Overview of ARQMath 2020 (Updated Working Notes Version): CLEF Lab on Answer Retrieval for Questions on Math

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

[1] Rochester Institute of Technology (USA)
{rxzvcs,axasma,bm3302}@rit.edu
[2] University of Maryland, College Park (USA)
oard@umd.edu

Abstract. The ARQMath Lab at CLEF considers finding answers to new mathematical questions among posted answers on a community question answering site (Math Stack Exchange). Queries are question posts held out from the searched collection, each containing both text and at least one formula. This is a challenging task, as both math and text may be needed to find relevant answer posts. ARQMath also includes a formula retrieval sub-task: individual formulas from question posts are used to locate formulae in earlier question and answer posts, with relevance determined considering the context of the post from which a query formula is taken, and the posts in which retrieved formulae appear.

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

1 Introduction

In a recent study, Mansouri et al. found that 20% of mathematical queries in a general-purpose search engine were expressed as well-formed questions, a rate ten times higher than that for all queries submitted [14]. Results such as these and the presence of Community Question Answering (CQA) sites such as Math Stack Exchange[3] suggest there is interest in finding answers to mathematical questions posed in natural language, using both text and mathematical notation. Related to this, there has also been increasing work on math-aware information retrieval and math question answering in both the Information Retrieval (IR) and Natural Language Processing (NLP) communities.

[3] https://math.stackexchange.com

Table 1. Examples of relevant and not-relevant results for tasks 1 and 2 [12]. For Task 2, formulas are associated with posts, indicated with ellipses at right (see Figure 1 for more details). Query formulae are from question posts (here, the question at left), and retrieved formulae are from either an answer or a question post.

Task 1: Question Answering	Task 2: Formula Retrieval
Question I have spent the better part of this day trying to show from first principles that this sequence tends to 1. Could anyone give me an idea of how I can approach this problem? $\lim_{n\to+\infty} n^{\frac{1}{n}}$	Query Formula $\ldots \quad \lim_{n\to+\infty} n^{\frac{1}{n}} \quad \ldots$
Relevant You can use AM $\geq$ GM. $\frac{1+1+\cdots+1+\sqrt{n}+\sqrt{n}}{n} \geq n^{1/n} \geq 1$ $1-\frac{2}{n}+\frac{2}{\sqrt{n}} \geq n^{1/n} \geq 1$	Relevant $\ldots \quad \lim_{n\to\infty} \sqrt[n]{n} \quad \ldots$
Not Relevant If you just want to show it converges, then the partial sums are increasing but the whole series is bounded above by $1+\int_1^{\infty} \frac{1}{x^2}dx = 2$	Not Relevant $\ldots \quad \sum_{k=1} \frac{1}{k^2} = \frac{\pi^2}{6} \quad \ldots$

In light of this growing interest, we organized this new lab at the Conference and Labs of the Evaluation Forum (CLEF) on Answer Retrieval for Questions about Math (ARQMath).[4] Using the formulae and text in posts from Math Stack Exchange, participating systems are given a question, and asked to return a ranked list of potential answers. Relevance is determined by how well each returned post answers the provided question. Through this task we explore leveraging math notation together with text to improve the quality of retrieval results. This is one case of what we generically call math retrieval, in which the focus is on leveraging the ability to process mathematical notation to enhance, rather than to replace, other information retrieval techniques. We also included a formula retrieval task, in which relevance is determined by how useful a retrieved formula is for the searcher's intended purpose, as best could be determined from the query formula's associated question post. Table 1 illustrates these two tasks, and Figure 1 shows the topic format for each task.

For the CQA task, 70,342 questions from 2019 that contained some text and at least one formula were considered as search topics, from which 77 were selected as test topics. Participants had the option to run queries using only the text or math portions of each question, or to use both math and text. One challenge inherent in this design is that the expressive power of text and formulae are sometimes complementary; so although all topics will include both text and formula(s), some may be better suited to text-based or math-based retrieval.

