

Ranked List Fusion and Re-ranking with Pre-trained Transformers for ARQMath Lab

Shaurya Rohatgi¹, Jian Wu² and C. Lee Giles¹

¹Pennsylvania State University

¹Old Dominion University

Abstract

This paper elaborates on our submission to the ARQMath track at CLEF 2021. For our submission this year we use a collection of methods to retrieve and re-rank the answers in Math Stack Exchange in addition to our two-stage model which was comparable to the best model last year in terms of NDCG'. We also provide a detailed analysis of what the transformers are learning and why is it hard to train a math language model using transformers. This year's submission to Task-1 includes summarizing long question-answer pairs to augment and index documents, using byte-pair encoding to tokenize formula and then re-rank them, and finally important keywords extraction from posts. Using an ensemble of these methods our approach shows a 20% improvement than our ARQMath'2020 Task-1 submission.

Keywords

Math Information Retrieval, Re-ranking, Math-aware search, Math formula search

1. Introduction

The ARQMath-2 lab at CLEF [1] is an effort to improve math information retrieval and aims to produce better math-aware retrieval systems. The tasks introduced by the lab serve as a benchmark for systems competing in this space. In CLEF-2020 several systems competed for both tasks [2, 3, 4, 5]. There are two tasks in the ARQMath-lab. This work presents our submission to Task-1: Answer Retrieval, where the participants are given a list of topics and are expected to return a list of relevant posts from the Math Stack Exchange collection. The challenge here is that these topics are multi-modal in the sense that they include both text and math formulas.

The Intelligent Information Systems Research Laboratory (IISRL) at Pennsylvania State University participated in the CLEF-2021 ARQMath-2 track to contribute and further push the limits of math-aware information retrieval. In addition to our approach for ARQMath-1 Lab [5] we add new ways to index by augmenting more data. We also extract keywords for better matching of the posts.

We used a combination of approaches which includes -

- keyword extraction
- augmenting data using post summary

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

✉ szr207@psu.edu (S. Rohatgi); jwu@cs.odu.edu (J. Wu); clg20@psu.edu (C. L. Giles)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

- BM25
- tf-idf
- transformer based re-ranking
- using byte-pair encoding (BPE) to rank formula separately

We merge these ranking and indexing methods using reciprocal rank fusion [6] which boosts the rankings of the posts which occur in more than one ranked list.

This approach has shown improvement in our submission for Task-1 over our previous year's submission by 20%. While keyword extraction and augmenting data helps, a major improvement is seen because of BPE for indexing and ranking formula separately hinting that transformers and their tokenization strategy are worth exploring for math formula. We also use this opportunity to showcase the BPE tokenization performance on another math formula retrieval task- NTCIR-12 [7].

The rest of the paper is structured as follows - Section 2 talks about previous work and other participant submissions from ARQMath lab in CLEF 2020; Section 3 discusses our current methodology and approach; Section 4 shows the results and analysis of our approach on Task-1 2020 as well as NTCIR-12 and finally we compare our 2021 submission to other participants and where the current system can make improvements. Finally, we conclude that the transformers might not be good at understanding standalone formulas but they are great at making connections between formulas and related text.

2. Related Work

Previous year's systems by MIRMU, zbMath, MathDowers all submitted competitive results for task-1. All systems brought in unique approaches and here we want to discuss some of them. MIRMU used an ensemble approach where they use a mix of novel approaches including CompuBERT, Soft Cosine Measure, and Formula2vec. While these approaches didn't perform well individually, their ensemble did well on task-1. We take inspiration from this approach and try different matching strategies in this work. zbMath went a more entity-based route, extracting Wikibase items from the collection and then doing manual runs using Google and Math Stack Exchange searches. They also used a mix of ElasticSearch, Doc2vec, and K-nearest neighbors to retrieve the relevant formulas. The main contribution here we think is that they do an inductive classification study on the ARQMath Topics.

MathDowers submitted the best performing system for task-1 for ARQMath Lab-1. They use a two-stage re-ranking approach where the re-ranking model is trained on the metadata attributes of the posts like upvotes, number of comments etc. In their first stage, they use Tangent-L for formula retrieval, and similar to our approach of indexing they concatenate the question and the answers before indexing. This joint post is one unit of retrieval. In our last year's submission, we show that their system performs better on formula-dependent queries (where answer relevance is more formula dependent) as compared to our system, while our system performs well in text-dependent queries.

3. Methodology

In this section, we describe our multi-stage ranking approach which fuses ranked lists from different approaches. The following subsections go through each approach in detail.

