NaturalProofs: Mathematics meets Natural Language

Sean Welleck[12], Jiacheng Liu[1], Ronan Le Bras[2],
Hannaneh Hajishirzi[12], Yejin Choi[12], and Kyunghyun Cho[3]

[1] University of Washington
[2] Allen Institute for Artificial Intelligence
[3] New York University

1 Introduction.

Solving the problem of understanding and creating mathematics using natural mathematical language – the mixture of symbolic and natural language used by humans – is a path towards developing agents capable of reasoning. The mixture of symbolic and natural text in informal mathematics, along with the existence of a formal counterpart, offers a unique setting for studying reasoning that complements research involving natural language alone or purely within a formal system. Moreover, systems that operate on informal mathematical text have applications in education and scientific discovery [2, 5, 9], while bridging informal and formal mathematics can be a key driver of progress in automated reasoning [10].

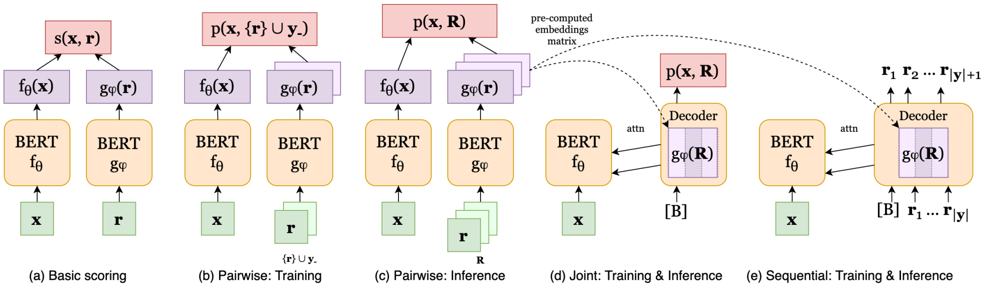
This talk will discuss NaturalProofs, a multi-domain corpus of mathematical statements and their proofs, written in natural mathematical language. NaturalProofs consists of 32k theorem statements and proofs, 14k definitions, and 2k other types of pages (e.g. axioms, corollaries) derived from three domains: *broad-coverage* data from ProofWiki,[1] an online compendium of mathematical proofs written by a community of contributors; *deep-coverage* data from the Stacks project,[2] a collaborative web-based textbook of algebraic geometry; and *low-resource, real-world* data from mathematics textbooks.[3] NaturalProofs unifies these sources in a common schema and is made publicly available as a resource to drive progress on tasks involving informal mathematics, complementing existing work in this direction (e.g. [4, 11, 7]).

We use NaturalProofs for *mathematical reference retrieval*, an analogue of premise selection [1, 4]: given a theorem $\mathbf{x}$, retrieve the set of references $\mathbf{y} = \{\mathbf{r}_1, \dots, \mathbf{r}_{|\mathbf{y}|}\}$ (theorems, lemmas, definitions) that occur in its proof. As a bridge towards generative tasks, we consider *mathematical reference generation*: given a theorem $\mathbf{x}$ generate the *sequence* of references in its proof, $\mathbf{y} = (\mathbf{r}_1, \dots, \mathbf{r}_{|\mathbf{y}|})$, which requires determining the order and number of references.

In addition to standard *in-distribution* evaluation, we evaluate *out-of-distribution*, zero-shot generalization to textbooks. We design an evaluation protocol that tests a system's ability to retrieve references for novel theorems in each setting, and benchmark methods based on large-scale neural sequence models [3, 6], including a strong *joint retrieval* method and a *sequential* variant for reference generation. The multiple informal domains, evaluation protocol, joint retrieval model, and reference generation task differ from previous work on ProofWiki [4] and formal [1, 8] premise selection.

We find that the neural methods are effective for in-domain retrieval compared to classical techniques, yet out-of-distribution generalization, leveraging symbolic mathematical content, and fully recovering a proof's references remain as fundamental challenges. The NaturalProofs data, code, and pretrained models are made publicly available.[4]

[1] https://proofwiki.org/
[2] https://stacks.math.columbia.edu/
[3] *Introduction to Real Analysis* by William F. Trench and *Elementary Number Theory* by William Stein.
[4] https://github.com/wellecks/naturalproofs.



(a) Basic scoring (b) Pairwise: Training (c) Pairwise: Inference (d) Joint: Training & Inference (e) Sequential: Training & Inference

Figure 1: The pairwise, joint, and sequential methods.