[4] `https://www.cs.rit.edu/~dprl/ARQMath`

Task 1: Question Answering

```
<Topics>
  ...
  <Topic number="A.9">
    <Title>Simplifying this series</Title>
    <Question>
      I need to write the series
      <span class=``math-container'' id=``q_52''>
        $$\sum_{n=0}^N nx^n$$
      </span>
      in a form that does not involve the summation
      notation, for example
      <span class=``math-container'' id=``q_53''>
        $\sum_{i=0}^n i^2 = \frac{(n^2+n)(2n+1)}{6}$
      </span>
      Does anyone have any idea how to do this?
      I have attempted multiple ways including using
      generating functions however no luck.
    </Question>
    <Tags>sequences-and-series</Tags>
  </Topic>
  ...
</Topics>
```

Task 2: Formula Retrieval

```
<Topics>
  ...
  <Topic number="B.9">
    <Formula_Id>q_52</Formula_Id>
    <Latex>\sum_{n=0}^N nx^n</Latex>
    <Title>Simplifying this series</Title>
    <Question>
      ...
    </Question>
    <Tags>sequences-and-series</Tags>
  </Topic>
  ...
</Topics>
```

Fig. 1. XML Topic File Formats for Tasks 1 and 2. Formula queries in Task 2 are taken from questions in Task 1. Here, formula topic B.9 is a copy of question topic A.9 with two additional tags for the query formula identifier and LaTeX before the question post.

For the formula search task, an individual formula is used as the query, and systems return a ranked list of other potentially useful instances of formulae found in the collection. Each of the 45 queries is a single formula extracted from a question used in the CQA task.

Mathematical problem solving was amongst the earliest applications of Artificial Intelligence, such as Newell and Simon's work on automatic theorem proving [15]. More recent work in math problem solving includes systems that solve algebraic word problems while providing a description of the solution method [11], and that solve algebra word problems expressed in text and math [10]. The focus of ARQMath is different; rather than prove or solve concrete mathematical problems, we instead look to find answers to informal, and potentially open-ended and incomplete questions posted naturally in a CQA setting.

The ARQMath lab provides an opportunity to push mathematical question answering in a new direction, where answers provided by a community are se-

lected and ranked rather than generated. We aim to produce test collections, drive innovation in evaluation methods, and drive innovation in the development of math-aware information retrieval systems. An additional goal is welcoming new researchers to work together on these challenging problems.

2 Related Work

The Mathematical Knowledge Management (MKM) research community is concerned with the representation, application, and search of mathematical information. Among other accomplishments, their activities informed the development of MathML[5] for math on the Web, and novel techniques for math representation, search, and applications such as theorem proving. This community continues to meet annually at the CICM conferences [8].

Math-aware search (sometimes called *Mathematical Information Retrieval*) has seen growing interest over the past decade. Math formula search has been studied since the mid-1990's for use in solving integrals, and publicly available math+text search engines have been around since the DLMF[6] system in the early 2000's [6, 21]. The most widely used evaluation resources for math-aware information retrieval were initially developed over a five-year period at the National Institute of Informatics (NII) Testbeds and Community for Information access Research (at NTCIR-10 [1], NTCIR-11 [2] and NTCIR-12 [20]). NTCIR-12 used two collections, one a set of arXiv papers from physics that is split into paragraph-sized documents, and the other a set of articles from English Wikipedia. The NTCIR Mathematical Information Retrieval (MathIR) tasks developed evaluation methods and allowed participating teams to establish baselines for both "text + math" queries (i.e., keywords and formulas) and isolated formula queries.

A recent math question answering task was held for SemEval 2019 [7]. Question sets from MathSAT (Scholastic Achievement Test) practice exams in three categories were used: Closed Algebra, Open Algebra and Geometry. A majority of the questions were multiple choice, with some having numeric answers. This is a valuable parallel development; the questions considered in the CQA task of ARQMath are more informal and open-ended, and selected from actual MSE user posts (a larger and less constrained set).