3.1. Byte Pair Encoding (BPE)

We take inspiration from the sentence piecing work demonstrated in the past [8]. Here we feed the only formula's to train a tokenizer that can look at frequently occurring characters together and then token a string accordingly. This helps it identify otherwise difficult tokens in the text.

This way of tokenizing can be exploited for formulas, as there is no strict way of tokenizing them linearly unlike text. While other methods use SLTs and OPTs we stick to a simpler solution of linearly tokenizing the formula. SLTs and OPTs have shown to perform really well for formula search but they require MathML representations, while BPE can work on straight \LaTeX . Let's consider an example to better understand BPE for formula retrieval -

$$x^2 - \frac{1}{2} \tag{1}$$

is tokenized as $x, \wedge, \{, 2, \}, -, \text{frac}, \{, 1, \}, \{, 2, \}$. Notice how *frac* is one token.

3.2. Summarizer and Posts Augmentation

A post can be written in various ways. In community question answering platforms like Math Stack Exchange the members, in addition to asking questions add their experience with the question they want to ask. For example, terms like "my assignment want me to . . .", "Please help me understand this concept" etc. These produce noise for the indexing and retrieval systems and need to be removed. Here are an example of summarization and what it can do -

Original Text : We began learning this is math analysis a week or so ago along with verifying identities, finding the exact value of trigonometric equations, and writing trigonometric equations as algebraic equations and i'm totally lost.

Summary : finding the exact value of trigonometric equations, and writing trigonometric equations as algebraic equations

The summarizer neural network uses a document-level encoder which is based on BERT [9] which is correctly able to learn the semantics of the document and obtain vector representations for its sentences. These vector representations go through a decoding step and the summary of the document is produced.

We run the summarizer for all question-answer pairs. This pair is then concatenated and indexed as a single unit of retrieval. We give this document the same id as the answer post, thus augmenting a post that is not there in the collection. This provides a question context for each answer in serving as an answer to newly posed topic questions. When this index is queried for relevant posts for the topics these documents are also returned in addition to the ARQMath collection.

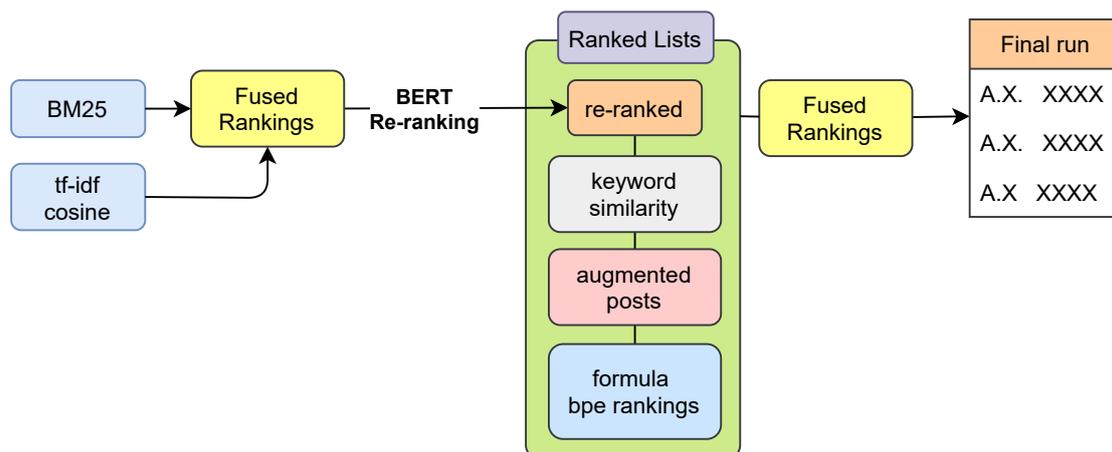


Figure 1: The architecture of the system. Ranked lists are lists of answer posts sorted by relevance returned by each approach. We fuse them using reciprocal rank fusion.

3.3. Keyword extraction

We generate keyword samples by automatically extracting noun phrases from the answer and question posts in the collection using a grammar-based chunking method. This lets us highlight the role of important keywords in the posts. We concatenate these keywords and create a separate index, which is queried against each topic. This step is an additional filtering step where we make sure that the important keywords are being matched between the posts which go into the fusion step.

3.4. PSU at ARQMath Lab 2020 Task-1

The approach we submitted to ARQMath’20 is used in addition to the ones mentioned above [5]. To summarize our earlier approach, we developed a two-stage retrieval and ranking system. The first stage was a fusion of the BM25 score and cosine distance using tf-idf. Before the second stage, we learn a language model on the math stack exchange collection. This made the language model more math-aware¹. We then use this model to re-rank the top-1000 posts retrieved by the first stage. This leads to an improvement in the overall ranking of the answers retrieved as demonstrated in the previous year’s working notes.

