NATURALPROOFS: Mathematical Theorem Proving in Natural Language

Sean Welleck [1 2] **Jiacheng Liu** [1] **Ronan Le Bras** [2] **Hannaneh Hajishirzi** [1 2] **Yejin Choi** [1 2] **Kyunghyun Cho** [3 4]

Abstract

Understanding and creating mathematics using natural mathematical language – the mixture of symbolic and natural language used by humans – is a challenging and important problem for driving progress in machine learning. As a step in this direction, we develop NATURALPROOFS, a large-scale dataset of mathematical statements and their proofs, written in natural mathematical language. Using NATURALPROOFS, we propose a mathematical reference retrieval task that tests a system's ability to determine the key results that appear in a proof. Large-scale sequence models excel at this task compared to classical information retrieval techniques, and benefit from language pretraining, yet their performance leaves substantial room for improvement. NATURALPROOFS opens many possibilities for future research on challenging mathematical tasks.[1]

Title	**Category of Monoids is Category**
Contents	Let Mon be the category of monoids. Then Mon is a metacategory.
Proof	Let us verify the axioms $(C1)$ up to $(C3)$ for a metacategory. We have Composite of Homomorphisms on Algebraic Structure is Homomorphism, verifying $(C1)$. We have monoid $(S, \circ)$. Now, $(C2)$ follows from Identity Mapping is Left Identity and Identity Mapping is Right Identity. Finally, $(C3)$ follows from Composition of Mappings is Associative. Hence Mon is a metacategory.

Table 1. A theorem and its proof from NATURALPROOFS. Given a theorem title and its contents, the mathematical retrieval task consists of retrieving the references (underlined) that occur in the proof. See Figure 6 and Figure 7 for details on the data format.

1. Introduction

Building artificial agents that are capable of mathematical reasoning is a stepping stone towards artificial intelligence. Constructing a mathematical proof involves symbolic manipulation, logical and analogical reasoning, as well as knowledge retrieval, while common sense and natural language abilities are needed to articulate the proof in a concise, comprehensible form. Mastering any of these skills represents a challenge for current machine learning methods, suggesting mathematics as a domain for driving progress in machine learning, while opening applications in education, software synthesis, and scientific discovery (Carter & Monks, 2013; Kang et al., 2020; Szegedy, 2020).

Recently, techniques developed for natural language processing have driven advances in *formalized mathematics* (e.g. Polu & Sutskever (2020); Rabe et al. (2021); Wu et al. (2021)), in which mathematics is written in a verifiable formal language that resembles source code, such as Mizar (Urban, 2006), Lean (de Moura et al., 2015), or Metamath (Megill & Wheeler, 2019). However, this setting does not directly address the *informal* aspect of human mathematics, which is conveyed with a mixture of symbolic and natural language (Gowers et al., 2008). We call it *natural mathematical language*. This aspect is crucial, since advancing *human understanding* is a goal of mathematics (Thurston, 1994), and a significant fraction of mathematical knowledge is contained in natural language text (Szegedy, 2020).

Developing machine learning systems that are capable of reasoning with natural mathematical language is a challenging research direction. It is not obvious how to define real mathematical tasks that are tractable, yet challenging, for current methods, and evaluating generated mathematical content is difficult. In this paper, we take a step towards this goal with NATURALPROOFS, a dataset of mathematical statements and their proofs, written in natural mathematical language. Using NATURALPROOFS, we propose a *mathe-*

[1]University of Washington [2]Allen Institute for Artificial Intelligence [3]New York University [4]CIFAR Fellow in Learning in Machines & Brains. Correspondence to: Sean Welleck <wellecks@uw.edu>.

[1]Dataset and code available at https://github.com/wellecks/naturalproofs.

matical reference retrieval task, which evaluates a system's ability to produce mathematical references that are used to prove a mathematical claim. This task represents a crucial aspect of theorem proving, in which a mathematician must determine the key results that appear in a proof. We expect that mastery of this task requires natural language ability, analogical and logical reasoning, knowledge retrieval, and mathematical domain knowledge. A machine learning system that performs well on this task could provide guidance to mathematicians or researchers while proving new theorems, aid in developing novel exercises, and offer hints to students that are stuck on how to proceed with a problem.

We demonstrate that large-scale neural sequence models developed for natural language processing excel at mathematical reference retrieval compared to classical information retrieval techniques. We find that language pretraining is an effective initialization for our mathematical task, and that the models leverage both the textual information in the title and the mathematical contents of each statement. However, in absolute terms their performance leaves substantial room for improvement. NATURALPROOFS opens many possibilities for developing and evaluating machine learning methods on challenging mathematical tasks.

2. Related Work

Machine learning for formalized mathematics. A large portion of work integrating machine learning with mathematical reasoning has focused on formalized mathematics. Early work by Urban (2006) used machine learning for selecting relevant premises in the Mizar mathematical library that are passed to an automated theorem prover, which was later explored with deep neural networks (Alemi et al., 2016). Bansal et al. (2019) developed the HOList benchmark based on the HOL Light theorem prover, while other benchmark tasks use the Coq (Huang et al., 2019; Yang & Deng, 2019), Metamath (Whalen, 2016; Wang & Deng, 2020; Polu & Sutskever, 2020), or Isabelle (Li et al., 2021) environments. These formalized settings differ from NATURALPROOFS, which uses mathematical language as humans write it. Szegedy (2020) argues for leveraging both informal and formal mathematics through autoformalization. Wang et al. (2020) explore translating between informal and formal mathematics, including via a dataset based on ProofWiki, on which NATURALPROOFS is also based, though their dataset is not made available.

