DPRL Systems in the CLEF 2021 ARQMath Lab: Sentence-BERT for Answer Retrieval, Learning-to-Rank for Formula Retrieval

Behrooz Mansouri[1], Douglas W. Oard[2] and Richard Zanibbi[1]

[1]Rochester Institute of Technology, NY, USA
[2]University of Maryland, College Park, USA

Abstract

This paper describes the participation of the Document and Pattern Recognition Lab from the Rochester Institute of Technology in the CLEF 2021 ARQMath lab. There are two tasks defined for ARQMath: (1) Question Answering, and (2) Formula Retrieval. Three systems were submitted for Task 1, all of which used two-stage retrieval models. First, a set of questions within the test collection that were similar to the query were found using a Sentence-BERT model that had first been trained on Quora Question Pairs and then fine tuned using duplicate question links found within the ARQMath test collection. Then in the second stage, answers given to those questions (identified using links within the collection) were ranked by one of three different similarity scores. For Task 2, five runs were submitted: one using only formula embedding, another using formula embedding followed by re-ranking based on tree-edit distance, the third run using Tangent-S, and the remaining two being alternative ways of reranking Tangent-S results using learning-to-rank techniques.

Keywords

Community Question Answering (CQA), Mathematical Information Retrieval (MIR), Math-aware search, Math formula search

1. Introduction

The ARQMath-2 lab at CLEF 2021 has the same two main tasks [1] as did the ARQMath-1 lab at CLEF 2020 [2]. In Task 1 the participants are given a new mathematical question (i.e., a question containing at least one mathematical formula) that had been posted in 2021, and are asked to return a set of relevant answers that had posted between 2010 and 2018. The other task in ARQMath is Formula Retrieval (Task 2), which takes a formula as the query, in that case the system's goal is to find a set of formula instances that are relevant to that query.

The Document and Pattern Recognition Lab (DPRL) from the Rochester Institute of Technology (RIT, USA) participated in both tasks. For Task 1, the design of our models is motivated by one aspect of user behavior that is fairly common in Math Stack Exchange (and other Community Question Answering forums).[1] When a new question is posted, the Math Stack Exchange moderators (and experienced users with high reputation scores[2]) can mark a question as a

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania
bm3302@rit.edu (B. Mansouri); oard@umd.edu (D. W. Oard); rxzvcs@rit.edu (R. Zanibbi)

[1]math.stackexchange.com/

[2]At the time of writing this paper, reputation scores above 3000 are considered.

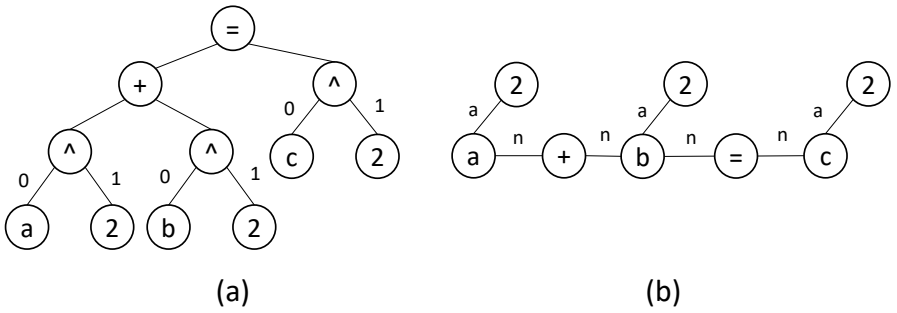

Figure 1: Formula $a^2 + b^2 = c^2$ represented as (a) Operator Tree and (b) Symbol Layout Tree.

duplicate, referring the asker to the similar question(s) that had previously been posted to Math Stack Exchange, where they can find relevant answers. Following a similar process, in all of our runs we first find similar questions (using some fully automatic technique) and then we rank only the answers given to those questions. Our systems differ in how the first and second stages of that process are implemented.

To find similar questions, we fine-tune a Sentence-BERT model [3], using both related and duplicate questions on Math Stack Exchange. For ranking the answers, we use three different scoring functions: (1) the Math Stack Exchange answer scores that are available as metadata in the test collection, (2) A computed score from a Sentence-BERT model trained on the ARQMath-1 relevance judgments that estimates the similarity between the Question and Answer pair (QASim), and (3) a combination of scores from those two approaches.

For Task 2, we modify our previous models Tangent-CFT [4] and Tangent-CFTED [5]. Tangent-CFT is an n-gram embedding model built on a linearized tree representation of formulae and Tangent-CFTED re-ranks the results from Tangent-CFT using tree-edit distance. Our two new runs for ARQMath, use learning-to-rank framework for mathematical formulae [6] trained on ARQMath-1 topics. One run is trained only on the 29 training queries from ARQMath-1, whereas the other was trained on all 77 ARQMath-1 formula queries. The Tangent-S [7] system is our last (baseline) run. All our runs use only formula matching to find relevant formulae, with no use of the text surounding those formulae in the question from which the query was extracted or in the test collection post from which a potentially relevant formula instance was extracted. Our models make use of both the Operator Tree (OPT) and Symbol Layout Tree (SLT) representations of formulae, one encoding the syntax and the other appearance of a formula. Figure 1 shows the OPT and SLT representations for the formula $a^2 + b^2 = c^2$. In the OPT representation, the edge labels for non-commutative operators indicate argument position. In the SLT representation, the edge labels show the spatial relationship between the formula elements. For instance, the edge label 'a' shows that the number '2' is located above the variable 'a', while the edge label 'n' shows operator '+' is located next after 'a'.