We use this model and add the aforementioned retrieval approaches. Figure 1 demonstrates our fusion and re-ranking architecture for ARQMath Lab-2.

3.5. Reciprocal Rank Fusion

After we get a ranked list from all the methods detailed above we merge them using Reciprocal Rank Fusion (RRF) [6] which then ranks the documents using a naive scoring formula. For a set of posts P to be ranked and a set of rankings R from different scoring schemes, for each permutation on $1 \cdots |P|$, we compute

¹<https://huggingface.co/shauryr/arqmath-roberta-base-1.5M>

$$RRFscore(p \in P) = \sum_{r \in R} \frac{1}{k + r(p)}, \quad (2)$$

where the original work suggests $k = 60$, a hyper-parameter which we keep constant.

4. Experiments

Our experiments were conducted on a 24 core machine with Intel(R) Xeon(R) Silver 4116 CPU @ 2.10GHz with 256GB of RAM with 4 RTX 2080 Ti GPUs. Default configurations were adopted in ElasticSearch(ES) and Anserini(BM25). Anserini runs on a single thread while ES uses a multi-threaded function. The size of ES Q+A and summarized posts index is 4.8GB whereas for Anserini the size is 3.2GB.

Each query takes on an average of 2.3 seconds for retrieval once everything is indexed. It should be noted that we are using a single shard in ES which makes it a little slow.

4.1. Fine-tuning math-aware language model and NTCIR-12

BPE is a subword unit model and is an unsupervised text tokenizer and detokenizer mainly for Neural Network-based text generation systems where the vocabulary size is predetermined before the neural model training [8]. We use it here as math expressions are hard to tokenize as there are no clear boundaries between tokens.

It should be noted that BPE is a linear tokenizer and might miss out on the structure which is better captured by tree-based methods. BPE is very cheap to train and requires less preprocessing of the math expressions.

In this case, we find that the vocabulary size for math is 250 tokens (excluding cardinal numbers). We feed the tokenizer all the expressions from the \LaTeX of the ARQMath collection to train it. It learns the patterns and creates a vocabulary using the co-occurrence frequency of characters in the corpus. This vocabulary is used to do the fine-tuning of the pre-trained weights of RoBERTa which is used in the ARQMath Lab-1 by PSU [5].

We build a nearest neighbor map of the formula in the collection and queried the topics against it to get a ranked list of math expressions that are visibly similar to the formula in the topic.

To understand if this method works for formulas we benchmarked it on the NTCIR-12 dataset as it is a more complete and well-studied dataset for formula retrieval. We compare it with the current state-of-the-art methods Tangent-CFT [10] and MathBERT [11] in Table 2. We also, fused the rankings for Tangent-CFT and our pre-trained model to see if our model can complement the tree-based embedding method. On further investigation, we find that the BPE suffers for partial matching of formula but is great for exact matches of expressions.

4.2. Results and Observations

We show the results of our system for Task-1 for the ARQMath-1 Lab in Table 3. Here we can see that the new system which includes data augmentation, BPE, and keyword extraction works well and shows improvement over the previous year’s approach. It is to be noted that all the

Table 1

Some examples of retrieval by BPE from the NTCIR-12 query benchmark dataset

NTCIR Query ID	MathWiki-17	MathWiki-13
Rank	$x - 1 - \frac{1}{2} - \frac{1}{4} - \frac{1}{5} - \frac{1}{6} - \frac{1}{9} - \dots = 1$	$A \oplus B = (A^c \ominus B^s)^c$
1	$x - 1 - \frac{1}{2} - \frac{1}{4} - \frac{1}{5} - \frac{1}{6} - \frac{1}{9} - \dots = 1$	$A \oplus B = (A^c \ominus B^s)^c$
2	$1 - \frac{1}{2} - \frac{1}{4} + \frac{1}{3} - \frac{1}{6} - \frac{1}{8} + \frac{1}{5} - \frac{1}{10} - \frac{1}{12} + \dots$	$A \circ B = (A \ominus B) \oplus B$
3	$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \dots$	$A \bullet B = (A \oplus B) \ominus B$

Table 2

NTCIR-12 Results (Avg. bpref@1000). H-Mean is the harmonic mean of partial relevance (L1) and full relevance score(L3). RRF-CFT+BERT is the reciprocal rank fusion of our BERT implementation and Tangent-CFT.