Mathematics and language benchmarks. Several existing datasets evaluate a model's ability to solve multiple choice algebraic word problems (Roy & Roth, 2015; Ling et al., 2017; Amini et al., 2019) or arithmetic problems (Saxton et al., 2019) expressed with varying degrees of natural language. Lample & Charton (2020) evaluate neural sequence models on purely symbolic integration problems. Recently, Hendrycks et al. (2021) propose a benchmark based on mathematics competition problems. NATURALPROOFS focuses on theorem proving rather than calculation, which we hypothesize evaluates different capabilities, and may prove useful in bridging formal and informal settings.

Type	Property	Mean	25%	75%
Theorems	N	20,201	-	-
	Characters	196	68	270
	Lines	3.8	1.0	5.0
	References	2.9	0.0	4.0
Proofs	N	20,215	-	-
	Characters	953	350	1,179
	Lines	25.7	9.0	32.0
	References	6.9	2.0	9.0
Definitions	N	12,418	-	-
	Characters	252	97	331
	Lines	4.3	1.0	6.0
	References	3.4	1.0	5.0
Other	N	943	-	-
	Characters	1,602	888	1,890
	Lines	47.2	29.0	49.0
	References	9.26	4.0	11.0

Table 2. NATURALPROOFS Dataset statistics.

Large-scale neural language models. Large-scale unsupervised pretraining of language models has led to significant advances in many natural language processing domains (e.g. Devlin et al. (2019); Radford et al. (2019); Raffel et al. (2020); Brown et al. (2020)). Recent work suggests that these models store knowledge in their parameters (Petroni et al., 2020), are capable of reasoning in mathematical (Rabe et al., 2021; Wu et al., 2021) and language (Clark et al., 2020; Tafjord et al., 2020) domains, and are effective for information retrieval tasks (Nogueira & Cho, 2020; Nogueira dos Santos et al., 2020). These advances motivate our work, which explores mathematical reasoning in natural language with large-scale language models through a retrieval task.

3. The NATURALPROOFS Dataset

The NATURALPROOFS Dataset is a large-scale dataset for studying mathematical reasoning in natural language. NATURALPROOFS consists of roughly 20,000 theorem statements and proofs, 12,500 definitions, and 1,000 additional pages (e.g. axioms, corollaries) derived from ProofWiki, an online compendium of mathematical proofs written by a community of contributors.[2] Table 1 shows an example

[2] https://proofwiki.org/

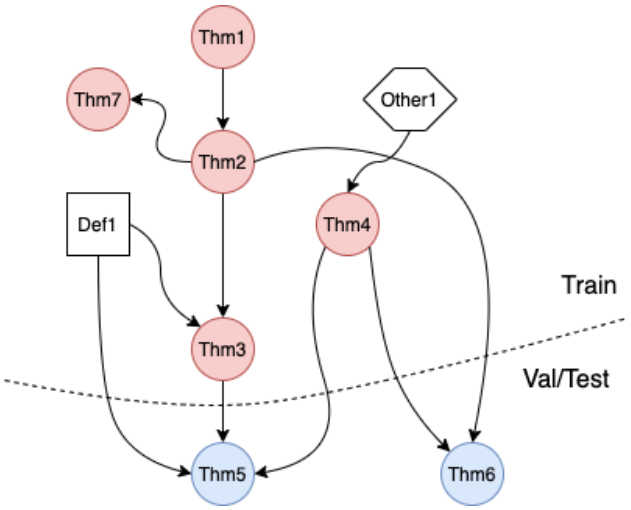

Figure 1. The reference graph. Nodes are *statements* (theorem / definition / others) and edges are *reference* links. An edge pointing from *statement* A to *theorem* B means that the proof for *theorem* B refers to *statement* A. Edges can start from any type of *statement* node, but they always end at a *theorem*. For the reference retrieval task, the dataset is split so that all theorems in the validation/test sets are *leaf* nodes in the reference graph. Here we show one possible split, where theorems in the validation/test sets are marked in blue, and those in the training set are marked in red.

theorem and its proof from NATURALPROOFS, and Table 2 shows dataset statistics.

Structure. NATURALPROOFS provides access to mathematical statements, proofs, and the references. A *statement* is either a theorem or definition. NATURALPROOFS provides the statement's title, contents, and references. The *contents* is a list of sequences, where each sequence contains one line of mixed text and LaTeX, with reference links displayed in their natural language form. A theorem is associated with one or more proofs when available. A *proof* contains a title, contents, and references in the same format as a statement. Finally, we collect *other* pages (e.g. axioms, corollaries). A *reference* is a theorem, definition, or other page that is linked to within the contents of a statement or proof. Figure 7 shows the data format for theorems, definitions, and proofs in NATURALPROOFS. All statements and the reference links connecting them form a *reference graph*, shown in Figure 1. The reference graph can contain cycles, e.g. `Pythagoras's Theorem` and `Sum of Squares of Sine and Cosine` refer to each other in their proofs.

Preprocessing. We filter and format the public ProofWiki XML dump.[3] We give a high-level overview of preprocessing here; for full details we release a Jupyter notebook. We filter pages according to manually designed rules (e.g. redirects, files, categories), and determine page type, contents, and references using each page's WikiMedia data structure. The theorem, definition, and proof contents are contained in a WikiMedia section that is determined for each page type according to a hand-defined rule. Since the roughly 1,000 other pages have varying page structures, we use their entire contents instead of a single section's contents. In addition to well-formed axiom and corollary statements, the other pages include misformatted theorem or definition statements that occur as references elsewhere in the corpus. We assign each reference (theorem, definition, or other page) a unique reference id, and assign each proof a unique proof id.

[3] https://proofwiki.org/xmldump/latest.xml. We use the November 12, 2020 version.