At NTCIR-11 and NTCIR-12, formula retrieval was considered in a variety of settings, including the use of wildcards and constraints on symbols or subexpressions (e.g., requiring matched argument symbols to be variables or constants). Our Task 2, Formula Retrieval, has similarities in design to the NTCIR-12 Wikipedia Formula Browsing task, but differs in how queries are defined and how evaluation is performed. In particular, for evaluation ARQMath uses the *visually distinct* formulas in a run, rather than all (possibly identical) formula instances, as had been done in NTCIR-12. The NTCIR-12 formula retrieval test collection also had a smaller number of queries, with 20 fully specified formula

[5] `https://www.w3.org/Math`
[6] `https://dlmf.nist.gov`

queries (plus 20 variants of those same queries with subexpressions replaced by wildcard characters). NTCIR-11 also had a formula retrieval task, with 100 queries, but in that case systems searched only for exact matches [19].

Over the years, the size of the NTCIR-12 formula browsing task topic set has limited the diversity of examples that can be studied, and made it difficult to measure statistically significant differences in formula retrieval effectiveness. To support research that is specifically focused on formula similarity measures, we have create a formula search test collection that is considerably larger, and in which the definition of relevance derives from the specific task for which retrieval is being performed, rather than isolated formula queries.

3 The ARQMath 2020 Math Stack Exchange Collection

In this section we describe the raw data from which we started, collection processing, and the resulting test that was used in both tasks. Topic development for each task is described in the two subsequent sections.

3.1 MSE Internet Archive Snapshot

We chose Math Stack Exchange (MSE), a popular community question answering site as the collection to be searched. The Internet Archive provides free public access to MSE snapshots.[7] We processed the 01-March-2020 snapshot, which in its original form contained the following in separate XML files:

- **Posts**: Each MSE post has a unique identifier, and can be a question or an answer, identified by '`post type id`' of 1 and 2 respectively. Each question has a title and a body (content of the question) while answers only have a body. Each answer has a '`parent id`' that associates it with the question it is an answer is for. There is other information available for each post, including its score, the post owner id and creation date.
- **Comments**: MSE users can comment on posts. Each comment has a unique identifier and a '`post id`' indicating which post the comment is written for.
- **Post links**: Moderators sometimes identify duplicate or related questions that have been previously asked. A '`post link type id`' of value 1 indicates related posts, while value 3 indicates duplicates.
- **Tags**: Questions can have one or more tags describing the subject matter of the question.
- **Votes**: While the post score shows the difference between up and down votes, there are other vote types such as '`offensive`' or '`spam.`' Each vote has a '`vote type id`' for the vote type and a '`post id`' for the associated post.
- **Users**: Registered MSE users have a unique id, and they can provide additional information such as their website. Each user has a reputation score, which may be increased through activities such as posting a high quality answer, or posting a question that receives up votes.

[7] `https://archive.org/download/stackexchange`

– **Badges**: Registered MSE users can also receive three badge types: bronze, silver and gold. The '`class`' attribute shows the type of the badge, value 3 indicating bronze, 2 silver and 1 gold.

The edit history for posts and comments is also available, but for this edition of the ARQMath lab, edit history information has not been used.

3.2 The ARQMath 2020 Test Collection

Because search topics are built from questions asked in 2019, all training and retrieval is performed on content from 2018 and earlier. We removed any data from the collection generated after the year 2018, using the '`creation date`' available for each item. The final collection contains roughly 1 million questions and 28 million formulae.

Formulae. While MSE provides a `<math-container>` HTML tag for some mathematical formulae, many are only present as a LaTeX string located between single or double '$' signs. Using the math-container tags and dollar sign delimiters we identified formulae in question posts, answer posts, and comments. Every identified instance of a formula was assigned a unique identifier, and then placed in a `<math-container>` HTML tag using the form:

<**span id**=FID **class**="math-container">... </**span**>

where FID is the formula id. Overall, 28,320,920 formulae were detected and annotated in this way.