In this paper, we first describe our runs in Task 2, then discuss our proposed models for Task 1, and finish with conclusions.

Table 1
Tuples created on the formula $a^2 + b^2 = c^2$ (from Figure 1. Each tuple has four elements: *(parent, child, path, path-from-root[PFR])*. **V!, N!, O!,** and **U!** are node types (operators like '+' have no SLT node type), 'eob' end-of-baseline, and '-' an empty path.

SLT Tuples				OPT Tuples			
(Parent,	Child,	Path,	PFR)	(Parent,	Child,	Path,	PFR)
(**V!** a,	**N!** 2,	a,	-)	(**U!** eq,	**U!** plus,	0,	-)
(**N!2**,	eob,	n,	a)	(**U!** plus,	**O!** SUP,	0,	0)
(**V!** a,	+,	n,	-)	(**O!** SUP,	**V!**a,	0,	00)
⋮				⋮			

2. Task 2: Formula Retrieval

In Task 2, one formula inside each topic from Task 1 is selected and the participants are asked to return a set of relevant formula instances from the questions and answers in the collection. As described in the ARQMath 2021 overview [8], to decide the relevance degree of a formula, the context in which the topic and retrieved formula appear is important. In all our models, we focus only on the structural matching of the formula, and text is ignored. All our models make use of both OPT and SLT representations. Different approaches such as [4, 7, 9] have found this beneficial for system effectiveness. Next, we describe our five runs.

2.1. Tangent-S

Tangent-S[3] [7] was reported by the organizers as the Task-2 baseline system in both ARQMath 2020 and 2021, and we also use components or scores from Tangent-S in all of our runs. In this system first a set of candidates are retrieved based on tuple similarity. Using depth-first traversals, tuples with 4 elements are created as (parent, child, path, path-from-root [PFR]). The parent and child are the node values in the form of *Type!Value*, where type can take values such as Variable (V) or Number(N) and value shows the variable name or the numeric value. Path shows the set edge labels visited connecting parent to child. Path-from-root shows edges labels visited when moving from the root node to the parent node. Table 1 shows the tuples created from SLT and OPT representations of the formula $a^2 + b^2 = c^2$ from Figure 1. As the first node is the root of the tree, the path-from-root is empty. The second tuple is showing that the node with type *Number* and value '2' has no children and we have reached end of baseline (eob). The default edge label for the eob is 'n'. After the tuples are generated, the harmonic mean of recall and precision for matched tuples is used for ranking.

To re-rank the candidates, three similarity scores are considered. The Maximum Subtree Similarity (MSS) is computed from the largest connected match between query and candidate formulae obtained using a greedy algorithm, evaluating pairwise alignments between trees using unified node values. Two other similarity features used in this system are query node matching after alignment, either with or without -Type unification. The SLT and OPT results

[3] https://github.com/MattLangsenkamp/tangent-s

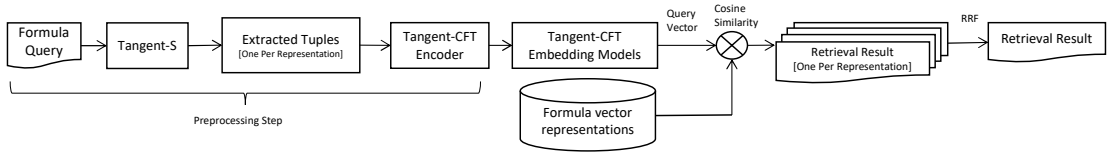

Figure 2: Overview of the Tangent-CFT2 Retrieval Process. The tuples for different representations are extracted using Tangent-S. After the tuples are encoded using the Tangent-CFT model, vector representations of a query formula are obtained. These vectors are compared with the vector representation of formulae in the collection by using cosine similarity, and top-1000 most similar formulae are returned. Finally, retrieval results from different representations are combined using modified Reciprocal Rank Fusion (RRF) to get the final retrieval result.

are next combined via linear regression over alignment measures from each representation to produce final similarity scores.

2.2. Tangent-CFT2

The Tangent-CFT model [4] was the first embedding model introduced for mathematical formula to use both SLT and OPT representations. In addition to the full representations (which include both the type and the value of a node), this model also employed unification on the SLT to produce a representation called SLT-Type. In the SLT-Type tree representation, only the type of each node was represented; the corresponding values were ignored. For each of the three representations, the model would then convert the tree representation to a vector, and then add the 3 vectors to get the final representation of a formula. Our the process thus has the following steps (refer to [4] for further details):

- **Tuple Extraction**: Presentation MathML and Content MathML representations (from LaTeXML[4]) are used as a basis from which to generate internal SLT and OPT formula representations. Built using depth-first traversals with the Tangent-S system, these internal tree representations are strings consisting of a sequence of tuples, as described above for the Tangent-S system. The only difference for Tangent-CFT is that the path-from-root element produced by Tangent-S is ignored.
- **Tuple Encoding**: The tuples are then tokenized and enumerated. The tokenization is based on type and value. For example the tuple (V!a, N!2, a), which represents a^2, will be tokenized to {'V', 'a', 'N', '2', 'a' }.
- **Training Embedding Models with fastText**: Mathematical formulae are diverse, with relatively few training examples being available for any particular formula. For this reason, Mansouri et al chose to apply [10] the multi-scale fastText [11] n-gram embedding model get vector representations for each tuple. As the name suggests, fastText was originally designed for text. To apply it to math, each token is treated as it it were a text character, every whole tuple is treated as a text word, and the generated set of tuples is

