Abstract-level Deductive Reasoning for Pre-trained Language Models

Xin Wu[1,2], **Yi Cai**[1,2,*], **Ho-fung Leung**[3]
[1]School of Software Engineering, South China University of Technology
[2]Key Laboratory of Big Data and Intelligent Robot
(South China University of Technology) Ministry of Education
[3]Independent Researcher
sexinw@mail.scut.edu.cn, ycai@scut.edu.cn, ho-fung.leung@outlook.com

Abstract

Pre-trained Language Models have been shown to be able to emulate deductive reasoning in natural language. However, PLMs are easily affected by irrelevant information (*e.g.*, entity) in instance-level proofs when learning deductive reasoning. To address this limitation, we propose an Abstract-level Deductive Reasoner (ADR). ADR is trained to predict the abstract reasoning proof of each sample, which guides PLMs to learn general reasoning patterns rather than instance-level knowledge. Experimental results demonstrate that ADR significantly reduces the impact of PLMs learning instance-level knowledge (over 70%).

Keywords: Deductive Reasoning, Abstract-level Proof, Proof Generation

1. Introduction

A long-term goal of AI is to build systems that can automatically reason over a given context and generate logically valid conclusions (McCarthy et al., 1960; Helwe et al., 2021). Recently, Pre-trained Language Models (PLMs, *e.g.*, RoBERTa (Liu et al., 2019), T5 (Raffel et al., 2020)) are able to predict the logical validity of natural language deductions (Clark et al., 2021; Tafjord et al., 2021; Sanyal et al., 2022b). For example in Figure 1, given a deduction about whether Gary is nice, PLMs utilize the provided context to determine its validity and generate the reasoning proof. Endowing machines with deductive reasoning abilities brings immense potential for downstream applications such as question answering (Yang et al., 2018; Dalvi et al., 2021), and argument mining (Habernal and Gurevych, 2017).

Existing methods (Tafjord et al., 2021) fine-tune PLMs using the validity label and instance-level reasoning proofs as supervision signals. This approach easily leads PLMs to learn narrow instance-level knowledge without learning general reasoning patterns between instances, making it difficult to generalize to out-of-distribution samples. For example in Figure 1 (a), in the fine-tuning set, the labels of samples containing "Gary is nice" are all valid. PLMs may learn the instance-level knowledge that there is a mapping relationship between "gary is nice" and the label "valid". This leads PLMs to incorrectly determine the invalid sample "Gary is smart. Smart things are **not** nice. Therefore, Garry is nice." as valid. In contrast, humans possess the ability to abstract information unrelated to reasoning, thus learning general reasoning patterns (Marcus and Davis, 2020). As illustrated in Figure 1 (b), the information unrelated to reasoning in "Gary is smart. Smart things are not nice. Therefore, Garry is nice." is abstracted into symbols as "A is B, B people are not C. Therefore, A is not C." This corresponds to an invalid pattern, so this sample should be considered invalid. We consider that utilizing abstract reasoning proof as supervision signals can compel PLMs to focus on the reasoning patterns when learning deductive reasoning, thereby avoiding the acquisition of instance-level knowledge unrelated to reasoning.

Additionally, existing methods (Tafjord et al., 2021) adopt a post hoc manner to generate reasoning proofs, wherein validity labels are first generated followed by the generation of reasoning proofs. When the predicted labels are incorrect, the proofs generated by PLMs are also incorrect. For instance, as depicted in Figure 1 (a), PLMs incorrectly predict "Gary is smart. Smart things are not nice. Therefore, Garry is nice." as valid based on instance-level knowledge. Then they generate incorrect reasoning proofs to "support" the incorrect prediction by using a nonexistent rule ("if someone is smart then it is nice"). We consider having PLMs generate abstract-level proof first, and then generating validity predictions based on the abstract-level proof. Because abstract-level proof is composed of abstracted symbols and does not include specific entities and predicates, it can avoid interference from instance-level knowledge. Additionally, the abstract-level proof generated first will serve as context to constrain PLMs in predicting correct labels.