Additional Formula Representations. Rather than use raw LaTeX, it is common for math-aware information retrieval systems to represent formulas as one or both of two types of rooted trees. Appearance is represented by the spatial arrangement of symbols on writing lines (in Symbol Layout Trees (SLTs)), and mathematical syntax (sometimes referred to as (shallow) semantics) is represented using a hierarchy of operators and arguments (in Operator Trees (OPTs)) [5, 13, 23]. The standard representations for these are Presentation MathML (SLT) and Content MathML (OPT). To simplify the processing required of participants, and to maximize comparability across submitted runs, we used LaTeXML[8] to generate Presentation MathML and Content MathML from LaTeX for each formula in the ARQMath collection. Some LaTeX formulas were malformed and LaTeXML has some processing limitations, resulting in conversion failures for 8% of SLTs, and and 10% of OPTs. Participants could elect to do their own formula extraction and conversions, although the formulae that could be submitted in system runs for Task 2 were limited to those with identifiers in the LaTeX TSV file.

ARQMath formulae are provided in LaTeX, SLT, and OPT representations, as Tab Separated Value (TSV) index files. Each line of a TSV file represents a single instance of a formula, containing the formula id, the id of the post in which the formula instance appeared, the id of the thread in which the post

[8] `https://dlmf.nist.gov/LaTeXML`

is located, a post type (title, question, answer or comment), and the formula representation in either LaTeX, SLT (Presentation MathML), or OPT (Content MathML). There are two sets of formula index files: one set is for the collection (i.e., the posts from 2018 and before), and the second set is for the search topics (see below), which are from 2019.

HTML Question Threads. HTML views of threads, similar to those on the MSE web site (a question, along with answers and other related information) are also included in the ARQMath test collection. The threads are constructed automatically from the MSE snapshot XML files described above. The threads are intended for use by teams who performed manual runs, or who wished to examine search results (on queries other than evaluation queries) for formative evaluation purposes. These threads were also used by assessors during evaluation. The HTML thread files were intended only for viewing threads; participants were asked to use the provided XML and formula index files (described above) to train their models.

Distribution. The MSE test collection was distributed to participants as XML files on Google drive.[9] To facilitate local processing, the organizers provided python code on GitHub[10] for reading and iterating over the XML data, and generating the HTML question threads.

4 Task 1: Answer Retrieval

The primary task for ARQMath 2020 was the answer retrieval task, in which participants were presented with a question that had actually been asked on MSE in 2019, and were asked to return a ranked list of up to 1,000 answers from prior years (2010-2018). System results ('runs') were evaluated using rank quality measures (e.g., nDCG′), so this is a ranking task rather than a set retrieval task, and participating teams were not asked to say where the searcher should stop reading. This section describes for Task 1 the search topics (i.e., the questions), the submissions and baseline systems, the process used for creating relevance judgments, the evaluation measures, and the results.

4.1 Topics

In Task 1 participants were given 101 questions as search topics, of which 3 were training examples. These questions are selected from questions asked on MSE in 2019. Because we wished to support experimentation with retrieval systems that use text, math, or both, we chose from only the 2019 questions that contain some text and at least one formula. Because ranking quality measures can distinguish between systems only on topics for which relevant documents exist, we calculated the number of duplicate and related posts for each question and chose only from those that had at least one duplicate or related post.[11] Because we were

[9] `https://drive.google.com/drive/folders/1ZPKIWDnhMGRaPNVLi1reQxZWTfH2R4u3`
[10] `https://github.com/ARQMath/ARQMathCode`
[11] Note that participating systems did not have access to this information.

interested in a diverse range of search tasks, we also calculated the number of formulae and Flesch's Reading Ease score [9] for each question. Finally, we noted the asker's reputation and the tags assigned for each question. We then manually drew a sample of 101 questions that was stratified along those dimensions. In the end, 77 of these questions were evaluated and included in the test collection.