1.1 Retrieval and generation methods.

As benchmark methods for our tasks, we introduce *pairwise* and *joint* retrieval methods, and a *sequential* method trained for reference generation. Figure 1 illustrates the methods.

Pairwise model. The pairwise model scores a reference $\mathbf{r}$ against a theorem $\mathbf{x}$ using two instances of BERT [3], $s_\theta(\mathbf{x}, \mathbf{r}) = f_{\theta_1}^{\text{thm}}(\mathbf{x})^\top g_{\theta_2}^{\text{ref}}(\mathbf{r})$, as illustrated in Figure 1 (a). The pairwise model is trained to contrast each positive reference with a set of negative references,

$$\mathcal{L}(\mathbf{x}, \mathbf{r}, \mathbf{y}_-) = -\log \frac{\exp(s_\theta(\mathbf{x}, \mathbf{r}))}{\exp(s_\theta(\mathbf{x}, \mathbf{r})) + \sum_{\mathbf{r}_- \in \mathbf{y}_-} \exp(s_\theta(\mathbf{x}, \mathbf{r}_-))}, \tag{1}$$

where $\mathbf{r}$ is a reference that occurs in the proof of $\mathbf{x}$, and $\mathbf{y}_-$ is a set of in-batch negatives [6] (Figure 1 (b)). This benchmark represents methods such as the dense passage retriever [6].

Joint model. The joint model scores all references in a single pass, $p_\theta(\cdot|\mathbf{x}) = \text{softmax}\left(\mathbf{R} f_\theta(\mathbf{x})\right)$, where $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}| \times d}$ is a reference embedding matrix and $f_\theta(\mathbf{x}) \in \mathbb{R}^d$ is a neural theorem encoder (Figure 1 (d)). This model has the advantage of computing the loss denominator in Equation 1 over *all* references rather than a subset of negatives. However, it must learn implicit representations of each reference without observing reference contents, thus we populate its embedding matrix using the pairwise reference encoder's embeddings, $\mathbf{R} = \left[g^{\text{ref}}(\mathbf{r}_1); \ldots; g^{\text{ref}}(\mathbf{r}_{|\mathcal{R}|})\right]$.

Retrieval evaluation. Given an input theorem $\mathbf{x}$, every reference is scored to induce a ranking $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(|R|)}$ (Figure 1 (c,d)). The ranked references are compared against the ground-truth references from the proof of $\mathbf{x}$ using standard retrieval metrics, as well as whether *all* ground-truth references were ranked in the top-k.

Sequential model. We use an autoregressive model, $p_\theta(\mathbf{r}_1, \ldots, \mathbf{r}_{|\mathbf{y}|}|\mathbf{x}) = \prod_{t=1}^{|\mathbf{y}|+1} p_\theta(\mathbf{r}_t|\mathbf{r}_{<t}, \mathbf{x})$, where $\mathbf{r}_{|\mathbf{y}|+1}$ is a special $\langle\text{eos}\rangle$ token denoting the end of the reference sequence (Figure 1 (e)). The autoregressive model is trained to maximize the log-likelihood of ground-truth reference sequences. Unlike the preceding retrieval models, this model predicts the order and total number of references and can predict multiple occurrences of each reference.

For reference generation, beam search is used to generate a reference sequence $\hat{\mathbf{y}} = (\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{|\hat{\mathbf{y}}|}\, \langle\text{eos}\rangle)$. For retrieval, we populate a ranked list using generations $\{\hat{\mathbf{r}}_1, \ldots, \hat{\mathbf{r}}_{|\hat{\mathbf{y}}|}\}$ followed by references ordered according to the first step's probabilities, $p_\theta(\mathbf{r}_1|\mathbf{x})$.

		mAP	R@10	Full@10
PWiki	**TF-IDF**	6.19	10.27	4.14
	BERT pair	16.82	23.73	7.31
	BERT joint	**36.75**	**42.45**	**20.35**
Stacks	**TF-IDF**	13.64	25.46	18.94
	BERT pair	20.93	37.43	30.03
	BERT joint	**28.32**	**39.10**	**31.96**

Table 1: _In-domain_ test performance on mathematical reference retrieval, measured with mean average precision (mAP), recall (R@10) and full retrieval (Full@10) at 10.