[4] https://dlmf.nist.gov/LaTeXML/

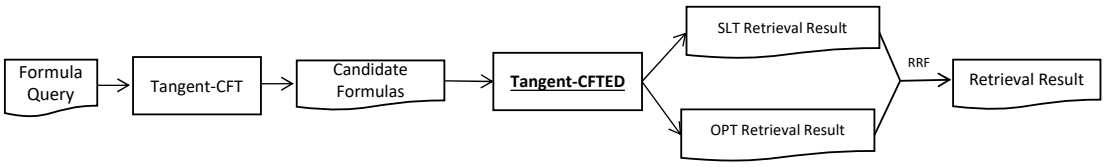

Figure 3: Overview of the Tangent-CFT2TED Retrieval Process. The candidate formulae are selected with the Tangent-CFT model. Then, using tree-edit distance, the formulae are re-ranked using SLT and OPT representations. The results are combined using Reciprocal Rank Fusion (RRF).

treated as a text sentence. The final vector representation for a formula is thus obtained by averaging the fastText representations for its individual tuples.

While the overall pipeline for Tangent-CFT2 is the same as for Tangent-CFT, two modifications were made to the architecture:

- **Formula Representations.** In Tangent-CFT2 we add another representation to the previous model by considering the OPT-Type. This representation is similar to SLT-Type, but for the operator tree.
- **Combining the Results.** After ARQMath 2020, we modified the result combination part of the Tangent-CFT system. In the previous system, the 3 vectors from each representation were added to get the final vector representation of a formula. In Tangent-CFT2, with each of the four representations of SLT and OPT (full and -Type), first, the top-k results are retrieved. This is done by computing the cosine similarity between the query vector representation and the vector representation of formulae in the collection. Then the four results are combined using modified Reciprocal Rank Fusion [12] with the following formula:

$$RRFscore(f \in F) = \sum_{m \in M} \frac{s_m(f)}{k + r_m(f)} \tag{1}$$

where f is a set of formulae to be ranked, M is a set of models, and s_m and r_m are the scores and the rank, respectively, of the retrieved formula by model m. The top-1000 results from each representation are computed as the cosine similarity between the vectors.

Figure 2 shows the overall architecture of Tangent-CFT2.

2.3. Tangent-CFT2TED

The Tangent-CFTED model was introduced in ARQMath 2020 [5]. This model first retrieves a set of candidate formulae using the Tangent-CFT model. It then re-ranks them using Tree-Edit Distance (TED), similarly to Kamali and Tompa[13]. Figure 3 shows an overview of this system. In this section, we first review the Tangent-CFTED model and then describe the changes in the new model.

Tangent-CFTED uses tree-edit distance to compare the similarity between two formulae using their tree representations. Tree-edit distance is the minimum cost of converting one tree to the other using a set of edit operations. In this work, we consider three edit operations: insertion, deletion, and substitution. Each operation has a unique weight. These weights are similar in both versions, learned on NTCIR-12 [14] topics. To calculate tree-edit distance, only the node values are used; the edge labels are ignored. The similarity measure is defined as inverse tree-edit distance as follows:

$$sim(T_1, T_2) = \frac{1}{TED(T_1, T_2) + 1}. \tag{2}$$

In our model, we use both SLT and OPT representations and re-rank the candidates with tree-edit distance. The results are then combined using RRF with equation 1. For more details refer to [4]. There are two modifications made in the new version:

- **Selecting the candidates.** The candidate formulae are selected using Tangent-CFT2.
- **Combining the Results.** To combine the re-ranked results from the SLT and OPT representations, we again use RRF (equation 1). In the previous version, the results were combined linearly with weights learned on the NTCIR-12 [14] dataset.

2.4. Learning-to-Rank

Broadly speaking, three main approaches have been used for the formula retrieval task: full-tree matching, sub-tree matching, and embedding models. In our learning-to-rank framework, we make use of all these approaches, using results from instances of each approach to create features for the SVM-rank [15] system for supervised learning to rank. We trained SVM-rank two ways, once with the 29 ARQMath-1 training queries (*LtR29*), and the other time with all 77 queries from the ARQMath-1 collection (*LtRall*). The penalty for misclassification during training (C) is 0.01, and the tolerance for termination criterion (epsilon) is 0.001. We computed the following kinds of similarity measures as features:

- Tuple matching scores
- Maximum Sub-tree Similarity (MSS)
- Node Matching scores
- Unweighted tree edit distance scores
- Weighted tree edit distance scores
- Cosine similarity from Tangent-CFT model

All features other than MSS were calculated on both OPT and SLT representations, with and without unification. The MSS features were only calculated for the unified SLT-Type and OPT-Type representations. The first three kinds of similarity features are from the Tangent-S model, which uses sub-tree matching. The tree edit distance features were calculated with the Tangent-CFTED system. For the weighted tree edit distance scores, we use the same weights that Tangent-CFTED uses for retrieval.[5]

[5]Note that for learning to rank we used the original Tangent-CFT and Tangent-CFTED models, not the Tangent-CFT2 and Tangent-CFT2TED versions that we used for single-system submissions this year.

Table 2
DPRL runs for Task 2 on ARQMath-1 (45) and ARQMath-2 (58) topics. Tangent-S is the baseline system.