The topics were selected from various domains (real analysis, calculus, linear algebra, discrete mathematics, set theory, number theory, etc.) that represent a broad spectrum of areas in mathematics that might be of interest to expert or non-expert users. The difficulty level of the topics spanned from easy problems that a beginning undergraduate student might be interested in to difficult problems that would be of interest to more advanced users. The bulk of the topics were aimed at the level of undergraduate math majors (in their 3rd or 4th year) or engineering majors fulfilling their math requirements.

Some topics had simple formulae; others had fairly complicated formulae with subscripts, superscripts, and special symbols like the double integral $\iint_V f(x, y)dx\,dy$ or binomial coefficients such as $\binom{n}{r}$. Some topics were primarily based on computational steps, and some asked about proof techniques (making extensive use of text). Some topics had named theorems or concepts (e.g. Cesàro-Stolz theorem, Axiom of choice).

As organizers, we labeled each question with one of three broad categories, *computation*, *concept* or *proof*. Out the 77 assessed questions, 26 were categorized as *computation*, 10 as *concept*, and 41 as *proof*. We also categorized the questions based on their perceived difficulty level, with 32 categorized as easy, 21 as medium, and 24 as hard.

The topics were published as an XML file with the format shown in Figure 1, where the topic number is an attribute of the Topic tag, and the Title, Question and asker-provided Tags are from the MSE question post. To facilitate system development, we provided python code that participants could use to load the topics. As in the collection, the formulae in the topic file are placed in '`math-container`' tags, with each formula instance being represented by a unique identifier and its LaTeX representation. And, as with the collection, we provided three TSV files, one each for the LaTeX, OPT and SLT representations of the formulae, in the same format as the collection's TSV files.

4.2 Runs Submitted by Participating Teams

Participating teams submitted runs using Google Drive. A total of 18 runs were were received from a total of 5 teams. Of these, 17 runs were declared as automatic, meaning that queries were automatically processed from the topic file, that no changes to the system had been made after seeing the queries, and that ranked lists for each query were produced with no human intervention. One run was declared as manual, meaning that there was some type of human involvement in generating the ranked list for each query. Manual runs can contribute diversity to the pool of documents that are judged for relevance, since their error characteristics typically differ from those of automatic runs. All submitted runs used both text and formulae. The teams and submissions are shown in Table

Table 2. Submitted Runs for Task 1 (18 runs) and Task 2 (11 runs). Additional baselines for Task 1 (5 runs) and Task 2 (1 run) were also generated by the organizers.

	Automatic Runs		**Manual Runs**	
	Primary	Alternate	Primary	Alternate
Task 1: Question Answering				
Baselines	4			1
DPRL	1	3		
MathDowsers	1	3		1
MIRMU	3	2		
PSU	1	2		
ZBMath			1	
Task 2: Formula Retrieval				
Baseline	1			
DPRL	1	3		
MIRMU	2	3		
NLP-NIST	1			
ZBMath			1	

2. Please see the participant papers in the working notes for descriptions of the systems that generated these runs.

Of the 17 runs declared as automatic, two were in fact manual runs (for ZBMath, see Table 2).

4.3 Baseline Runs

As organizers, we ran five baseline systems for Task 1. The first baseline is a TF-IDF (term frequency–inverse document frequency) model using the Terrier system [17]. In the TF-IDF baseline, formulae are represented using their LaTeX string. The second baseline is Tangent-S, a formula search engine using SLT and OPT formula representations [5]. One formula was selected from each Task 1 question title if possible; if there was no formula in the title, then one formula was instead chosen from the question's body. If there were multiple formulae in the selected field, the formula with the largest number of nodes in its SLT representation was chosen. Finally, if there were multiple formulae with the highest number of nodes, one of these was chosen randomly. The third baseline is a linear combination of TF-IDF and Tangent-S results. To create this combination, first the relevance scores from both systems were normalized between 0 and 1 using min-max normalization, and then the two normalized scores were combined using an unweighted average.

The TF-IDF baseline used default parameters in Terrier. The second baseline (Tangent-S) retrieves formulae independently for each representation, and then linearly combines SLT and OPT scoring vectors for retrieved formulae [5]. For ARQMath, we used the average weight vector from cross validation results obtained on the NTCIR-12 formula retrieval task.

