Towards Grounded Natural Language Proof Generation

Anonymous Author(s)
Affiliation
Address
email

Abstract

When a student is working on a mathematical proof, it is often helpful to receive suggestions about how to proceed. To this end, we provide an initial study of two generation tasks in natural mathematical language: suggesting the next step in a proof, and full-proof generation. As proofs are grounded in past results– e.g. theorems, definitions– we study knowledge-grounded generation methods, and find that conditioning on retrieved or ground-truth knowledge greatly improves generations. We characterize error types and provide directions for future research.

1 Introduction

Proving a mathematical claim involves constructing an argument that is grounded in knowledge from past results, including previously proved theorems, established definitions and equations, as well as similarly structured arguments. Moreover, proofs require multi-step, logically consistent arguments, rather than extracting a span of text, producing an abstractive summary, or answering a question, thus complimenting existing knowledge-intensive tasks studied in natural language processing.

We provide a preliminary study of two proof generation tasks using the recently released NATURALPROOFS dataset [Welleck et al., 2021]: *next-step suggestion*, where a model generates the next step of the proof, given a statement and the preceding proof steps, and *full-proof generation*, where a model generates a complete proof. The next-step setting is motivated by educational applications, such as a student querying the system for hints, and is inspired by ML-based tactics in interactive (formalized) proof assistants, such as the `gpt-f` tactic [Han et al., 2021]. Full proof generation presents a challenging, long-form generation task. As each proof contains statements that are grounded in past results– theorems, definitions, etc – we empirically study various knowledge-grounded generation methods, and provide baseline task definitions and metrics.

We find that conditioning on reference information substantially improves the quality of generated proofs. We provide results with knowledge from retrieved references, which show modest but nontrivial gains over baselines that rely on parametric, closed-book knowledge, and results with knowledge from ground-truth references that yield substantial improvements. We inspect and characterize the generations, finding that even with access to ground-truth knowledge, models can produce mathematically incorrect statements (e.g. generating $\ln x = \frac{1}{x}$ without writing $\frac{d}{dx}$), hallucinate references, and produce proofs that are shorter than those written by humans, leaving room for progress on improving and automatically evaluating machine-generated mathematics.

2 Tasks

For both of our tasks we use the NATURALPROOFS dataset, a multi-domain corpus of theorems, definitions, and proofs in natural mathematical language. We use the PROOFWIKI do-

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021) Workshop on Math AI for Education (MATHAI4ED). Do not distribute.

main of NATURALPROOFS in our experiments, which provides broad coverage of predominantly undergraduate-level mathematics. NATURALPROOFS pairs theorems $\mathbf{x}$ and their proofs $\mathbf{y}$, where $\mathbf{x}$ and $\mathbf{y}$ are variable-length token sequences. Each proof contains references to previous definitions, theorems, or other pages from a reference set $\mathcal{R}$. For instance, the proof of `Equivalence Class is not Empty` contains references to `Equivalence Class` and `Empty Set`. Refer to Welleck et al. [2021] for further details about NATURALPROOFS.

Next-step suggestion. When a student or researcher is working on a proof, it is often helpful to receive suggestions about how to proceed. We envision a system in which a user scans a list of suggested next-steps, analogous to machine-learning based suggestions in interactive theorem provers. Next-step suggestion is the related task of suggesting a set of next steps $\{y_t^{(k)}\}_{k=1}^K$ of a proof, given a theorem statement $\mathbf{x}$ and preceding steps $y_{<t}$. This task assumes that each proof is segmented into a variable number of steps, $\mathbf{y} = (y_1, \ldots, y_T)$, where each step is a variable-length token sequence $y_t = (w_1, \ldots, w_L)$, which is the case for the PROOFWIKI domain of NATURALPROOFS. Next-step suggestion is analogous to next-utterance prediction in dialogue modeling, the task of predicting the next turn of a conversation (e.g. Zhang et al. [2018]). In this setting, dialogue models use ground-truth conversation histories, analogous to using ground-truth proof histories.