			Evaluation Measures					
			ARQMath-1			ARQMath-2		
	Data	Primary	nDCG′	MAP′	P′@10	nDCG′	MAP′	P′@10
Tangent-S	Math	✓	0.691	0.446	0.453	**0.492**	**0.272**	0.419
LtR29	Math		0.736	0.522	0.520	0.454	0.221	0.317
LtRall	Math	✓	**0.738**	**0.525**	**0.542**	0.445	0.216	0.333
Tangent-CFT2TED	Math		0.648	0.480	0.502	0.410	0.253	**0.464**
Tangent-CFT2	Math		0.607	0.437	0.480	0.338	0.188	0.297

2.5. Results for Submitted Runs

Tables 2 and 3 show the DPRL runs results on Task 2, both for our official submitted runs (Table 2), and for runs obtained after correcting errors in how our systems were run (Table 3). In this section we summarize results for our submitted runs, shown in Table 2.

ARQMath-1 Results. For the ARQMath-1 topics, our learning-to-rank framework has better effectiveness compared to the Tangent-S system, in particular for P′@10 which is higher for all of our runs. Interestingly, our learning-to-rank framework shows similar effectiveness when trained on all training and test topics (LtRall) versus trained on only training topics (LtR29). Using a one-way analysis of variance (ANOVA) test with a .05 significance level, the differences between our runs were not significant in terms of P′@10 and MAP′. However, with nDCG′, the differences between Tangent-CFT2 and both our learning-to-rank models are significant (using posthoc Tukey HSD Test, $p < 0.05$).

Tangent-CFT2, Tangent-CFT2TED, and Tangent-S take different approaches for formula retrieval using embedding, full-tree, and sub-tree matching. Our learning-to-rank model uses SVM-rank to linearly combine similarity measures from each of these models, in order to overcome their individual limitations.

In our next analysis, we compare three runs from the Tangent family on the ARQMath-1 topics, and show how learning-to-rank can help. Tangent-CFT2, being an n-gram embedding model, focuses on retrieving formulae that share common n-grams with the input query. This can be beneficial for formulae such as $\sum_{n=0}^{N} nx^n$, which are small, and perhaps users are sensitive to the variables or numbers they use in their formula. Both Tangent-S and Tangent-CFT2TED return non-relevant formulae such as $\sum_{n=0}^{N}(-1)^n x^n$ in their top-10 results, which have SLTs and OPTs that are similar to the query, but are not relevant. For this query, the P′@10 for Tangent-CFT2 was 0.9, 0.4 for Tangent-CFT2TED, and 0.6 for Tangent-S. Tangent-CFT2TED uses tree-edit distance as a full-tree matching score.

Looking at full trees provides better results for formulae such as:

$$\emptyset, \{1\}, \{2\}, \{1,2\}, \{3\}, \{1,3\}, \{2,3\}, \{1,2,3\}, \{4\}, \ldots$$

where partial matches are unlikely to provide useful information. The P′@10 values for this query for Tangent [-CFT2TED,-CFT2,-S] are 0.6, 0.4 and 0.3, respectively. Finally, Tangent-S is a

system using sub-tree matching, and for complex formulae such as:

$$\iint_V f(x,y)dx\,dy = \iint_Q f(\Phi(u,v)\left|\frac{\partial\Phi}{\partial u}\times\frac{\partial\Phi}{\partial v}\right|$$

finding sub-matches can also be useful, returning highly relevant formulae such as:

$$\iint\limits_{D_{x,y}} f(x,y)dxdy = \iint\limits_{D_{u,v}} f\left(T(u,v)\right)|J(u,v)|dudv,$$

that the other two models did not return in their top-1000 results. Tangent-S has P′@5 of 0.5 for this formula, whereas this value is 0.3 for both Tangent-CFT2 and Tangent-CFT2TED.

From these examples, we can see that each of these models have their strengths and limitations. With our learning-to-rank model, we re-rank Tangent-S results using similarity scores from multiple retrieval models. For example, for the query $\text{lcm}(n_1,n_2) = \frac{n_1 n_2}{\text{gcd}(n_1,n_2)}$ Tangent-S ranks non-relevant formulae such as $L = \text{lcm}(n_1,n_2,\ldots,n_k)$, that share sub-trees with the query in its top-10 results. Using our proposed learning-to-rank model, relevant formulae such as $\text{lcm}(a,b) = \frac{a\cdot b}{\text{gcd}(a,b)}$ are pushed to the top-10 results. As can be seen, the first formula can be converted to the second with a pair of substitutions (a for n_1, b for n_2) and removing the multiplication dot in the second formula.

The learning to rank models use only formula tree similarity features. However, there are formulae that need more features and processing. For instance, for the query:

$$Empty(x) \iff \nexists y(y \in x)$$

there are relevant formulae such as $\vdash \exists x \forall y(y \notin x)$ that may not necessarily share SLT or OPT structure with the query. Perhaps using canonicalization methods [16] can improve effectiveness for these queries to convert them to one unified format. While we focused on structural similarity features, there are still formulae for which the effectiveness is low. Textual features are another missing part of our current model. There are queries such as $\frac{df}{dx} = f(x+1)$, appearing in a question related to differential equations, for which returning a structurally similar formula such as $\frac{dy}{dx} = f(x)$ is considered non-relevant due to its appearance in a different context (a post on another topic). We consider exploring these features in our future work.

ARQMath-2 Results. The results for ARQMath-2 for our systems and the baseline are substantially lower than for ARQMath-1, with the baseline outperforming all of our models in nDCG′ and MAP′. This was due in part to changes in relevance assessment for Task 2 (see the ARQMath-2 working notes overview paper for details [17]), and to errors made while computing our runs, described in the next Section.