System	Partial	Full	H-Mean
tangent-cft	0.7134	0.5963	0.6496
Finetuned-BERT	0.5657	0.5747	0.5701
MathBERT	0.7361	0.6135	0.6692
RRF-CFT+BERT	0.7414	0.6163	0.6730

Table 3

Our best runs compared to the baselines systems for ARQMath-1 Lab. The new system achieves better results than the previous years submission by 20%.

Type	System	nDCG'	mAP'	p'@10	Run Type
Baselines	linked_results	0.303	0.210	0.418	Automatic
	arq2020-task1-a0-submission	0.250	0.100	0.186	Manual
	combined_tf_idf_tangents	0.248	0.047	0.073	Automatic
	tf_idf_task1_final	0.204	0.049	0.074	Automatic
	TangentS_Res	0.158	0.033	0.051	Automatic
Our Runs	BM25+tf+tf.BERT (2020)	0.263	0.082	0.116	Automatic
	Current system	0.317	0.116	0.165	Automatic

approaches mentioned are unsupervised and no learning is done on the previous year's task's ranked list.

5. Conclusion and Future Work

We presented an automatic system that uses techniques to shorten long posts and extract important keywords for more precise retrieval. Unsupervised ranking methods like tf-idf, BM25, and BPE with the nearest neighbor search are used here and they demonstrate robustness. Our last year's submission fused with these new techniques improves our overall NDCG' score.

In the future, we want to improve over this ranking by learning to rank. This collection has an abundance of ground truth data in the form of accepted answers. We could potentially train a model using the metadata of the posts like the number of comments, the number of upvotes, and the total number of answers to name a few; to train a learning to rank model which can re-rank the results.

Identifying the important formula for relevant retrieval from a number of formulas in a post is hard. We want to focus our efforts on identifying the significant formulas in a post and then giving a higher score to the formulas which match them rather than treating all formulas equally.

Lastly, we want to incorporate an actual tree-based formula retrieval technique like Approach0 or Tangent-S in our rankings. This can boost our relevance scores as these methods have been shown to perform better with partial relevance when matching formulas, unlike BPE.

Acknowledgments

We would like to thank members of the ARQMath lab at the Department of Computer Science at Rochester Institute of Technology for organizing this track. Special thanks to Behrooz Mansouri for providing the dataset, initial analysis of topics, and starter code to all the participants of the task; it made it easier for us to pre-process the data and jump directly to the experiments which have been presented in this work.

References

- [1] B. Mansouri, A. Agarwal, D. W. Oard, R. Zanibbi, Advancing math-aware search: The arqmath-2 lab at clef 2021, in: D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), *Advances in Information Retrieval*, Springer International Publishing, Cham, 2021, pp. 631–638.
- [2] P. Scharpf, M. Schubotz, A. Greiner-Petter, M. Ostendorff, O. Teschke, B. Gipp, Arqmath lab: An incubator for semantic formula search in zbmath open?, arXiv preprint arXiv:2012.02413 (2020).
- [3] N. Yin Ki, D. J. Fraser, B. Kassaie, G. Labahn, M. S. Marzouk, F. W. Tompa, K. Wang, Dowsing for math answers with tangent-l, in: *CEUR Workshop Proceedings*. Thessaloniki, Greece, 2020.
- [4] V. Novotný, P. Sojka, M. Štefánik, D. Lupták, Three is better than one, in: *CEUR Workshop Proceedings*. Thessaloniki, Greece, 2020.
- [5] S. Rohatgi, J. Wu, C. L. Giles, Psu at clef-2020 arqmath track: Unsupervised re-ranking using pretraining, in: *CEUR Workshop Proceedings*. Thessaloniki, Greece, 2020.
- [6] G. V. Cormack, C. L. Clarke, S. Buettcher, Reciprocal rank fusion outperforms condorcet and individual rank learning methods, in: *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2009, pp. 758–759.
- [7] R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topic, K. Davila, Ntcir-12 mathir task overview., in: *NTCIR*, 2016.
- [8] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1715–1725. URL: <https://www.aclweb.org/anthology/P16-1162>. doi:10.18653/v1/P16-1162.

- [9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
- [10] B. Mansouri, S. Rohatgi, D. W. Oard, J. Wu, C. L. Giles, R. Zanibbi, Tangent-cft: An embedding model for mathematical formulas, in: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval, ICTIR '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 11–18. URL: <https://doi.org/10.1145/3341981.3344235>. doi:10.1145/3341981.3344235.
- [11] S. Peng, K. Yuan, L. Gao, Z. Tang, Mathbert: A pre-trained model for mathematical formula understanding, ArXiv abs/2105.00377 (2021).