Full proof generation. Full proof generation is the task of generating a full proof $\mathbf{y} = (y_1, \ldots, y_T)$ given a theorem statement $\mathbf{x}$. Naturally, a next-step suggestion model can be used for full proof generation, and vice-versa. Continuing the analogy with dialogue, a next-utterance model can be evaluated in a full dialogue setting, in which the conversation history consists of model generations.

2.1 Evaluation metrics.

Evaluating proofs that are generated in natural mathematical language is challenging, as there is not direct access to a verifier. Nevertheless, existing knowledge-intensive generation tasks in NLP face a similar state-of-affairs: for instance, the KILT benchmark [Petroni et al., 2021] relies on F1-score for evaluating knowledge-grounded dialogue [Dinan et al., 2019] and ROUGE for long-form QA [Fan et al., 2019]. Like our setting, these tasks also typically rely on a single ground-truth output, despite there being many valid possibilities. Here, we use simple metrics for evaluating the content and knowledge grounding in generated proofs. An interesting research direction is automatically evaluating machine-generated mathematics to improve upon the foundation that we establish here.

Lexical content metrics. To evaluate the language modeling quality of each model, we use held-out perplexity. To compare each generated proof against its ground-truth counterpart, we use sentence-BLEU, METEOR, and edit distance.

Knowledge grounding metrics. We define knowledge grounding as meaning that a generated proof contains the same references as those found in the ground-truth proof, as measured by precision, recall, and F1 score between the reference sets contained in the generated and ground-truth proofs.

Multiple candidate evaluation. The next-step setting simulates providing a user with suggestions about which step to take next in a proof. Since the user is free to choose from among the suggestions, we measure metrics using the best per-example metric out of each suggestion set.

3 Methods

As our base conditional language model we use BART [Lewis et al., 2020a], an encoder-decoder model pretrained with denoising tasks on natural language text. We consider two paradigms for enabling the model to ground its generations in knowledge contained in references.

Intermediate denoising training. Neural language models have demonstrated an ability to act as knowledge bases, generating facts when suitably prompted [Petroni et al., 2019]. To see whether this impacts knowledge-grounded proof generation, we train a model to denoise training theorems, proofs, and references, with various denoising tasks. Intuitively, the model can store reference knowledge and deploy it at generation time. This paradigm is limited, as the model cannot access non-training references and is not told which references it should use to ground its generations at test-time.

Knowledge-grounding methods. Various methods augment the model's input with sequences containing knowledge used to ground generations. We use Fusion-in-Decoder [Izacard and Grave, 2020], an encoder-decoder architecture that encodes reference-augmented inputs,

		Lexical				Grounding		
Model	**Knowledge**	**SBleu-2**	**SBleu**	**Meteor**	**Edit**	**Prec**	**Rec**	**F1**
BART	–	6.84	3.75	12.55	94.02	18.14	9.63	10.96
BART-FID	Retrieved refs (8)	8.21	4.37	14.10	94.49	21.07	13.04	13.79
BART-FID	Retrieved refs (20)	6.31	3.58	11.75	93.38	23.21	11.98	13.81
BART-FID	Ground-truth titles (20)	9.46	5.90	**15.99**	91.69	56.73	33.13	38.65
BART-FID	Ground-truth refs (8)	8.19	5.22	14.89	91.11	61.40	31.89	39.05
BART-FID	Ground-truth refs (20)	**9.54**	**6.10**	15.93	**91.00**	**63.54**	**33.90**	**41.00**

Table 1: Automatic metrics for full proof generation.