To address the above limitations, we propose an Abstract-level Deductive Reasoner (ADR). Specifically, ADR symbolizes the entities and predicates in instance-level reasoning proofs to obtain abstract

*Corresponding authors

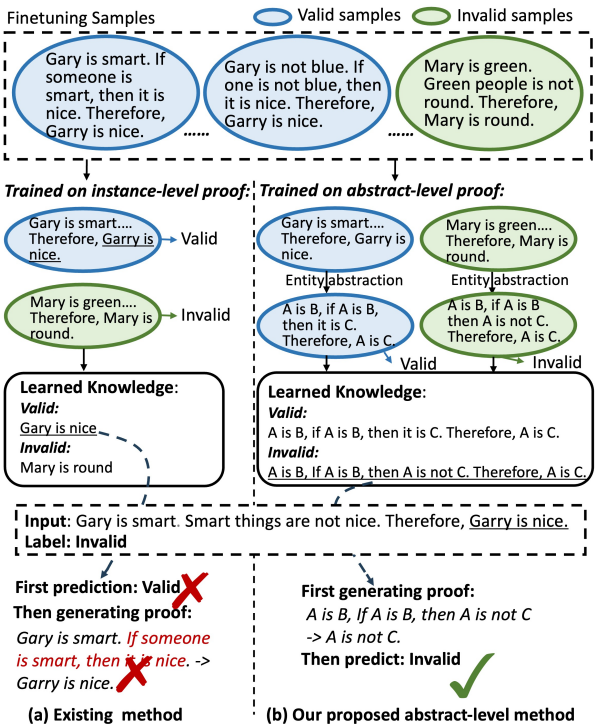

Figure 1: Comparison of (a) existing methods based on instance-level proof and (b) our proposed method based on abstract-level proof.

reasoning proofs. ADR utilizes logical validity labels and abstract reasoning proofs as supervised signals for training. This approach prevents PLMs from learning instance-level knowledge that may affect their generalization in deductive reasoning. Additionally, to address the issue of PLMs fabricating reasoning proofs, ADR adopts a reason-and-decide manner that generates reasoning proofs first and then generates validity labels. To better observe whether PLMs have learned instance-level knowledge, we propose to inject label-specific knowledge into the training data. This knowledge would only appear in one category of the training set (*e.g.*, valid) but appear in another category in the test set (*e.g.*, invalid). If PLMs make incorrect predictions, it can indicate that they have learned instance-level knowledge. We utilize this approach to construct an injection dataset and test the ADR and the baseline models on it.

The contributions of the paper are as follows:

- We show that PLMs are easily affected by irrelevant information in instance-level proofs when learning deductive reasoning, making them difficult to generalize to out-of-distribution deductive reasoning data.
- We propose to train PLMs for deductive reasoning by abstracting irrelevant information (*e.g.*, entity), enabling PLMs to learn general reasoning patterns across instances, and improving their robustness on deductive reasoning data from different distributions.
- We conduct extensive experiments to evaluate our proposed ADR. The results indicate that ADR significantly reduces the impact of PLMs learning instance-level knowledge (over 70%).

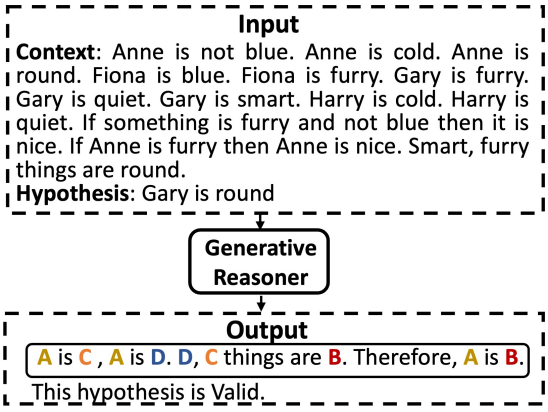

Figure 2: The overall architecture of ADR.