		Train	Valid	Test
Examples $(\mathbf{x}, \mathbf{y})$		11,399	1,099	1,099
Characters ($\|\mathbf{x}\|$)	*Mean*	219	224	222
	25%	110	113	114
	75%	280	288	287
References ($\|\mathbf{y}\|$)	*Mean*	8.2	7.7	7.7
	25%	3.0	3.0	3.0
	75%	10.0	10.0	10.0
References $\mathbf{r}$		28,473	30,671	30,671
Characters ($\|\mathbf{r}\|$)	*Mean*	273	269	269
	25%	94	96	96
	75%	316	314	314

Table 3. Retrieval dataset statistics. See Figure 4 for the distribution of references per example, and Figure 5 for the distribution of mentions per reference.

4. The NATURALPROOFS Retrieval Task

NATURALPROOFS opens many possible machine learning tasks that involve natural mathematical language. In this section, we propose a *mathematical reference retrieval* task: given a theorem $\mathbf{x}$, retrieve the set of references $\mathbf{y}$ that occur in the theorem's proof. An example is shown in Table 1, where the task is to retrieve the underlined references given the title and contents of the theorem `Category of Monoids is Category`. This task represents a crucial aspect of theorem proving, in which a mathematician must determine the key results that appear in a proof.

4.1. Retrieval Examples

Using NATURALPROOFS, we derive retrieval examples of the form $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} = (x_1, \ldots, x_T)$ is a *theorem statement*, and $\mathbf{y} = \{\mathbf{r}_1, \ldots, \mathbf{r}_M\}$, with $\mathbf{r}_m \in \mathcal{R}$, is the set of *references* that occur in a proof of statement $\mathbf{x}$. The set of all references, $\mathcal{R}$, consists of theorems, definitions, and other pages (see §3). We create a dataset containing 13,597 ex-

	recall@10	recall@100	avg-prec@100	full@100	full@1000
Random	0.05	0.27	0.01	0.00	0.00
Frequency	5.38	23.11	2.45	1.82	9.83
TF-IDF	10.32	23.20	6.05	8.92	16.92
BM25	10.54	23.93	5.95	9.74	16.47
LSTM	8.62	32.14	4.97	4.00	17.38
BERT No-Tune	1.45	5.81	0.86	1.91	6.55
BERT No-Pretrain	13.98	44.20	8.77	9.55	45.40
BERT	**20.27**	**59.44**	**14.01**	**27.39**	**70.52**

Table 4. Performance on the NATURALPROOFS retrieval task (test set). Recall is macro-averaged.

amples, with the corresponding reference set $\mathcal{R}$ containing 30,671 references. Figure 6 shows the data format for the retrieval example corresponding to the example in Table 1.

Table 3 contains statistics that summarize the sequence lengths and number of references for each data type. On average, each example contains a theorem that is roughly 220 characters long, paired with roughly 8 references. We show the distribution of references per validation example in Figure 4. Naturally, certain references are mentioned in statements and proofs more often than others; Figure 5 shows the number of mentions per reference, and Table 8 shows the most-mentioned references (e.g. `Set`, `Principle of Mathematical Induction`).

Training and evaluation splits. We design training and evaluation splits that reflect the real-world scenario of proving *newly seen* theorems at evaluation time. This requires careful attention, since each theorem and reference statement contains references to previous results; naively sampling evaluation examples would yield evaluation theorems that are referred to in the training set. However, newly seen theorems should not appear as previous results; that is, they are leaf nodes on the reference graph. Based on this observation, we form an evaluation set using a randomly sampled subset of reference graph leaf nodes, and use the remaining nodes as the training set (Figure 1). We use half of the evaluation set for validation and half for testing, resulting in 1,099 validation, 1,099 test, and 11,399 training examples. Since evaluation theorems are not referred to in training examples, the reference set for training is smaller than that for evaluation (Table 3).

5. Experiments

The aim of our experiments is to benchmark existing methods developed in natural language processing on the NATURALPROOFS mathematical reference retrieval task. To this end, we frame retrieval as binary classification (Nogueira & Cho, 2020) by training a model to predict whether a reference $\mathbf{r}$ occurs in the proof of a statement $\mathbf{x}$,

$$p_\theta(\mathbf{r} \in \text{proof}(\mathbf{x}) \mid \mathbf{x}) = \sigma\left(h_\theta\left(f_\theta(\mathbf{x}), g_\theta(\mathbf{r})\right)\right), \quad (1)$$

where $f_\theta(\mathbf{x}) \in \mathbb{R}^d$, $g_\theta(\mathbf{r}) \in \mathbb{R}^d$, and $h_\theta(f_\theta(\mathbf{x}), g_\theta(\mathbf{r})) \in \mathbb{R}$. By varying the parameterization of f_θ and g_θ, which encode the statement and reference, respectively, we can evaluate the performance of commonly used natural language processing methods on the task of reference retrieval.

Our main model parameterizes f_θ and g_θ with separate instances of BERT (Devlin et al., 2019), a large-scale transformer encoder pretrained on natural language text. We also include variants without pretraining, without training on NATURALPROOFS, and a model that uses an LSTM instead of the BERT transformer.

To train a model on NATURALPROOFS we use binary cross entropy loss with positive and negative references. That is, given an example $(\mathbf{x}, \mathbf{y})$, we sample a set of negative references $\mathbf{y}^-$ from $\mathcal{R} \backslash \mathbf{y}$ and use the loss,

$$\mathcal{L} = -\sum_{\mathbf{r} \in \mathbf{y}} \log p_\theta(\mathbf{r}) - \sum_{\mathbf{r}_- \in \mathbf{y}_-} \log\left(1 - p_\theta(\mathbf{r}^-)\right), \quad (2)$$

where p_θ is given by Equation 1.