2.6. Corrected Unofficial ARQMath-2 Post-Hoc Runs

After ARQMath 2021, we determined that we had executed our embedding models incorrectly for ARQMath-2 topics, which impacted all of our official Task 2 runs. Tangent-CFT2 uses embeddings, Tangent-CFT2TED re-ranks Tangent-CFT2 results, and our learning to rank models

Table 3
DPRL Corrected Runs for Task 2 on ARQMath-1 (45) and ARQMath-2 (58) topics. Tangent-S is the baseline system. Runs for ARQMath-2 are corrected (*).

			Evaluation Measures					
			ARQMath-1			ARQMath-2*		
	Data	Primary	nDCG′	MAP′	P′@10	nDCG′	MAP′	P′@10
Tangent-S	Math	✓	0.691	0.446	0.453	0.492	0.272	0.419
Tangent-CFT2TED	Math		0.648	0.480	0.502	**0.580**	**0.381**	**0.545**
Tangent-CFT2	Math		0.607	0.437	0.480	0.565	0.364	0.516
LtRall	Math	✓	**0.738**	**0.525**	**0.542**	0.548	0.342	0.539
LtR29	Math		0.736	0.522	0.520	0.548	0.333	0.517

use cosine similarities over embedding vectors as features. We fixed the issue and recalculated the effectiveness measures shown in Table 3.

After correction, the learning-to-rank models improve the Tangent-S (baseline) results similarly for both ARQMath-1 and ARQMath-2 topics (roughly 5-6% for nDCG′, 7-8% for MAP′, and 9-12% for P′@10). For both topic sets, training our learning to rank models using all ARQMath-1 topics versus just the 29 ARQMath-1 training topics produces similar results. We choose to re-rank Tangent-S results for our learning to rank models, as Tangent-S had the best performance for Task 2 systems at ARQMath-1. However, for ARQMath-2, this is not the case. Re-ranking results from another system may have provided better re-ranked results. Our strongest results were obtained by the Tangent-CFT2TED system, which had better effectiveness than Tangent-CFT2. Tangent-CFT2TED reranks Tangent-CFT2 results using SLT and OPT tree edit distances with RRF (see above).

Even after correction, all of our runs have lower nDCG′ and MAP′ measures on ARQMath-2 topics than on ARQMath-1 topics. However, our highest $P'@10$ is nearly identical, increasing just slightly (by 0.3%) for Tangent-CFT2TED for ARQMath-2 topics vs. ARQMath-1 topics. Note that our proposed models are all based on formula similarity and the context is ignored: there are several queries where our models retrieve formulas with identical or nearly identical SLT/OPT representations, but they are assessed as low relevance or not relevant due to differences in the posts where query and retrieved formulas appear (e.g., different data types and ranges for variables). Some rather small changes can cause formulas to be deemed not-relevant, as illustrated in Table 4.

3. Task 1: Answer Retrieval

In Task 1, the goal is to find relevant answers to a mathematical question. The topics are selected among questions posted on Math Stack Exchange in 2020. The answers in the collection are posted from 2010 to 2018.

We had 3 runs for Task 1, all following a two-step retrieval model. First, similar questions are retrieved. Then, all answers given to similar questions are ranked using 3 different scoring functions. In all our runs, the mathematical formulae are represented using their original LaTeX strings. Finally, the answers are sorted in descending order by their vote scores. We explain our two-step retrieval model next.

Table 4
ARQMath-2 Task 2 queries where formulae with similar SLT/OPT representations are assessed as having low relevance, or being not relevant. Retrieved formulas are from the Tangent-CFT2TED system.

Query	Retrieved Formula	Relevance
$\alpha_1 + \alpha_2 + \alpha_3 = 3$	$\alpha_1 + \alpha_2 + \alpha_3 = 1$	Not Relevant
$x^n + y^n + z^n$	$x^n + y^n + z^n$	Low
$\cos\frac{\pi}{5} - \cos\frac{2\pi}{5} = \frac{1}{2}$	$\cos\frac{\pi}{5}\cos\frac{2\pi}{5} = \frac{1}{4}$	Not Relevant
$A = \bigcup_{n=1}^{\infty} A_n$	$A = \bigcup_{n=1}^{+\infty} A_n$	Low
$R(n_1, ..., n_c)$	$R(u_1, ..., u_n)$	Not Relevant

3.1. STEP 1: Finding Similar Questions

Math Stack Exchange provides links to related and duplicate questions. The related questions have a similar topic, but they are not exactly the same question. The duplicate questions are tagged by the Math Stack Exchange moderators of users with high reputation scores. A duplicate question is a newly posted question that has been asked before on Math Stack Exchange.

In our retrieval models, to first identify similar questions to a topic, we used the Sentence-BERT Cross-Encoders with the pre-trained model on the Quora question pairs dataset.[6] The model was trained on over 500,000 sentences with over 400,000 pairwise annotations indicating whether two questions are a duplicate or not. Using this model, we did two-step fine-tuning. First, we trained the model on both duplicate and related questions. Then, another fine-tuning was done, using only the duplicate questions. For our training, we used the posts provided in the ARQMath collection (from 2010 to 2018). In the first fine-tuning, 358,306 pairs, and in the second, 57,670 pairs were used. In both cases, half of the pairs were positive samples and the other half were the negative ones, chosen randomly from the collection.

