

ARQMath CLEF 2020

Evaluation Protocols

Organizers:

**Behrooz Mansouri,
Wei Zhong,
Anurag Agarwal,
Douglas Oard,
Richard Zanibbi**

bm3302@rit.edu
wxz8033@rit.edu
axasma@rit.edu
oard@umd.edu
rlaz@cs.rit.edu

Contents

Evaluation Task 1: Answer Retrieval	3
Pooling	3
Annotation	4
Evaluation Task 2: Formula Retrieval	6
Pooling	6
Annotation	7
Effectiveness Measure	9

Evaluation Task 1: Answer Retrieval

Pooling

For answer retrieval task, the participants will submit their retrieval results file in a TSV (Tab-separated values) file, with the following columns:

Query_Id Post_Id Rank Score Run_Number

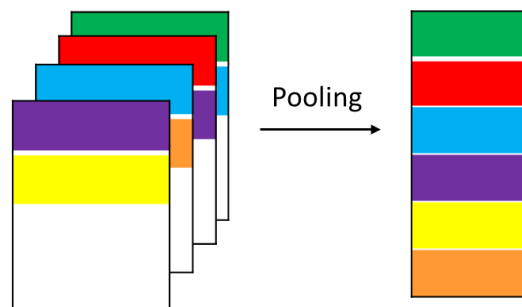
- **Query_Id.** This column shows the query id. For Task 1 all the query ids start with A, follow by a dot and then the query id.
- **Post_Id.** The identifier for the retrieved answer.
- **Rank.** Is the rank of retrieved answer. This is an integer number between 1 and 1000.
- **Score.** A float number value showing the relevance score of an answer to the question query.
- **Run_Number.** An identifier for the run.

Here is an example of a result file (the Ids are just for explanation):

```
A.1      19105    1      1      Run_0
A.1      5042     2     0.98   Run_0
A.1      77842    3     0.87   Run_0
...
```

which is showing the first 3 retrieved answers from a system for the first query in Task 1. The top-ranked answer for query 1, has the post id of ‘19105’ with relevance score of 1.

For each question query in Task 1, the participants will retrieve up to 1000 related answers. Then pooling is applied to the top-k results from all the participants as shown in Figure 1. Note that the posts retrieved across results files are merged; i.e., each posting appears at most once in the pool.



Top-k answers selected from each participant results for each query.

Figure 1. Creating a pool for a given query from results of 4 participants.

Annotation

To annotate each answer, we will use Turkle¹. The overview of the annotation tool is shown in Figure 2. For each question, all the retrieved answers in the pool will be shown to the annotator, each in a separate task. The annotators will look at each answer and decide if the retrieved answer is related. There will be 4 levels of relevance: high, medium, low and non-relevant. If the annotator cannot decide the relevance, they will choose “Do not know” and if by any chance the annotation system fails, they will click on “System failure”. To decide the relevance, the annotator will look at the question and the retrieved answer and decide to what extent the given answer is related to the question query. For instance, an annotator can give a high relevance score if the retrieved answer is clearly and completely addressing the question, a medium score for an answer that is helpful but not complete, a low score for an answer which is helpful but not well-explained.

Note that for each of the answers, annotators can click on the Thread link above the answer to view the thread in which the answer is located to read more. This can help the annotator to better understand the context and if they cannot decide the relevance of the answers, by looking at the complete thread including the question and other related answers.

As Figure 2 shows, for each annotation task, an annotator can see the question (which is the query in Task 1) on the left side and then look at a retrieved answer from the pool in the middle. If by reading the answer the annotator cannot decide the relevance, they can click on the blue link above the answer (Thread) and go to the main thread where the answer being judged is located. This will open a new tab in the browser for the annotator. Then the annotator can use one of the buttons on the right to decide the relevance score and also add additional comment (not required), below the buttons. After the annotation is done, the annotator will click on the submit button.