We form a dataset of $(\mathbf{x}, \mathbf{r}, y \in \{0, 1\})$ tuples containing a statement, a reference, and an indicator of whether the reference is positive or negative. To do so, for each example $(\mathbf{x}, \mathbf{y} = \{\mathbf{r}_1, \ldots, \mathbf{r}_M\})$, we uniformly sample n negative references, $\mathbf{y}^- = \{\mathbf{r}_1^-, \ldots, \mathbf{r}_n^-\}$, from $\mathcal{R} \backslash \mathbf{y}$. We report results with $n = 200$. We leave more sophisticated negative sampling strategies as future work.

5.1. Results

The main results for mathematical reference retrieval are shown in Table 4. The BERT model outperforms the neural LSTM baseline, classical information retrieval baselines, and naive strategies in terms of recall, average precision, and the ability to fully recover all true references within the top-k results. The BERT model ranks roughly one out of

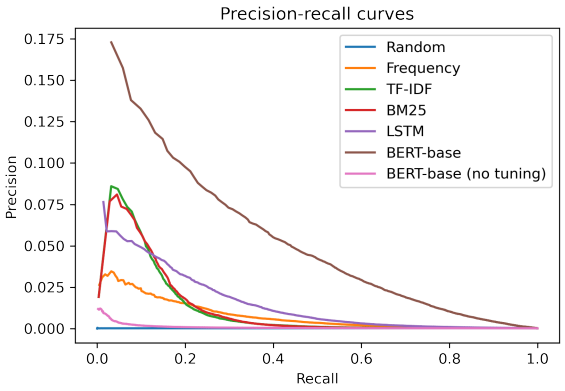

Figure 2. Precision-recall curves on NATURALPROOFS retrieval.

Model	Title	Content	recall@100	full@100
TF-IDF	✗	✓	17.05	6.46
	✓	✗	**28.80**	**12.28**
	✓	✓	21.88	8.28
BERT	✗	✓	54.63	20.93
	✓	✗	54.28	24.84
	✓	✓	**58.20**	**25.66**

Table 5. Excluding (✗) the title or content of theorems and references (validation set). The best metric for each model class is shown in bold.

every five true references within its top 10 results (recall@10 20.27), and more than half within the top 100 (recall@100 59.44). For one out of four theorems, the model's top 100 references contain *all* of the references that occur in the theorem's proof (full@100 27.39). Finally, the BERT model gives the most favorable trade-off between precision and recall (Figure 2).

The results demonstrate that large-scale pretrained neural language models are effective for mathematical reference retrieval compared to previous alternatives. However, despite their strong relative performance, their absolute performance leaves a large amount of room for improvement. For instance, although each theorem contains on average *eight* true references, the BERT model fails to retrieve all of them in the top *hundred* roughly 75% of the time. To better understand the model's strengths, weaknesses, and the factors that contribute to its performance, we perform several analyses beginning with a qualitative evaluation.

Qualitative evaluation. Table 6 shows model predictions for a representative theorem, `Category of Monoids is Category`. The model retrieves four out of seven true references within its top 100 results, including one definition and three theorems. Its top 10 results are comprised of definitions that are related to category theory, which is the subject of the theorem. This illustrates the model's ability to retrieve *relevant* references, while highlighting its inability to always perform the fine-grained distinction between a relevant reference and one that occurs in the ground-truth proof(s). Arguably, such a system is still useful for providing hints to a user, so long as the user is confident that all of the true references are in a reasonably small set of results relative to the full set of around 30,000 references. Thus the model's failure to retrieve all of the necessary theorems in the top 100 – e.g. `Identity Mapping is Automorphism` is ranked 168 – points to an important performance gap that can be narrowed by future models. In the Appendix, we provide additional predictions on an easy example (Table 10), a difficult example (Table 11), and an example with a large set of true references (Table 12).

Language pretraining and NATURALPROOFS training. The BERT model has two learning phases: pretraining on language data, and fine-tuning on NATURALPROOFS. We investigate how each phase contributes to BERT's performance with two baselines. The **No-Tune** baseline uses a pretrained BERT model without fine-tuning on NATURALPROOFS. As seen in Table 4, this baseline's performance is low, meaning that fine-tuning on NATURALPROOFS was necessary. The **No-Pretrain** baseline trains a randomly initialized BERT model from scratch on NATURALPROOFS. This baseline outperforms the classical retrieval methods, yet it substantially underperforms the BERT model that included both pretraining and finetuning, meaning that language pretraining served as an effective initialization for the mathematical retrieval task.

Learned representations. To further investigate the effect of NATURALPROOFS fine-tuning, we compare the vector representations of references from the finetuned BERT model with the non-finetuned BERT model in Figure 3. We run T-SNE on the non-finetuned model's representation, and use the resulting embeddings to initialize T-SNE on the finetuned model's representations. We use the tags provided by NATURALPROOFS to illustrate how the reference representations moved, by showing a line when the representation with the particular tag has moved significantly. The learned model's representation groups similar mathematical concepts together that are initially in separate regions of the representation space. For instance, the learned model groups Tutte's Wheel Theorem, Kruskal's Algorithm, and the Handshake Lemma in the Graph Theory cluster, while the non-finetuned model places these in separate regions.