To train both models, we used multi-task learning, considering two loss functions: constrastive [18] and multiple negatives ranking loss [19]. The constrastive loss function minimizes the distance between positive pairs and maximizes it for negative ones, making it suitable for classification tasks. The multiple negatives ranking loss function, however, considers only positive pairs and minimizes the distance between positive pairs out of a large set of possible candidates. We set the batch size to 64 and number of training epochs to 20. The maximum sequence size was set to 128.

Figure 4(a) shows the Cross-Encoder model trained for finding similar questions. In the first fine-tuning, a question title and body are concatenated. In the second fine-tuning, however, we considered the same process for training, with three different inputs:

- Using the question title, with a maximum sequence length of 128 tokens.
- Using the first 128 tokens of question body.
- Using the last 128 tokens of question body.

[6] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

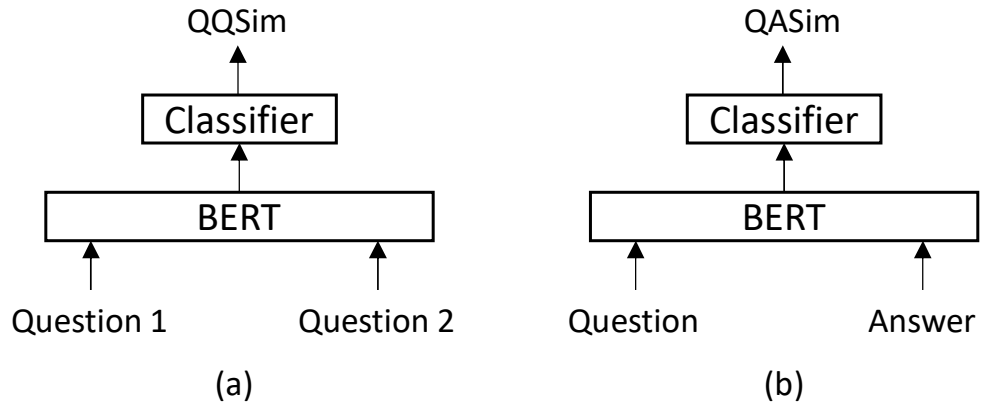

Figure 4: Sentence-BERT Cross-Encoder for identifying similar questions (a) and similarity of question and answer (b). The classifier gives a probability of relevance.

To find a similar question, we use the three models to separately retrieve the top-1000 most similar questions. The retrieved results are then combined using RRF as shown in Eq. 1. We call this similarity score Question-Question Similarity (**QQSim**).

3.2. STEP 2: Finding Related Answers

After similar questions are found for a topic, all the answers given to them are compiled and ranked based on the question and answer similarity. In each run, we used a different similarity function as follows:

1. **Math Stack Exchange score [MathSE].** Each post on Math Stack Exchange (MathSE) has a score given by the users. The score is the difference between the positive and negative votes given to that post. In our first run, we simply consider this score as an indicator of answer relevance. We used MinMax normalization to map answer scores to values between 0 to 1. Therefore, our final relevance score between a question and a candidate answer is calculated as:

$$Relevance(Q_T, A) = QQSim(Q_T, Q_A) \cdot MathSE_{score}(A) \tag{3}$$

   where the Q_T is the question topic and A is the candidate answer and Q_A is the question to which answer A was given.
2. **Two-step Hierarchical Sentence-BERT [QASim].** ARQMath-1 results showed not all the answers with a high score on Math Stack Exchange are relevant. An example is shown in Table 5.
   Therefore, in our second run, we train Sentence-BERT Cross-encoder fine-tuned on question and answer pairs from ARQMath-1 topics and their assessed hits. Our pre-trained model is Tiny-BERT with 6 layers trained on the "MS Marco Passage Reranking" [20] task. The inputs are triplets of (Question, Answer, Relevance), where the relevance is a number between 0 and 1. In ARQMath-1 evaluation [21], high and medium relevance degrees were considered as relevant for precision-based measures. In our training, we

Table 5
An example of accepted answer for a question similar to ARQMath topic, assessed as Non-relevant.

ARQMath Topic Title	Finding the last two digits of $9^{9^{9^{\cdot^{\cdot^{\cdot^{9}}}}}}$ (nine 9s)
Relevant Question Title	The last two digits of 9^{9^9}
Answer	At this point, it would seem to me the easiest thing to do is just do 99 mod 100 by hand. The computation should only take a few minutes. In particular, you can compute 93 and then cube that.

Table 6
DPRL runs for Task 1 on ARQMath-1 (77) and ARQMath-2 (71) topics. The "linked MathSE posts" is a baseline system.

			Evaluation Measures					
			ARQMath-1			ARQMath-2		
	Data	Primary	nDCG′	MAP′	P′@10	nDCG′	MAP′	P′@10
Linked MathSE posts	n/a	✓	0.303	0.210	**0.418**	0.203	0.120	**(0.282)**
QASim	Both		0.417	0.234	0.369	**0.388**	**0.147**	**0.193**
RRF	Both	✓	**0.422**	**0.247**	**0.386**	0.347	0.101	0.132
MathSE	Both		0.409	0.232	0.322	0.323	0.083	0.078

consider a relevance score of 1 for answers assessed with high or medium, 0.5 for low, and 0 for non-relevant. As for the cross-encoder for Question-Question similarity, we use the LaTeX representation for formulae. The input question is the concatenation of the question title and body. We use a batch size of 64, with 20 epochs and a maximum sequence length of 128. We keep our loss functions similar to our Question-Question model, using multi-task learning by constrastive and multiple negatives ranking loss functions. After the training, the cross-encoder outputs the similarity of question and answer, called **QASim** as shown in Figure 4(b). In this run, after similar questions are found, the model predicts the similarity of question and answer pair, **QASim**. Our final ranking score considers two similarity scores; between the question and answer, and also the similarity between the topic question and the question to which the answer is given (calculated in step 1). The ranking function is:

$$Relevance(Q_T, A) = QQSim(Q_T, Q_A) \cdot QASim(Q_T, A) \tag{4}$$

where Q_A is the question to which answer A was given.