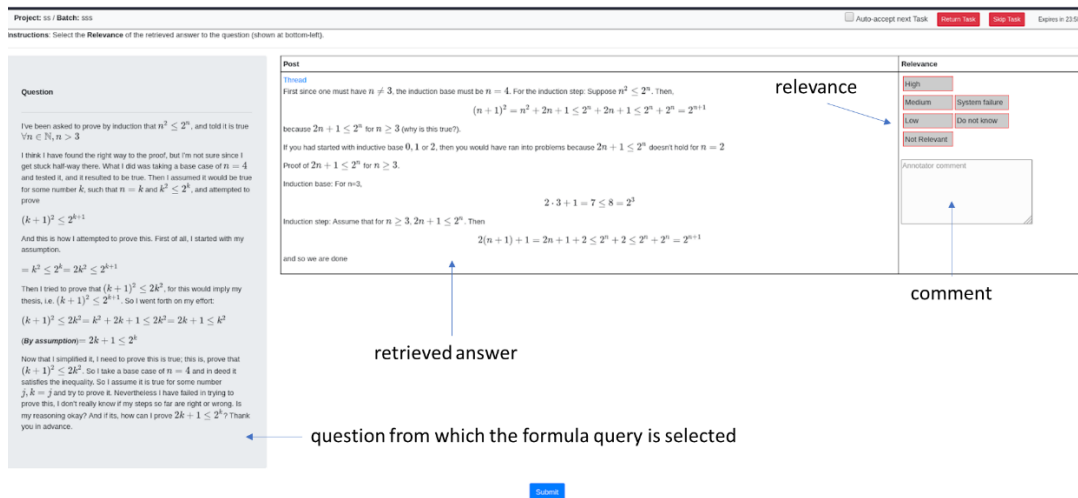


Figure 2. Annotation tool for Task 1.

Important Note: No data from the external links will be available to the annotators; if there is a link to another post inside the ARQMath dataset, the annotators can click on that and see the post. For example, if an answer is linked to “https://en.wikipedia.org/wiki/Pythagorean_theorem” which

¹ <https://github.com/hltcoe/turkle>

is a Wikipedia page, the annotator will not look at the page. But if an answer is linked to a page like “<https://math.stackexchange.com/questions/163309/pythagorean-theorem>”, they can open the link and check the thread. Therefore, if the href attribute of ‘a’ html tag is linked to any pages outside the “math.stachexchange.com” they will be ignored.

Evaluation Task 2: Formula Retrieval

Pooling

The retrieval results submitted by the participants should be in a TSV file, with the following columns:

Query_Id	Formula_Id	Post_Id	Rank	Score	Run_Number
----------	------------	---------	------	-------	------------

The different between the result files for Task 1 and 2, is that participant should provide the Formula_Id along with the Post_Id. The Formula_Id is the id of the retrieved formula and the Post_Id is the id of the answer in which the retrieved formula appears. Here is an example of a result file (the Ids are just for explanation):

B.1	241	232342	1	1	Run_0
B.1	1005	8502	2	0.98	Run_0
B.1	251	232342	3	0.87	Run_0
...					

which is showing the first 3 retrieved formulas from a system for the first query in Task 2. The first relevant formula for query 1, has id of 241, and is retrieved from an answer with id ‘232342’.

For each formula query in Task 2, the participants will retrieve up to k related formulas from the answers. Then pooling is applied to the top-k results from all the participants as shown in Figure 1 in a similar way as in Task 1, with this difference that individual formulas will be pooled.

The next step after the pooling is relevance annotation. To understand how this is done, consider the query ‘ $F(x) = x$ ’ for which formula such as ‘ $F(x) = -x$ ’ is retrieved by different participants. This formula ‘ $F(x) = -x$ ’ can be retrieved by a participant from a thread (question and answers) with id equal to ‘1’ where another participant can retrieve the same formula from a different thread with id equal to ‘99’. Also note that as a thread has multiple answers, and a given formula can be repeated in different answers from the same thread.

Looking at our example, if the formula ‘ $F(x) = -x$ ’ is in the pool, we first create a dictionary for which the keys are the thread ids in the pool that contains this formula (each formula is located in an answer and each answer is located in a thread). Looking at Figure 3, the threads 1, 2, 3 and 4 have the formula ‘ $F(x) = -x$ ’. The values for this dictionary are the list of answer ids from threads that have this formula (and are in the pool). For example, thread 1, has this formula in different answers with ids A, C, and G.

After creating this map, to decide if the retrieved formula is related to the query, we show a **maximum** of N different answers that are in the pool and contain that formula. Choosing these N answers will follow these policies:

- 1- If the number of answers (that are in the pool and contain the formula being judged) is less or equal than N , they will be all judged by the annotators.

- 2- If the number of answers is more than \underline{N} but the number of threads is less than \underline{N} , first from each thread one answer will be randomly chosen to be judged. The rest of the answers are selected randomly from the unselected posts.
- 3- If the numbers of answers and threads are more than \underline{N} , then \underline{N} unique threads are chosen randomly and for each thread a random answer is chosen.