Title and content ablation. Each theorem statement and reference consists of a natural language title, as well as *contents* that is a mixture of symbolic mathematics and

Title	**Category of Monoids is Category**	
Contents	Let Mon be the category of monoids. Then Mon is a metacategory.	
In Top-100	**Rank**	**Reference**	**Type**
✓	7	Metacategory	Definition
✓	40	Identity Mapping is Left Identity	Theorem
✓	61	Identity Mapping is Right Identity	Theorem
✓	89	Composition of Mappings is Associative	Theorem
	125	Monoid	Definition
	136	Composite of Homomorphisms is Homomorphism/Algebraic Structure	Theorem
	168	Identity Mapping is Automorphism	Theorem
In True	**Rank**	**Reference**	**Type**
	1	Identity Morphism	Definition
	2	Monomorphism (Category Theory)	Definition
	3	Epimorphism (Category Theory)	Definition
	4	Morphism	Definition
	5	Commutative Diagram	Definition
	6	Composition of Morphisms	Definition
✓	7	Metacategory	Definition
	8	Functor	Definition
	9	Morphism Property	Definition
	10	Closure (Abstract Algebra)/Algebraic Structure	Definition

Table 6. Retrieval for a representative theorem, meaning the theorem has an average number of true references and the model predicts roughly half within the top-100.

	All	**Thms**	**Defs**	**Others**
Frequency	23.69	9.40	31.92	13.39
TF-IDF	21.88	35.16	14.99	19.67
LSTM	31.86	16.31	40.64	23.43
BERT	58.20	48.22	64.34	44.77

Table 7. Retrieval performance (recall@100) by reference type on the validation set.

natural language. We investigate whether the learned model relies on the titles, contents, or both for performing the mathematical reference retrieval task. Table 5 shows results for models trained and evaluated with titles alone, contents alone, or both titles and contents.

The classical tf-idf strategy, which operates using token frequencies, performs best when only the titles are provided. The BERT model, in contrast, leverages the contents of the theorem and references, in addition to the language present in their titles. For instance, Table 5 shows that using titles alone gives lower recall but higher full recovery than using contents alone, but when the model has access to both sources of information, the model achieves highest performance according to both metrics.

Performance by reference type. In Table 7 we break down the retrieval performance by reference type. The BERT model's performance improvement over TF-IDF is much larger for definitions than it is for theorems. The LSTM model shows an analogous, but more extreme, relative performance: it is much better than TF-IDF on definitions, but *worse* than TF-IDF on theorems. We speculate that it is easier for the neural model to semantically group definitions, compared to theorems.

5.2. Future Directions

We demonstrated how NATURALPROOFS is used for a mathematical reference retrieval task that reflects a key step in real-world theorem proving, and allows for automatic evaluation with standard information retrieval metrics. In contrast, evaluating generative tasks is challenging as it involves evaluating a proof's correctness, so we do not consider these tasks here. Using NATURALPROOFS to develop methods for evaluating generated mathematical content is an interesting direction for future work.

The performance improvements brought by large-scale sequence models and language pretraining suggest that further exploring NLP techniques for mathematical tasks is another fruitful direction. The relatively small gap between the BERT model and its title-only variant leaves room for im-

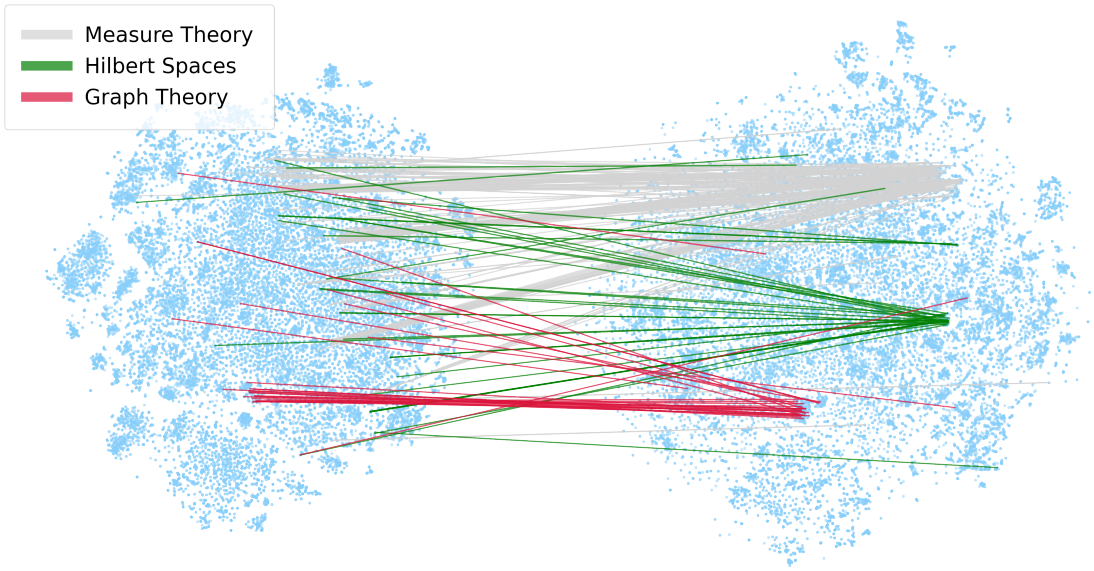

Figure 3. TSNE embeddings for the reference representations before finetuning (left) and after finetuning on NATURALPROOFS (right).

provement with methods that better take advantage of the mathematical contents. Recovering the number of occurrences of each reference – e.g. framing retrieval as multiset prediction (Welleck et al., 2018) – or recovering the order of references are also interesting to investigate.

Finally, NATURALPROOFS provides a data schema for mathematical theorems, proofs, and references that is independent of the ProofWiki data used to the populate the dataset. With this schema in place, an exciting direction is leveraging other mathematical data sources for expanding the training set or deriving challenging evaluation sets.

6. Conclusion

Building agents that understand and create mathematics using *natural mathematical language* is a challenging research direction, providing a means for evaluating and developing machine learning methods capable of symbolic and natural language understanding. As a step in this direction, we develop NATURALPROOFS, a large-scale dataset for studying mathematical reasoning in natural language. We propose a retrieval task that represents a key step in real-world theorem proving: choosing the existing results that occur in the proof of a novel mathematical claim. Our experiments suggest that the task is tractable, yet challenging, for current large-scale neural sequence models. NATURALPROOFS opens many promising avenues for future research.