3. **Combined model [RRF].** In our last run, we combine the similarity scores obtained from the two previous runs using RRF as given in equation 1.

3.3. Results

Table 6 shows our run results on ARQMath topics for Task 1. Using a one-way analysis of variance (ANOVA) test with a .05 significance level, none of our runs on ARQMath-1 and ARQMath-2 topics were significantly different in any of the effectiveness measures. The top-1000 results returned by all three runs differ only in their rank ordering. This is due to the effect

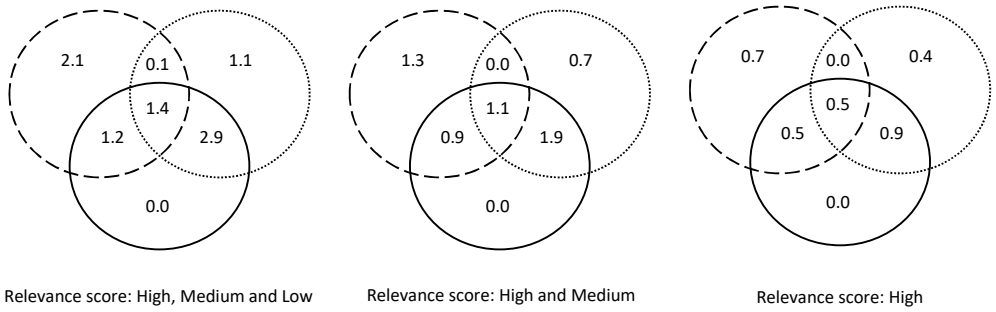

Figure 5: Venn diagram for average relevant answers retrieved by DPRL runs, on top-10 assessed hits. The dashed circle shows RRF, the dotted circle QASim and the circle with straight line indicates MathSE run. The summation of numbers in one circle shows is the average P′@10 for that system. Intersections between pairs of system are the average number of relevant hits shared by those two systems *not* in the intersection for all three systems. Numbers associated with only one circle/system are the average number of hits relevant hits found only by each individual system.

of question-question similarity in all our runs on the final similarity score. However, looking at intersections of the top-10 annotated results for each run in Figure 5 (averaged over 77 topics), our first and second runs, which use different scoring functions for the candidate answers, are able to find a set of relevant answers the the other run cannot find. When combining the results, these relevant answers are left out. For example, when considering all relevance degrees, on average there are 2.1 relevant answers retrieved by our first run (that considers only the Math Stack Exchange score), which are not included in the combined results (run RRF). Therefore, for future work, we might consider a different strategy for combining the results.

ARQMath provides different types of topics. Topics are categorized based on their difficulty into hard, easy and medium. Another grouping is based on whether the topic is dependent on the text, formula or both. The last category divides the questions based on their subject into concept, computation and proof. We separate topics based on these categories and calculated P′@10 for each group. The results are shown in Table 7. As shown in this table, re-ranking results with the Cross-Encoder trained on ARQMath-1 topics can improve effectiveness for text-dependent questions. The same effect can be seen for topics related to concepts. In this category, 40% of topics are text-dependent and the other are dependent on both formula and text. In contrast to concept-related questions, 50% of questions related to computation are formula-dependent, causing low effectiveness for our models.

For ARQMath-2 topics, we see a drop in all effectiveness measures, including for the baseline. The increase in the ratio of topics that depend upon both text and formula may partly explain this.

4. Conclusion

This paper describes the DPRL runs for the ARQMath lab at CLEF 2021. Five runs were submitted for the Formula Retrieval task. For ARQMath-2, our initial formula retrieval runs were computed incorrectly. In our corrected runs, the Tangent-CFT2TED model did better [17]. Our learning-to-

Table 7
P′@10 values for DPRL runs one different categories of questions in Task 1. There are 77 topics in ARQMath-1 and 71 in ARQMath-2 Task 1.

	Difficulty			Dependency			Subject		
	Hard	Medium	Easy	Formula	Text	Both	Computation	Concept	Proof
Topic Count (ARQMath-1):	24	21	32	32	13	32	26	10	41
QASim	0.342	0.376	0.384	0.288	0.638	0.341	0.296	0.520	0.378
RRF	0.383	0.371	0.397	0.316	0.638	0.353	0.296	0.540	0.405
MathSE	0.346	0.314	0.309	0.284	0.438	0.313	0.246	0.370	0.359
Topic Count (ARQMath-2):	19	20	32	21	10	40	25	19	27
QASim	0.184	0.115	0.247	0.300	0.130	0.153	0.204	0.116	0.237
RRF	0.105	0.065	0.191	0.181	0.140	0.105	0.132	0.100	0.156
MathSE	0.053	0.050	0.109	0.076	0.120	0.068	0.056	0.084	0.093

rank models consistently improved results, but re-ranking the baseline Tangent-S runs produced substantially weaker results for ARQMath-2 than ARQMath-1 topics in terms of nDCG′ and MAP′ metrics, perhaps because the initial Tangent-S results were weaker for ARQMath-2. Our models retrieved formulae with similar or identical Symbol Layout Tree and Operator Tree representations for some queries. However, when formulae appeared in a different context than the formula query, they could be assessed as being of low relevance, or not relevant at all.