2. Method

2.1. Task Definition

In the deductive reasoning task, given a context C, the goal is to predict whether C can support a hypothesis H. Examples are shown in Figure 2. The label for each hypothesis can be either *Valid* (the hypothesis can be provably supported) or *Invalid* (the hypothesis can be provably unsupported).

2.2. Generative Reasoner

The input to the generative reasoner is of the form: "$hypothesis$ = HT ; $context$ = CT", where HT and CT are the text of hypothesis and context respectively. The output of the generative reasoner is in the form of: "ABP, so the answer is L". The ABP is the abstract reasoning proof, which is described in the following section. The L is the validity label of each sample, which is *Valid* or *Invalid*.

2.3. Obtaining Abstract Reasoning Proof

We propose to use the abstract reasoning proof to guide the model fine-tuning. In abstract reasoning proof, entities and predicates are replaced with meaningless symbols, thus forcing the model to learn the reasoning pattern rather than instance-level knowledge. Given an instance-level proof, we replace the fact and rule indicators with their corresponding text to obtain the instance-level proof (*e.g.*, "Gary is furry. Gary is smart. Smart, furry

things are round. Gary is round"), there are three steps to obtain the abstract reasoning proof: **Extracting.** We extract the entities and predicates in the instance-level proof. **Mapping.** We map each entity and predicate with a letter symbol from a symbol list $[A, B, C, ..., Z]$. **Replacing.** We replace each entity and predicate with their letter symbols in the instance-level proof.

For example in Figure 2, we firstly extract "Gary", "furry", "smart", and "round" in the instance-level proof. Then we map them into "A", "B", "C", and "D" respectively. Finally, we obtain abstract reasoning proof "A is C. A is D. D, C things are B, so A is B".

3. Label-specific Knowledge Injection

We construct an instance-level knowledge injection dataset (injection dataset for short) to evaluate whether PLMs have learned label-specific instance-level knowledge. The core idea is to inject label-specific instance-level knowledge into training samples of each label (*e.g.*, valid and invalid) and then inject this instance-level knowledge into test samples of different labels. If the PLMs learn this label-specific instance-level knowledge during training, they will perform poorly on the test set. In contrast, if the models learn general reasoning patterns, their test performance will be robust. In this paper, we focus on injecting instance-level knowledge related to entities and predicates. Specifically, we use the same entities and predicates when constructing the valid samples of the training set and the invalid samples of the test set; and use the same entities and predicates when constructing the invalid samples of the training set and the valid samples of the test set. If the PLMs learn instance-level knowledge related to entities or predicates during training, they will perform poorly on the test set.

For example, the sentences "Gary is nice" and "The cat needs the cow" only appear in the "valid" samples of the training set. The model may learn a spurious association between these sentences and the "valid" label. When these two sentences appear in "invalid" samples in the test set, the model may make incorrect prediction.

4. Experiments and Results

4.1. Experimental Setup

Datasets: We conduct experiments on the RuleTaker (Clark et al., 2021; Tafjord et al., 2021) deductive reasoning dataset (Original dataset for short) and the constructed injection dataset. Each dataset contains 56,512 samples (39,864 for training, 5,272 for development, and 11,376 for testing). The training, development, and test set are all with an equal balance of valid and invalid samples.

Table 1: The test set accuracy of each model on the original and injection dataset.

	Original	Injection
RoBERTa-large	99.8	0.0
T5	99.1	0.0
ProofWriter	99.7	10.0
ADR (Ours)	**99.9**	**80.9**

Baselines: **RoBERTa**: We follow (Clark et al., 2021) who adopt a RoBERTa-large (Liu et al., 2019) as the baseline. According to their experiments, the RoBERTa-large is additionally fine-tuned on the RACE dataset, and then fine-tuned on the reasoning dataset. **T5**: We contain facts, rules, and the question as the input, and use the label as the output. **ProofWriter**: ProofWriter uses the classification label and Polish Notations proof with sentence identifiers as the output (Tafjord et al., 2021).