Theorem	**Co-Countable Measure is Probability Measure** Let X be an uncountable set. Let $\mathcal{A}$ be the σ-algebra of countable sets on X. Then the co-countable measure μ on X is a probability measure.
BART	Proof by induction: For all $n \in \mathbb{N}_{>0}$, let $P(n)$ be the proposition μ on X. $P(1)$ is the case: $\mu(1) = \bigcup_{n=1}^{\infty} \{1, 2, \dots\}$ which is trivially true, as $\mathcal{A}$ is a σ-algebra of countable sets. This is our piecewise disjoint axioms for the induction. Now we need to show that, if $g(j)$ is true, where $j \geq 1$, then it logically follows that $g_{j+1}(j+2) \geq g(j+1)$ for all j such that $1 \leq j \leq 1$, and hence $\mathrm{d}(j, j) = \mu(j, k)$! ...
BART-FID	By Co-Countable Measure is Measure, μ is a probability measure.
BART-FID+denoise	From Co-Countable Measure is Measure, μ is a measure. From Relative Complement with Self is Empty Set: $\mathrm{C}_S(X) = \emptyset$ Hence the result.
Ground-truth	By Co-Countable Measure is Measure, μ is a measure. By Relative Complement with Self is Empty Set, have $X^C(X) = \emptyset$. As $\emptyset$ is countable, it follows that X is co-countable. Hence $\mu(X) = 1$, and so μ is a probability measure.

Table 2: *Full-proof generation* example. The colors denote: Undefined term. Hallucinated reference. Non-ground-truth reference. Improper/irrelevant statement. Does not follow. Term appears in reference. Ground-truth reference.

$(\mathbf{r}_1, \mathbf{x}), (\mathbf{r}_2, \mathbf{x}), \dots, (\mathbf{r}_K, \mathbf{x})$, into a sequence of vectors that are attended to by the decoder. An interesting future work direction is providing references at decoding time rather than through the architecture [Lu et al., 2021]. We consider two settings for the references $\mathbf{r}_1, \dots, \mathbf{r}_K$: (i) ground-truth references, analogous to *document-grounded generation* tasks [Prabhumoye et al., 2021], and (ii) retrieved references, analogous to *retrieval-augmented generation* [Lewis et al., 2020b].

Decoding. For full proof generation, we use beam search, and for next-step suggestion we generate four suggestions: the top beam candidate, and 3 candidates sampled with ancestral sampling. Designing algorithms for multi-step proofs or better suggestion selection is interesting future work.

4 Experiments

Automatic metrics for full-proof generation are shown in Table 1. Providing the model with knowledge improved the lexical content and knowledge grounding in generated proofs. Providing retrieved references yielded modest improvements, while ground-truth references yielded large improvements of roughly 3 points on 2-gram sentence-bleu and 30 points on F1. Increasing the number of ground-truth references to 20 and using the full reference content versus only the title gave the best performance. As seen in Table 5, denoising results in lower perplexity, but without guaranteeing better generations. The example generation in Table 2 shows that providing knowledge can yield major differences in the subjective quality of generated content.

The next-step metrics (Table 3) indicate similar patterns, though without improved lexical metrics using retrieved references. Denoising improved lexical quality across all metrics, at a small cost to knowledge grounding. Altogether, the results indicate that grounding the model's generation through

Model	Knowledge	Lexical				Grounding		
		SBleu-2	SBleu	Meteor	Edit	Prec	Rec	F1
BART	–	12.34	7.62	14.42	86.63	6.02	5.24	5.32
BART-FID	Retrieved refs (8)	11.83	7.34	14.46	87.59	6.29	6.93	6.22
BART-FID	Ground-truth refs (8)	13.01	8.18	15.46	85.96	**14.70**	**13.82**	**13.62**
BART-FID+denoise	Ground-truth refs (8)	**13.48**	**8.45**	**16.41**	**85.77**	14.28	13.43	13.14

Table 3: Automatic metrics for next-step suggestion.

Proof History	By Co-Countable Measure is Measure, μ is a measure.
Ground-truth	By Relative Complement with Self is Empty Set, have $X^C(X) = \emptyset$.
BART	- Proof by induction: - By definition, the co-countable measure μ on X is: - By definition, μ is a probability measure iff: - Let $\mathcal{A}$ be the co-countable measure on X.
BART-FID	- By Relative Complement with Self is Empty Set: - The result follows from Relative Complement with Self is Empty Set: - The result follows from Relative Complement with Self is Empty Set and Relation Complement is Self-Empty Set. - qed
BART-FID + denoise	- From Relative Complement with Self is Empty Set: - The result follows from Relative Complement with Self is Empty Set: - The result follows from Relative Complement with Self is Empty Set: - qed

Table 4: *Next-step suggestion* example. The colors denote: Undefined term. Hallucinated reference. Reference not in entire ground-truth proof. Improper statement. Does not follow. Reference in ground-truth next step.

conditioning on knowledge is important, and the results show room for improvement in both the retrieval-augmented and document-grounded settings.