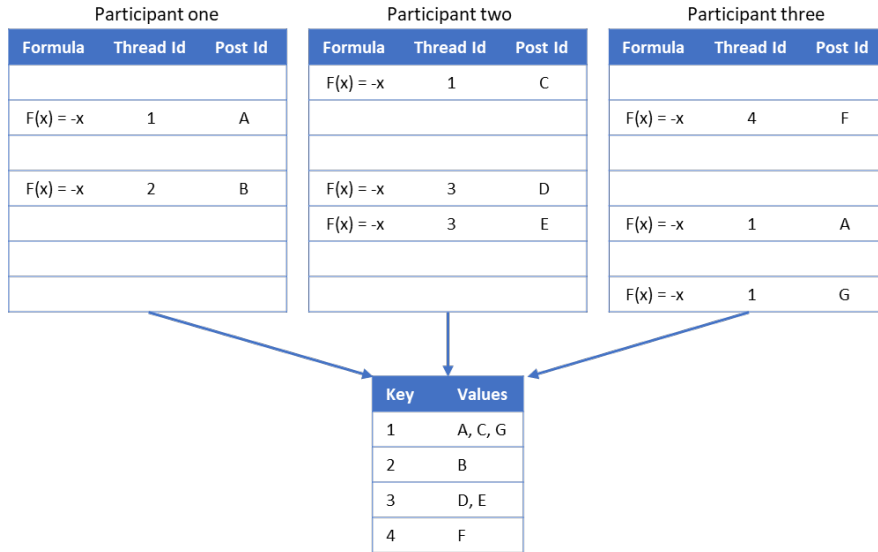


Figure 3. Example of creating a map for annotation for a given formula in the pool. Note that the participants will only submit their result files and this figure is just for illustration.

Annotation

To annotate each formula, we will use Turkle as we do for Task 1. The overview of the annotation tool is shown in Figure 5. For each query, all the retrieved formulas in the pool will be shown to the annotator, each in a separate task. The formula query will be highlighted and shown to annotators along with the question that the formula is selected from. The question will be used as a narrative, which gives a concise description of what makes a formula relevant. The annotators will look at each formula (highlighted), along with the answer from which they were retrieved and decide if the retrieved formula is related using the same 4 relevance levels as in Task 1. To decide the relevance score, the annotator will consider the formula query along with the question from which it was chosen (as the context) and decide the relevance of the retrieved formula in the context of the answer in which the formula appears.

As can be seen in Figure 5, the annotator will look at the retrieved formula on the top left of the tool, and will judge its relevance to the formula highlighted in the question below it (bottom left) by looking at the answer(s) in which the retrieved formula is located.

Note that for each of the answers, users can click on the thread to view the thread in which the answer is located to read more. This can help the annotator to better understand the context and if they cannot decide the relevance of the retrieved formula by looking at the answer in which it is located, they can view the whole thread. For example, to judge the formula in Figure 4, the annotators may want to see the whole thread. Also note that when clicking on the thread, the

annotators can see other information such as comments, scores and user information. After annotators have scored all the answers from the pool in which the retrieved formula has appeared, the final relevance score of formula will be the maximum relevance score for each unique formula.


▲ Completing the previous answers, it's enough to look for matrices $n \times n$ that satisfy

1 $(A^2) = (A^2)^{-1}$

▼ where $A^2 \neq I$. Check this in the previous answers.

share cite improve this answer

answered Feb 25 '18 at 18:36

 Corrêa

3,632 ● 1 ■ 5 ▲ 23

Figure 4. Example of a retrieved formula that need the whole thread to be judged.

Important Note: Same as task 1, no data from the external links will be available to the annotators; if there is a link to another post inside the ARQMath dataset, the annotators are able to click on that and see the post.

The screenshot shows the annotation tool interface for Task 2. It features a central 'retrieved formula' section with the formula $2n + 1 \leq 2^n$. To the left is a 'question from which the formula query is selected' containing a detailed mathematical proof attempt. To the right is a 'comment' section with a 'relevance' score field. Annotations with blue arrows point to specific elements: 'thread link' points to a link in the question, 'retrieved answer(s)' points to the formula, and 'relevance' points to the score field. A 'question from which the formula query is selected' label points to the question text. The interface also includes a 'share' button at the bottom.

Figure 5. Annotation tool for Task 2.

Effectiveness Measure

For both tasks 1 and 2, the nDCG-prime measure, introduced by Sakai and Kando ², will be used. This measure is calculated in a similar way to nDCG, with removing the unjudged documents (formulas) as the TREC Evaluation tool ³ considers those as non-relevant. The motivation for this measure is to support fair comparison later on for systems without results submitted to ARQMath.

² Sakai, T., & Kando, N. (2008). On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5), 447-470.

³ https://github.com/usnistgov/trec_eval