Setup: The input length of each model is set to 512, the batch size is set to 4. The learning rate of the RoBERTa-large is set to 1e-5, and the learning rate of the other T5-based models (*i.e.*, T5-base, ProofWriter, ADR) is set to 1e-3. All the models are trained on a V100. We follow the (Clark et al., 2021) to use the classification accuracy as the metric. The decoding parameter for all models is set to a greedy search with a maximum length of 512. Formally, the training data for T5 consists of (X, Y) pairs. The purpose of ADR is to modify the original training data Y, denoted as f(Y). This transforms the T5 training data into (X, f(Y)) pairs.

4.2. Overall Performance

The results of all tested models are shown in Table 1. Firstly, we observe that all the tested models can achieve >99% accuracy on the original dataset, which demonstrates their capability of learning deductive reasoning. However, when training models on the injection dataset, the performance of the RobERTa-large and T5-base drops to 0%. This indicates that these two models have learned instance-level knowledge. Models that are only trained with validity label is prone to learn instance-level knowledge. In contrast, models trained with proof supervision (*i.e.*, ProofWriter, and ADR) can alleviate the problem. However, the accuracy of the ProofWriter still drops to 10%, which drops nearly 90%. This indicates that simply using the instance-level proof is not enough to train a robust reasoner. ProofWriter still learned the instance-level knowledge. Compared with all the baselines, ADR achieves the best and the most robust performance on both the original and injection datasets. This indicates that using abstract reasoning proofs prevents PLMs from learning narrow instance-level knowledge.

Table 2: The ablation results of ADR.

Models	Abstract proof	Reason-and-decide	Accuracy
ADR	✓	✓	80.9
(a)	✓		10.0
(b)		✓	19.5

Table 3: The results on human-written dataset.

Models	Human-written Dataset
RoBERTa-large	2.4
T5	1.0
ProofWriter	29.8
ADR (Ours)	87.5

4.3. Ablation Study

Given a context C and a hypothesis H, the label L for whether C supports H is $Valid, Invalid$. Additionally, the instance-level proof for L is P_i, and the abstract-level proof is P_a. The training data for ADR consists of pairs (X, Y), where X is $[C, H]$ and Y is $[P_a, L]$. Here, $[a, b]$ indicates concatenating a and b in order. Therefore, in the ablation experiments, the training data Y for models (a) and (b) correspond to $[L, P_a]$ and $[P_i, L]$, respectively.

Table 2 shows the results of the ablation experiments on abstract reasoning proofs and the reason-and-decide manner. We make the following observations: (1) The reason-and-decide manner is crucial for ADR. When not using the reason-and-decide manner (using post hoc manner instead), ADR's performance dropped from 80.9% to 10.0% (compare ADR and (a)). During training, PLMs first learn to predict validity labels and then learn proof generation. In this situation, PLMs are prone to learning instance-level knowledge, which prevents abstract reasoning proofs from influencing the model's learning process. (2) Abstract reasoning proofs are also crucial for ADR. When using instance-level reasoning proof, ADR's performance dropped to 19.5% (Compare ADR and (b)). This indicates that instance-level reasoning proof fails to reveal general reasoning patterns, leading PLMs to learn instance-level knowledge.

4.4. Qualitative Analysis

We manually evaluate proof generated by ADR and ProofWriter. We observe that ProofWriter often generates proofs unrelated to the hypothesis. For example, given the hypothesis "warrigal is enlivened," ProofWriter generates a proof related to the "kaffir cat": "kaffir cat is automotive, kaffir cat is germy. And germy, automotive things are syphilitic. So kaffir cat is syphilitic." This is because ProofWriter has learned instance-level knowledge that "warrigal is not enlivened" during training and determines that the hypothesis "warrigal is enlivened" is invalid, leading to the generation of an unrelated proof. In contrast, ADR, by abstracting entities and predicates during training, can avoid the influence of instance-level knowledge and generate correct proofs. Moreover, we observe that the proof generated by ADR may differ from the ground truth but is still logically valid. This suggests that ADR contributes to learning deductive reasoning.