Source	**ProofWiki**
Theorem	**Category of Monoids is Category** Let Mon be the category of monoids. Then Mon is a metacategory.
Rank	**Retrieved Reference (Joint Model)**
1	_Metacategory_
2	_Monoid_
3	Identity Morphism
4	_Identity Mapping is Right Identity_
5	_Identity Mapping is Left Identity_
6	Associative
7	Identity (Abstract Algebra)/Two-Sided Identity
8	_Composition of Mappings is Associative_
9	Composition of Morphisms
10	Semigroup

Table 3: Example top-10 retrievals. Italicized references are in the ground-truth proof.

		mAP	R@10	Full@10
RA	**TF-IDF**	**15.79**	**34.65**	**27.54**
	BERT pair	13.24	24.01	19.16
	BERT joint	11.24	20.97	16.77
NT	**TF-IDF**	**16.42**	39.62	30.00
	BERT pair	15.12	**41.51**	**35.00**
	BERT joint	15.85	41.51	35.00

Table 2: _Zero-shot_ retrieval performance on out-of-domain Real Analysis (**RA**) and Number Theory (**NT**) textbooks. We show results for BERT models trained on ProofWiki.

	Edit(↓)	**BLEU(↑)**	**EM(↑)**	**F1(↑)**
*-set	58.51	7.18	18.09	97.04
*-multiset	58.09	16.68	19.23	100.0
*-halfseq	58.84	25.88	0.00	56.86
Joint	93.03	0.00	0.09	25.30
Sequential	84.30	5.48	3.78	25.61

Table 4: _Reference generation_ performance on ProofWiki. We show oracle benchmarks for correctly predicting the first half of the sequence (_*-halfseq_), the full multiset (_*-multiset_) set (_*-set)_ with random order. Metrics are computed on reference ids.

1.2 Main Results.

We overview key results here and provide further results and analysis in the talk. Table 1 shows _in-domain_ retrieval performance, meaning that each model was trained and evaluated on the same domain. The BERT models substantially outperform the TF-IDF baseline, with the joint model showing the best performance. Table 3 shows example retrievals from the joint model.

We find substantial room for future improvement on out-of-domain generalization and reference generation. As seen in Table 2, the BERT models trained on ProofWiki show worse or similar retrieval performance as TF-IDF on the Real Analysis and Number Theory textbooks, which we also found was the case for models trained on Stacks, or both ProofWiki and Stacks.

On reference generation (Table 4), the sequential model improves over using the top-5 predictions from the retrieval model, yet falls behind oracle benchmarks that only predict the correct set (_*-set_) or half of the correct sequence (_*-halfseq_), leaving much room for improvement.

1.3 Looking forward.

Overall, our results show both promising immediate use of neural models for in-domain retrieval, and open challenges for the future. In the final part of the talk we discuss future work based on using or extending NaturalProofs, as well as NLP techniques that may be of interest. We hope to promote discussion about which tasks serve as meaningful proxies, the difficulty of evaluation, and bridging informal and formal reasoning.

References

[1] Alexander A Alemi, François Chollet, Niklas Een, Geoffrey Irving, Christian Szegedy, and Josef Urban. DeepMath - Deep sequence models for premise selection. In *Advances in Neural Information Processing Systems*, pages 2243–2251, 2016.

[2] Nathan C. Carter and Kenneth G. Monks. Lurch: a word processor that can grade students' proofs. In Christoph Lange, David Aspinall, Jacques Carette, James H. Davenport, Andrea Kohlhase, Michael Kohlhase, Paul Libbrecht, Pedro Quaresma, Florian Rabe, Petr Sojka, Iain Whiteside, and Wolfgang Windsteiger, editors, *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.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[4] Deborah Ferreira and André Freitas. Natural language premise selection: Finding supporting statements for mathematical text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2175–2182, Marseille, France, May 2020. European Language Resources Association.

[5] Dongyeop Kang, Andrew Head, Risham Sidhu, Kyle Lo, Daniel S. Weld, and Marti A. Hearst. Document-level definition detection in scholarly documents: Existing models, error analyses, and future directions, 2020.

[6] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[7] Aditya Ohri and Tanya Schmah. Machine Translation of Mathematical Text. *IEEE Access*, 2021.

[8] Bartosz Piotrowski and J. Urban. Stateful premise selection by recurrent neural networks. In *LPAR*, 2020.

[9] Yiannos Stathopoulos, Angeliki Koutsoukou-Argyraki, and Lawrence Paulson. Developing a concept-oriented search engine for isabelle based on natural language: Technical challenges. 12 2020.

[10] Christian Szegedy, editor. *A Promising Path Towards Autoformalization and General Artificial Intelligence*, 2020.

[11] Qingxiang Wang, Chad Brown, Cezary Kaliszyk, and Josef Urban. 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.