References

Alemi, A. A., Chollet, F., Een, N., Irving, G., Szegedy, C., and Urban, J. DeepMath - Deep sequence models for premise selection. In Advances in Neural Information Processing Systems, pp. 2243–2251, 2016.

Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, volume 1, 2019.

Bansal, K., Loos, S., Rabe, M., Szegedy, C., and Wilcox, S. Holist: An environment for machine learning of higher-order theorem proving. In 36th International Conference on Machine Learning, ICML 2019, volume 2019-June, 2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing

Systems, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Carter, N. C. and Monks, K. G. Lurch: a word processor that can grade students' proofs. In Lange, C., Aspinall, D., Carette, J., Davenport, J. H., Kohlhase, A., Kohlhase, M., Libbrecht, P., Quaresma, P., Rabe, F., Sojka, P., Whiteside, I., and Windsteiger, W. (eds.), Joint Proceedings of the MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at CICM, Bath, UK, volume 1010 of CEUR Workshop Proceedings. CEUR-WS.org, 2013. URL http://ceur-ws.org/Vol-1010/paper-04.pdf.

Clark, P., Tafjord, O., and Richardson, K. Transformers as Soft Reasoners over Language. In Bessiere, C. (ed.), Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 3882–3890. International Joint Conferences on Artificial Intelligence Organization, 2020.

de Moura, L. M., Kong, S., Avigad, J., van Doorn, F., and von Raumer, J. The lean theorem prover (system description). In Felty, A. P. and Middeldorp, A. (eds.), CADE, volume 9195 of Lecture Notes in Computer Science, pp. 378–388. Springer, 2015. ISBN 978-3-319-21400-9. URL http://dblp.uni-trier.de/db/conf/cade/cade2015.html#MouraKADR15.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Gowers, T., Barrow-Green, J., and Leader, I. The Princeton Companion to Mathematics. Princeton University Press, USA, illustrated edition edition, 2008. ISBN 0691118809.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset, 2021.

Huang, D., Dhariwal, P., Song, D., and Sutskever, I. Gamepad: A learning environment for theorem proving. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=r1xwKoR9Y7.

Kang, D., Head, A., Sidhu, R., Lo, K., Weld, D., and Hearst, M. A. Document-Level Definition Detection in Scholarly Documents: Existing Models, Error Analyses, and Future Directions. In Proceedings of the First Workshop on Scholarly Document Processing, pp. 196–206, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.sdp-1.22. URL https://www.aclweb.org/anthology/2020.sdp-1.22.

Lample, G. and Charton, F. Deep learning for symbolic mathematics. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=S1eZYeHFDS.

Li, W., Yu, L., Wu, Y., and Paulson, L. C. Isarstep: a benchmark for high-level mathematical reasoning. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=Pzj6fzU6wkj.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), volume 1, 2017. doi: 10.18653/v1/P17-1015.

Megill, N. D. and Wheeler, D. A. Metamath: A Computer Language for Mathematical Proofs. Lulu Press, Morrisville, North Carolina, 2019. http://us.metamath.org/downloads/metamath.pdf.

Nogueira, R. and Cho, K. Passage re-ranking with bert, 2020.

Nogueira dos Santos, C., Ma, X., Nallapati, R., Huang, Z., and Xiang, B. Beyond [CLS] through ranking by generation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1722–1727, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.134. URL https://www.aclweb.org/anthology/2020.emnlp-main.134.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., and Riedel, S. Language models as knowledge bases? In EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference, 2020. doi: 10.18653/v1/d19-1250.

Polu, S. and Sutskever, I. Generative language modeling for automated theorem proving, 2020.

Rabe, M. N., Lee, D., Bansal, K., and Szegedy, C. Mathematical reasoning via self-supervised skip-tree training. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=YmqAnY0CMEy.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Roy, S. and Roth, D. Solving general arithmetic word problems. In Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, 2015. doi: 10.18653/v1/d15-1202.

Saxton, D., Grefenstette, E., Hill, F., and Kohli, P. Analysing mathematical reasoning abilities of neural models. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=H1gR5iR5FX.

Szegedy, C. (ed.). A Promising Path Towards Autoformalization and General Artificial Intelligence, 2020.

Tafjord, O., Mishra, B. D., and Clark, P. Proofwriter: Generating implications, proofs, and abductive statements over natural language. ArXiv, abs/2012.13048, 2020.

Thurston, W. P. On proof and progress in mathematics. arXiv:math/9404236, March 1994. URL http://arxiv.org/abs/math/9404236. arXiv: math/9404236.

Urban, J. Mptp 0.2: Design, implementation, and initial experiments. J. Autom. Reason., 37(1–2):21–43, August 2006. ISSN 0168-7433. doi: 10.1007/s10817-006-9032-3. URL https://doi.org/10.1007/s10817-006-9032-3.

Wang, M. and Deng, J. Learning to Prove Theorems by Learning to Generate Theorems. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 18146–18157. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d2a27e83d429f0dcae6b937cf440aeb1-Paper.pdf.

Wang, Q., Brown, C., Kaliszyk, C., and Urban, J. Exploration of neural machine translation in autoformalization of mathematics in Mizar. In CPP 2020 - Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs, co-located with POPL 2020, 2020. doi: 10.1145/3372885.3373827.

Welleck, S., Yao, Z., Gai, Y., Mao, J., Zhang, Z., and Cho, K. Loss functions for multiset prediction. In NeurIPS, 2018.

Whalen, D. Holophrasm: a neural automated theorem prover for higher-order logic, 2016.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

Wu, Y., Rabe, M., Li, W., Ba, J., Grosse, R., and Szegedy, C. Lime: Learning inductive bias for primitives of mathematical reasoning, 2021.