4.5. Knowledge Injection Percentage

We test the performance of ADR and ProofWriter at different levels of injection. Specifically, we construct training and test sets with different injection levels by controlling the percentage of injection samples. We observe that at injection percentage of 0%, 50%, 90%, and 100%, the accuracy of ProofWriter is 99.1%, 96.6%, 85.0%, and 10.0%, respectively, while the accuracy of ADR is 99.9%, 96.8%, 90.7%, and 80.9%, respectively. The improvements brought by ADR are 0.2%, 0.2%, 5.7%, and 70.9%, respectively. This indicates that both the model's performance and the improvements brought by ADR are directly proportional to the percentage of injection samples. Therefore, the more instance-level knowledge that exists in a dataset, the more improvements ADR can bring.

4.6. Human-written Deductive Reasoning

We also evaluate ADR and the baseline models on the human-paraphrased version of Ruletaker (Clark et al., 2021) dataset. We use the same injection procedure to inject label-specific knowledge into the dataset. The results are illustrated in Table 3. We make the following observations: (1) On the human-written dataset, RoBERTa-large, T5, and Proofwriter also easily learn instance-level knowledge, which leads to their overall performance being unsatisfactory. (2) On the contrary, ADR is still more robust and outperforms the baseline models. This indicates that ADR can be applied to human-written deductive reasoning. (3) Proofwriter and ADR both achieve better results on human-written data compared to synthetic data. One possible reason is that the diversity of natural language expressions increases, which also makes it more challenging for the model to learn instance-level knowledge. This results in the model being more inclined to learn general reasoning patterns.

5. Related Work

Previous works on deductive reasoning are mostly based on formal language (Musen and Van der Lei, 1988; Metaxiotis et al., 2002; Paulin-Mohring, 2011;

Phillips and Stanovskỳ, 2008), which cannot be directly applied to the daily used natural language. Some prior works parse natural language into formal forms (Martínez-Gómez et al., 2016; Suzuki et al., 2019; Kamath and Das, 2018; Wu et al., 2023), then apply formal reasoner to perform logical reasoning. Recently, Pre-trained language models have achieved impressive performance across multiple tasks (Bu et al., 2022; Yuan et al., 2022; Bu et al., 2023). (Clark et al., 2021) proposes a natural language deductive reasoning task and dataset called RuleTaker. They find that transformer-based (Vaswani et al., 2017) pre-trained RoBERTe-large (Liu et al., 2019) are able to emulate deductive reasoning on natural language. The follow-up work (Tafjord et al., 2021) refines the RuleTaker and proposes proof generation tasks. They use the T5 (Raffel et al., 2020) instead of RoBERTa as the neural reasoner. And several neural reasoners (Sanyal et al., 2022b; Liang et al., 2021; Yang et al., 2022) are proposed to perform deductive reasoning, which is mostly based on T5 or RoBERTa. Recent works evaluate the robustness of neural reasoners from logic operators (Sanyal et al., 2022a) and logic consistent adversarial attacks (Gaskell et al., 2022) in two respects. Compared with their works, we focus on the issue of instance-level knowledge, which has not been discussed in previous work.

The Selection-Inference method (Creswell et al., 2022) decouples selection and inference, generating proofs through multiple iterations rather than providing all proofs at once. Additionally, the Selection-Inference method employs in-context learning for both selection and inference, which largely avoids instance-level knowledge during training and fine-tuning processes. However, due to the absence of fine-tuning, the Selection-Inference method may have lower performance for specific tasks compared to methods like ProofWriter and ADR. Moreover, research indicates (Si et al., 2023) that in-context learning may be influenced by bias present in the demonstration. Therefore, we believe future research could explore combining ADR methods with in-context learning in the Selection-Inference approach.