For the Answer Retrieval task, three runs were submitted. In our runs, first, a set of similar questions are found for a topic, and then answers given to them are ranked by criteria that are specific to each run. To find similar questions we used Sentence-BERT Cross-encoder fined-tuned on the ARQMath Math Stack Exchange collection. For ranking candidate answers, we used three approaches: (1) using Math Stack Exchange answer scores, (2) using similarity of question and answer with Cross-encoder model trained on ARQMath-1 assessment, and (3) using Reciprocal Rank Fusion to combine the two previous scores. The final relevance score is calculated as the multiplication of question-question similarity and question-answer similarity score. These runs were competitive (obtaining the second-highest nDCG′ for submitted runs [17]), even though we treated formulas represented in LaTeX as text.

For future work, in the Formula Retrieval task, we aim to consider other similarity features such as spatial features [22], to improve our learning-to-rank results. Also, text features are ignored in our current formula retrieval models, and we aim to include them. For the Answer Retrieval task, our work is in an early stage, and there are several possible directions. First, our models are mostly trained with default parameters from the Sentence-BERT model and we plan to do a grid search on our models' parameters. Second, we represent formulae as LaTeX strings, and would like to use formula structure features within our model. One possible approach is replacing formulae by their concept name where available, similar to identifying synonymns in text search. For example, the formula $a^2 + b^2 = c^2$ could be replaced with "Pythagorean Theorem".

We have found ARQMath-2 to be an excellent opportunity to further develop our ideas, and we look forward to ARQMath-3!

Acknowledgments

We would like to thank Matt Langsenkamp for providing the Tangent-S code on GitHub.
This material is based upon work supported by the Alfred P. Sloan Foundation under Grant No. G-2017-9827 and the National Science Foundation (USA) under Grant No. IIS-1717997.

References

[1] B. Mansouri, A. Agarwal, D. Oard, R. Zanibbi, Advancing Math-Aware Search: The ARQMath-2 lab at CLEF 2021, in: European Conference on Information Retrieval, Springer, 2021.
[2] B. Mansouri, A. Agarwal, D. Oard, R. Zanibbi, Finding old answers to new math questions: the ARQMath lab at CLEF 2020, in: European Conference on Information Retrieval, Springer, 2020.
[3] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-Networks, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Association for Computational Linguistics, 2019.
[4] 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, 2019.
[5] B. Mansouri, D. W. Oard, R. Zanibbi, DPRL systems in the CLEF 2020 ARQMath lab, in: Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum, 2020.
[6] B. Mansouri, R. Zanibbi, D. W. Oard, Learning to rank for mathematical formula, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021.
[7] K. Davila, R. Zanibbi, Layout and semantics: Combining representations for mathematical formula search, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017.
[8] B. Mansouri, R. Zanibbi, D. W. Oard, A. Agarwal, Overview of ARQMath-2 (2021): Second CLEF lab on answer retrieval for questions on math, in: International Conference of the Cross-Language Evaluation Forum for European Languages, LNCS, Springer, 2021.
[9] G. Y. Kristianto, G. Topic, A. Aizawa, Mcat math retrieval system for ntcir-12 mathir task., in: NTCIR, 2016.
[10] B. Mansouri, R. Zanibbi, D. W. Oard, Characterizing searches for mathematical concepts, in: 2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL), IEEE, 2019.
[11] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Transactions of the Association for Computational Linguistics 5 (2017).
[12] 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.
[13] S. Kamali, F. W. Tompa, Retrieving documents with mathematical content, in: Proceed-

ings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2013.
[14] R. Zanibbi, A. Aizawa, M. Kohlhase, I. Ounis, G. Topic, K. Davila, Ntcir-12 mathir task overview., in: NTCIR, 2016.
[15] T. Joachims, Training linear SVMs in linear time, in: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006.
[16] I. Normann, M. Kohlhase, Extended formula normalization for ε-retrieval and sharing of mathematical knowledge, in: Towards Mechanized Mathematical Assistants, Springer, 2007.
[17] B. Mansouri, R. Zanibbi, D. W. Oard, A. Agarwal, Overview of arqmath-2 (2021):second clef lab on answer retrieval for questionson math (working notes version), in: Proc. International Conference of the Cross-Language Evaluation Forum for European Languages, CEUR, 2021. Online.
[18] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), volume 1, IEEE, 2005.
[19] M. Henderson, R. Al-Rfou, B. Strope, Y.-H. Sung, L. Lukács, R. Guo, S. Kumar, B. Miklos, R. Kurzweil, Efficient natural language response suggestion for smart reply, arXiv preprint arXiv:1705.00652 (2017).
[20] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, L. Deng, MS MARCO: A human generated machine reading comprehension dataset, in: CoCo@ NIPS, 2016.
[21] R. Zanibbi, D. W. Oard, A. Agarwal, B. Mansouri, Overview of ARQMath 2020: CLEF lab on answer retrieval for questions on math, in: International Conference of the Cross-Language Evaluation Forum for European Languages, Springer, 2020.
[22] R. Avenoso, Spatial vs. Graph-Based Formula Retrieval, Master's thesis, Rochester Institute of Technology, 2021. URL: https://scholarworks.rit.edu/theses/10784.