Yang, K. and Deng, J. Learning to prove theorems via interacting with proof assistants. In 36th International Conference on Machine Learning, ICML 2019, volume 2019-June, 2019.

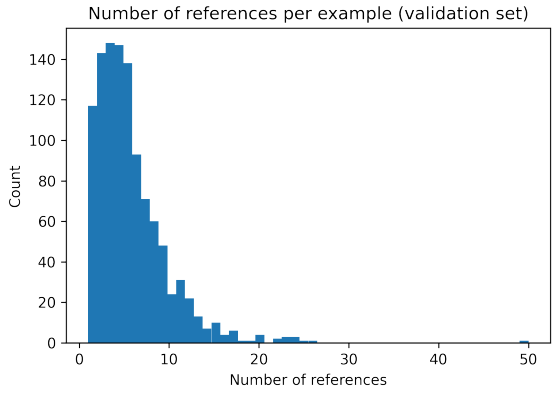

Figure 4. True references per example (validation set).

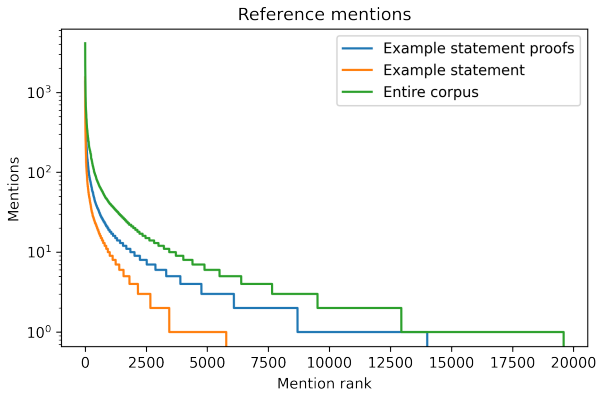

Figure 5. Reference mentions. See Table 8 for the titles of the most mentioned references.

Definitions	Set
	Subset
	Element
	Prime Number
	Mapping
	Integer
	Divisor (Algebra)/Integer
	Real Number
	Open Set/Topology
	Topological Space
Non-definitions	Principle of Mathematical Induction
	Proof by Contradiction
	Integration by Substitution
	Power Rule for Derivatives
	Linear Combination of Integrals
	Primitive of Power
	Sigma Function of Integer
	Integration by Parts
	Derivative of Composite Function
	Primitive of Constant Multiple of Function

Table 8. Top-10 most mentioned references (entire corpus).

# True refs	N	full@100	recall@100
$[1, 5)$	555	40.00	61.88
$[5, 10)$	410	13.66	61.27
$[10, 20)$	119	3.36	53.65
$[20, 51)$	15	0.00	39.62

Table 9. BERT-Base performance binned by number of true references (see Figure 4).

A. Dataset Details

Data format. We format each sequence ($\mathbf{x}$ or $\mathbf{r}$) as, `[CLS]` title `[SEP]` statement `[SEP]`, and we truncate the statement when the sequence exceeds the model's maximum length.

Most mentioned references. Table 8 shows the most mentioned references in the NATURALPROOFS corpus.

B. Experimental Setup

Training. Models are implemented with `transformers` (Wolf et al., 2020) and `pytorch-lightning`[4] and trained for 500,000 steps on four Quadro RTX 8000 GPUs. Each batch contains a maximum of 16,384 (2^{14}) tokens. Validation is done every 5,000 steps, with metrics derived from the $(\mathbf{x}, \mathbf{r}, \{0, 1\})$ tuples in the validation set (versus producing a full ranked list as in the Evaluation paragraph below). The model with the highest F1 computed on these validation tuples is selected for final evaluation.

Evaluation. The full set of inputs $\mathbf{x}$ and the full set of references $\mathcal{R}$ are pre-encoded using their respective trained models (i.e. two instances of BERT). Then the encodings for each possible $\mathbf{x}, \mathbf{r}$ pair are passed through the model's trained MLP to obtain a scalar score, which induces a ranked list of references for each input.

C. Additional Results

Table 9 shows retrieval performance binned by the number of true references (validation set).

[4] https://github.com/PyTorchLightning/pytorch-lightning

```
{
  'type': 'theorem',
  'has_proof': True,
  'title': 'Category of Monoids is Category',
  'proof_titles': ['Category of Monoids is Category'],
  'categories': ['Category of Monoids'],
  'statement': {
    'contents': [
        'Let $\\mathbf{Mon}$ be the [[Definition:Category of Monoids|category of monoids]].',
        'Then $\\mathbf{Mon}$ is a [[Definition:Metacategory|metacategory]].'
    ],
    'refs': ['Definition:Category of Monoids', 'Definition:Metacategory'],
    'read_contents': [
        'Let $\\mathbf{Mon}$ be the category of monoids.',
        'Then $\\mathbf{Mon}$ is a metacategory.'
    ],
    'ref_ids': [22919, 21454]
  },
  'proofs': [{
    'title': 'Category of Monoids is Category',
    'refs': [
        'Definition:Metacategory',
        'Composite of Homomorphisms is Homomorphism/Algebraic Structure',
        'Identity Mapping is Automorphism',
        'Definition:Monoid',
        'Identity Mapping is Left Identity',
        'Identity Mapping is Right Identity',
        'Composition of Mappings is Associative',
        'Definition:Metacategory'
    ],
    'ref_ids': [21454, 3852, 418, 19948, 217, 4387, 1494, 21454],
    'proof_id': 4682}
  ],
  'example_id': 4359,
  'theorem_id': 5480
}
```