6. Conclusion

In this paper, we reveal that abstract reasoning proofs can effectively alleviate the issue of PLMs learning undesired instance-level knowledge when learning deductive reasoning. Additionally, we propose to generate proofs first and then determine their validity, which plays a significant role in producing valid proofs. It is easy to learn instance-level knowledge even with ADR without this manner.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (62076100), Fundamental Research Funds for the Central Universities, SCUT (x2rjD2230080), the Science and Technology Planning Project of Guangdong Province (2020B0101100002), Guangdong Provincial Fund for Basic and Applied Basic Research - Regional Joint Fund Project (Key Project) (2320191025000318,308155351064), CAAI-Huawei MindSpore Open Fund, CCF-Zhipu AI Large Model Fund.

Limitations

ADR currently still needs abstract reasoning proofs for fine-tuning, which can be time-consuming. How to make the model autonomously discover reasoning patterns in the data is also a problem worthy of research. Especially when many large language models currently only provide API interfaces, how to equip these LLMs with robust deductive reasoning capabilities is also worth investigating.

References

Yuqi Bu, Liuwu Li, Jiayuan Xie, Qiong Liu, Yi Cai, Qingbao Huang, and Qing Li. 2022. Scene-text oriented referring expression comprehension. *IEEE Transactions on Multimedia*.

Yuqi Bu, Xin Wu, Liuwu Li, Yi Cai, Qiong Liu, and Qingbao Huang. 2023. Segment-level and category-oriented network for knowledge-based referring expression comprehension. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8745–8757.

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

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. 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. In *Proceedings of the 2021 Conference on Empirical Methods*

in Natural Language Processing, pages 7358–7370.

Alexander Gaskell, Yishu Miao, Francesca Toni, and Lucia Specia. 2022. Logically consistent adversarial attacks for soft theorem provers. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 4129–4135. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

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*.

Aishwarya Kamath and Rajarshi Das. 2018. A survey on semantic parsing. In *Automated Knowledge Base Construction (AKBC)*.

Zhengzhong Liang, Steven Bethard, and Mihai Surdeanu. 2021. Explainable multi-hop verbal reasoning through internal monologue. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1225–1250.

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. *arXiv preprint arXiv:1907.11692*.

Gary Marcus and Ernest Davis. 2020. Insights for ai from the human mind. *Communications of the ACM*, 64(1):38–41.

Pascual Martínez-Gómez, Koji Mineshima, Yusuke Miyao, and Daisuke Bekki. 2016. ccg2lambda: A compositional semantics system. In *Proceedings of ACL-2016 System Demonstrations*, pages 85–90.

John McCarthy et al. 1960. *Programs with common sense*. RLE and MIT computation center Cambridge, MA, USA.

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

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

Christine Paulin-Mohring. 2011. Introduction to the coq proof-assistant for practical software verification. In *LASER Summer School on Software Engineering*, pages 45–95. Springer.

JD Phillips and David Stanovskỳ. 2008. Automated theorem proving in loop theory. In *Proceedings of the CICM Workshop on Empirically Successful Automated Reasoning in Mathematics*, 378, pages 42–54.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022a. RobustLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9614–9631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Soumya Sanyal, Harman Singh, and Xiang Ren. 2022b. Fairr: Faithful and robust deductive reasoning over natural language. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1075–1093.

Chenglei Si, Dan Friedman, Nitish Joshi, Shi Feng, Danqi Chen, and He He. 2023. Measuring inductive biases of in-context learning with underspecified demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11289–11310.

Riko Suzuki, Hitomi Yanaka, Masashi Yoshikawa, Koji Mineshima, and Daisuke Bekki. 2019. Multimodal logical inference system for visual-textual entailment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 386–392.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xin Wu, Yi Cai, Zetao Lian, Ho-fung Leung, and Tao Wang. 2023. Generating natural language from logic expressions with structural representation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022. Generating natural language proofs with verifier-guided search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 89–105, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2022. Hierarchical template transformer for fine-grained sentiment controllable generation. *Information Processing and Management*, 59(5):103048.