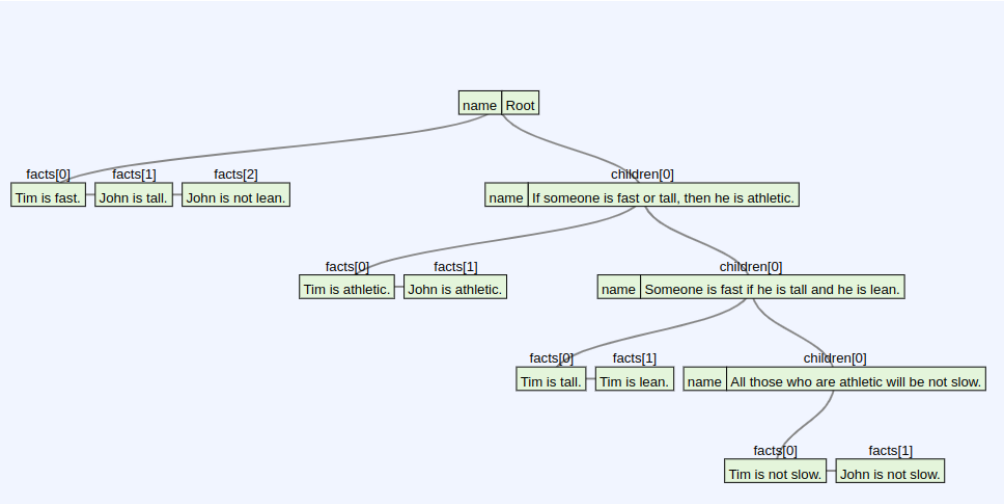
Figure A-2 Algorithm Chainformer Pesudocode

Labels	Test		
	P	R	F1
Contradiction	82.7	81.6	82.1
Entailment	81.3	82.0	81.6
Neutral	75.0	92.6	82.9
Paradox	86.0	65.6	74.4

Table A-2: Analysis of Inference Relations

Performance over FOLs

LogicNLI dataset tags over 23 classes of FOLs. Each class is named using an abbreviation of the rule members as below. Given a rule, we denote the FOL connectives viz. conjunction ∧ (**C**), disjunction ∨ (**D**), implication → (**I**), equation ≡ (**Q**), universal quantifier ∀ (**A**), and existential quantifier ∃ (**E**), with a letter (bracketed), and concatenate their respective letters in the order they appear in the rule. For example, Rule 4 in Figure 4 would belong to the class 'ACQ'. The accuracy results of all classes are presented in Table A-3.



Inference : Entailment

Reasoning Chain:

Tim is not slow.
Tim is fast.
Tim is athletic.
John is tall.
John is not slow.

Figure A-3: Sample Output of Chainformer with for instance in Figure A-1.

FOL Class	Accuracy
CIC	100.0%
CQC	100.0%
EDI	97.3%
EIC	97.1%
EI	92.9%
CE	92.9%
EC	91.7%
ADI	91.4%
ECI	90.9%
CQ	90.0%
ACI	90.0%
AQ	89.3%
EDIC	88.9%
I	85.2%
Q	81.5%
ECIC	80.0%
AI	80.0%
AIC	75.6%
ACIC	75.0%
DI	73.9%
ACQ	44.8%
ACQC	41.7%
ADIC	0.0%

Table A-3 Performance over FOLs

Analysis of FOLs with Conjunction (Table A-4)
We also analyzed the accuracy over FOLs with conjunctions in implication rule (before and after →) and similarly in rules with equivalence. Results imply that conjunctions in the consequent are harder for implications. In case of equivalence, it is even harder possibly the implication works both ways.

Sample Instance from LogicNLI
Figure A-4 presents a sample instance from LogicNLI as an illustration.

	Analysis of Conjunctions			
	bef →	aft →	bef ≡	aft ≡
# of FOLs	5	7	4	2
Accuracy	87.8	73.8	69.1	70.8

Table A-4. Analysis of Conjunctions

Facts: (F1) Pierce is not crazy.(F2) Norman is breakable.(F3) Travis is not terrible.(F4) Alfred is terrible.(F5) Norman is crazy.(F6) Norman is difficult.(F7) Pierce is difficult.(F8) Kerry is cautious.(F9) Pierce is not cautious.(F10) Alfred is not breakable.(F11) Travis is not breakable.(F12) Kerry is careful.

Rules: ((R1) Norris being not cautious implies that Norman is not crazy. (R2) If someone is not difficult or he is terrible, then he is cautious. (R3) Pierce being breakable implies that Norman is difficult. (R4) If Travis is crazy, then Alfred is cautious and Pierce is not difficult. (R5) As long as someone is either not difficult or breakable, he is terrible and not careful. (R6) As long as someone is difficult, he is not crazy and breakable. (R7) If there is at least one people who is not breakable, then Kerry is careful. (R8) If Kerry is not careful and Alfred is cautious, then Pierce is not difficult. (R9) If there is someone who is both crazy and terrible, then Alfred is careful. (R10) If someone is not cautious, then he is crazy. (R11) If someone is terrible or not cautious, then he is difficult. (R12) If there is at least one people who is both not breakable and crazy, then Kerry is careful.

Statement: Pierce is not careful.

Label: entailment

Reasoning Path: [[FACT(7)--> RULE (6)]--> RULE(5)]

Statement: Norman is careful.

Label: contradiction

Reasoning Path: FACT(2) --> RULE (5)

Figure A-4: An instance from LogicNLI,dataset showing facts, rules, Statements, reasoning paths and labels.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 5.1

☒ A2. Did you discuss any potential risks of your work?
No potential risks anticipated.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☐ Did you use or create scientific artifacts?

Not applicable. Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Section 4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix A

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.