Figure 6. NATURALPROOFS JSON for the retrieval example corresponding to the theorem and proof shown in Table 1. Using the notation of subsection 4.1, an $(\mathbf{x}, \mathbf{y})$ example is formed where $\mathbf{x}$ is the concatenation of `'title'` and `'read_contents'`, and $\mathbf{y}$ is a set formed with the proof's `'ref_ids'`. NATURALPROOFS provides additional JSON objects for each proof and reference; see Figure 7.


```
{
    'type': 'theorem',
    'has_proof': boolean,
    'has_contents': boolean,
    'title': string,
    'proof_titles': [string],
    'contents': [string],
    'read_contents': [string],
    'refs': [string],
    'categories': [string],
    'id': int
}
```

```
{
    'type': 'definition',
    'title': string,
    'has_contents': boolean,
    'contents': [string],
    'refs': [string],
    'categories': [string],
    'read_contents': [string],
    'id': int
}
```

```
{
    'type': 'proof',
    'title': string,
    'contents': [string],
    'refs': [string],
    'categories': [string],
    'read_contents': [string],
    'proof_id': int
}
```


Figure 7. NATURALPROOFS schema for theorems (left), definitions (middle), and proofs (right).

Sum of Strictly Positive Real Numbers is Strictly Positive

$x, y \in \mathbb{R}_{>0} \implies x + y \in \mathbb{R}_{>0}$

In Top-100	Rank	Reference	Type
✓	3	Real Number/Axioms	Definition
✓	5	Real Number Ordering is Compatible with Addition	Theorem
✓	13	Real Number Inequalities can be Added	Theorem
In True	**Rank**	**Reference**	**Type**
	1	Extended Real Addition	Definition
	2	Strictly Monotone/Mapping	Definition
✓	3	Real Number/Axioms	Definition
	4	Strictly Monotone Mapping with Totally Ordered Domain is Injective	Theorem
✓	5	Real Number Ordering is Compatible with Addition	Theorem
	6	Ordering on Extended Real Numbers	Definition
	7	Strictly Monotone/Real Function	Definition
	8	Real Number Ordering is Transitive	Theorem
	9	Extended Real Subtraction	Definition
	10	Monotone (Order Theory)/Mapping	Definition
	11	Strictly Decreasing Mapping is Decreasing	Theorem
	12	Strictly Monotone Mapping is Monotone	Theorem
✓	13	Real Number Inequalities can be Added	Theorem
	14	Upper Bound of Mapping	Definition
	15	Real Number Ordering is Compatible with Multiplication	Theorem

Table 10. Retrieval for a theorem that was easy for the model, meaning all of the true references were in the top-15 predictions.

Characteristic Subgroup is Transitive

Let G be a group.
Let H be a characteristic subgroup of G.
Let K be a characteristic subgroup of H.
Then K is a characteristic subgroup of G.

In Top-100	Rank	Reference	Type
	174	Group Automorphism	Definition
	308	Characteristic Subgroup	Definition
	429	Group Homomorphism Preserves Subgroups	Theorem
	616	Restriction/Mapping	Definition
In True	**Rank**	**Reference**	**Type**
	1	Subgroup	Definition
	2	Normal Subgroup	Definition
	3	Group	Definition
	4	Abelian Group	Definition
	5	Quotient Group	Definition
	6	Identity (Abstract Algebra)/Two-Sided Identity	Definition
	7	Subgroup of Abelian Group is Normal	Theorem
	8	Generator of Group	Definition
	9	Subgroup of Abelian Group is Abelian	Theorem
	10	Quotient Group is Group	Theorem
	11	Trivial Subgroup	Definition
	12	Intersection of Subgroups is Subgroup	Theorem
	13	Order of Structure	Definition
	14	Coset/Left Coset	Definition
	15	Normalizer	Definition

Table 11. Retrieval for a theorem that was difficult for the model, meaning none of the true references were in the top-100 predictions.

Schur-Zassenhaus Theorem

The square root of any prime number is irrational.

In Top-100	Rank	Reference	Type
✓	3	Normal Subgroup	Definition
✓	7	Center (Abstract Algebra)/Group	Definition
✓	8	Normalizer	Definition
✓	15	Group	Definition
✓	21	Identity (Abstract Algebra)/Two-Sided Identity	Definition
✓	25	Order of Structure	Definition
✓	37	Lagrange's Theorem (Group Theory)	Theorem
✓	50	Sylow p-Subgroup	Definition
✓	77	Set	Definition
✓	88	Index of Subgroup	Definition
	104	First Sylow Theorem	Theorem
	113	Second Isomorphism Theorem/Groups	Theorem
	156	Frattini's Argument	Theorem
	164	Principle of Mathematical Induction	Theorem
	225	Prime Number	Definition
	383	Equivalence Relation	Definition
	388	Complement of Subgroup	Definition
	678	Coprime/Integers	Definition
	1184	Characteristic Subgroup of Normal Subgroup is Normal	Theorem
	1311	Center is Characteristic Subgroup	Theorem
	2129	Hall Subgroup	Definition
	5733	Preimage/Relation/Subset	Definition
	13888	Cohomology Groups	Definition
In True	**Rank**	**Reference**	**Type**
	1	Subgroup	Definition
	2	Subgroup of Abelian Group is Normal	Theorem
✓	3	Normal Subgroup	Definition
	4	Correspondence Theorem (Group Theory)	Theorem
	5	Intersection of Subgroups is Subgroup	Theorem
	6	Finite Group	Definition
✓	7	Center (Abstract Algebra)/Group	Definition
✓	8	Normalizer	Definition
	9	Trivial Group	Definition
	10	Simple Group	Definition
	11	Trivial Subgroup	Definition
	12	Subgroup of Abelian Group is Abelian	Theorem
	13	Abelian Group	Definition
	14	Proper Subgroup	Definition
✓	15	Group	Definition

Table 12. Retrieval for a theorem with many true references.