Case study – characterizing quality. In Tables 2 and 4 we analyze example generations for full proof and next-step generation, respectively, to (i) study the potential impact of conditioning on reference knowledge, and (ii) better understand errors to motivate research on improved generation methods and automatic evaluation of mathematical content.

We identify five error types: (1) *hallucinated references*, meaning the reference does not occur in NaturalProofs; (2) *non-ground-truth reference*, meaning the reference does not occur in the ground-truth proof; (3) *undefined terms*; (4) *improper or irrelevant statement*, meaning a statement that is mathematically invalid (e.g. $\frac{2}{3} \in \mathbb{Z}$) or irrelevant to the proof; and (5) statements that *do not follow* logically from the preceding statements. We also identify two positive properties: (A) using a term from a ground-truth reference, and (B) referring to a ground-truth reference. Properties (1), (2), (A), and (B) are currently feasible to automatically evaluate. An interesting research topic is automatically evaluating (3), (4), and (5), and testing whether these properties correlate with experts' proof quality.

Denoise	Knowledge	PPL	SBleu	F1
✓	GT-refs (8)	1.884	5.42	36.59
✓	–	1.940	3.66	8.98
–	GT-refs (8)	2.024	5.22	39.05
–	–	2.041	3.75	10.96

Table 5: Perplexity versus generation metrics for models with intermediate denoising training and without. Lower perplexity did not always imply better generation metrics.

5 Conclusion

We report initial results on two new knowledge-grounded generation tasks with educational implications: *next-step suggestion* and *full-proof generation*. The results indicate the importance of providing grounded knowledge to the model, and suggest future directions for improving performance in the document-grounded and retrieval-augmented settings, decoding algorithms that provide knowledge and leverage the proof structure, and automatically evaluating machine-generated content.

References

E. Dinan, S. Roller, K. Shuster, A. Fan, M. Auli, and J. Weston. Wizard of Wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

A. Fan, Y. Jernite, E. Perez, D. Grangier, J. Weston, and M. Auli. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL `https://aclanthology.org/P19-1346`.

J. M. Han, J. Rute, Y. Wu, E. W. Ayers, and S. Polu. Proof artifact co-training for theorem proving with language models, 2021.

G. Izacard and E. Grave. Leveraging passage retrieval with generative models for open domain question answering, 2020.

M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL `https://aclanthology.org/2020.acl-main.703`.

P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020b.

X. Lu, P. West, R. Zellers, R. Le Bras, C. Bhagavatula, and Y. Choi. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.339. URL `https://aclanthology.org/2021.naacl-main.339`.

F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL `https://aclanthology.org/D19-1250`.

F. Petroni, A. Piktus, A. Fan, P. Lewis, M. Yazdani, N. De Cao, J. Thorne, Y. Jernite, V. Karpukhin, J. Maillard, V. Plachouras, T. Rocktäschel, and S. Riedel. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL `https://aclanthology.org/2021.naacl-main.200`.

S. Prabhumoye, K. Hashimoto, Y. Zhou, A. W. Black, and R. Salakhutdinov. Focused attention improves document-grounded generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4274–4287, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.338. URL `https://aclanthology.org/2021.naacl-main.338`.

S. Welleck, J. Liu, R. L. Bras, H. Hajishirzi, Y. Choi, and K. Cho. Naturalproofs: Mathematical theorem proving in natural language. In *NeurIPS 2021 Track on Datasets and Benchmarks*, 2021.

S. Zhang, E. Dinan, J. Urbanek, A. Szlam, D. Kiela, and J. Weston. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1205. URL `https://aclanthology.org/P18-1205`.