The fourth baseline was the ECIR 2020 version of the Approach0 text + math search engine [22], using queries manually created by the third and fourth authors. This baseline was not available in time to contribute to the judgment pools and thus was scored post hoc.

Table 3. Retrieval Time in Seconds for Task 1 Baseline Systems.

System	Run Time (seconds) Min (Topic)	Max (Topic)	(Avg, StDev)
TF-IDF (Terrier)	0.316 (A.42)	1.278 (A.1)	(0.733, 0.168)
Tangent-S	0.152 (A.72)	160.436 (A.60)	(6.002, 18.496)
TF-IDF + Tangent-S	0.795 (A.72)	161.166 (A.60)	(6.740, 18.483)
Approach0	0.007 (A.3)	91.719 (A.5)	(17.743, 18.789)

The final baseline was built from duplicate post links from 2019 in the MSE collection (which were not available to participants). This baseline returns *all* answer posts from 2018 or earlier that were in threads from 2019 or earlier that MSE moderators had marked as duplicating the question post in a topic. The posts are sorted in descending order by their vote scores.

Performance. Table 3 shows the minimum, maximum, average, and standard deviation of retrieval times for each of the baseline systems. For running all the baselines, we used a system of 528 GB Ram, with Intel(R) Xeon(R) CPU E5-2667 v4 @ 3.20GHz.

4.4 Assessment

Pooling. Participants were asked to rank 1,000 (or fewer) answer posts for each Task 1 topic. Top-k pooling was then performed to create pools of answer posts to be judged for relevance to each topic. The top 50 results were combined from all 7 primary runs, 4 baselines, and 1 manual run. To this, we added the top 20 results from each of the 10 automatic alternate runs. Duplicates were then deleted, and the resulting pool was sorted in random order for display to assessors. The pooling process is illustrated in Figure 2. This process was designed to identify as many relevant answer posts as possible given the available assessment resources. On average, pools contained about 500 answers per topic.

Relevance definition. Some questions might offer clues as to the level of mathematical knowledge on the part of the person posing the question; others might not. To avoid the need for the assessor to guess about the level of mathematical knowledge available to the person interpreting the answer, we asked assessors to base their judgments on degree of usefulness for an expert (modeled in this case as a math professor) who might then try to use that answer to help the person who had asked the original question. We defined four levels of relevance, as shown in Table 4.

Assessors were allowed to consult external sources on their own in order to familiarize themselves with the topic of a question, but the relevance judgments for each answer post were performed using only information available within the collection. For example, if an answer contained an MSE link such as `https://math.stackexchange.com/questions/163309/pythagorean-theorem`, they could follow that link to better understand the intent of the person writing the

Table 4. Relevance Scores, Ratings, and Definitions for Tasks 1 and 2.

Task 1: Question Answering		
Score	Rating	Definition
3	High	Sufficient to answer the complete question on its own
2	Medium	Provides some path towards the solution. This path might come from clarifying the question, or identifying steps towards a solution
1	Low	Provides information that could be useful for finding or interpreting an answer, or interpreting the question
0	Not Relevant	Provides no information pertinent to the question or its answers. A post that restates the question without providing any new information is considered non-relevant
Task 2: Formula Retrieval		
Score	Rating	Definition
3	High	Just as good as finding an exact match to the query formula would be
2	Medium	Useful but not as good as the original formula would be
1	Low	There is some chance of finding something useful
0	Not Relevant	Not expected to be useful

answer, but an external link to the Wikipedia page `https://en.wikipedia.org/wiki/Pythagorean_theorem` would not be followed.

Training Set. The fourth author created a small set of relevance judgment files for three topics. We used duplicate question links to find possibly relevant answers, and then performed relevance judgments on the same 0, 1, 2 and 3 scale that was later used by the assessors. We referred to this as a 'training set,' although in practice such a small collection is at best a sanity check to see if

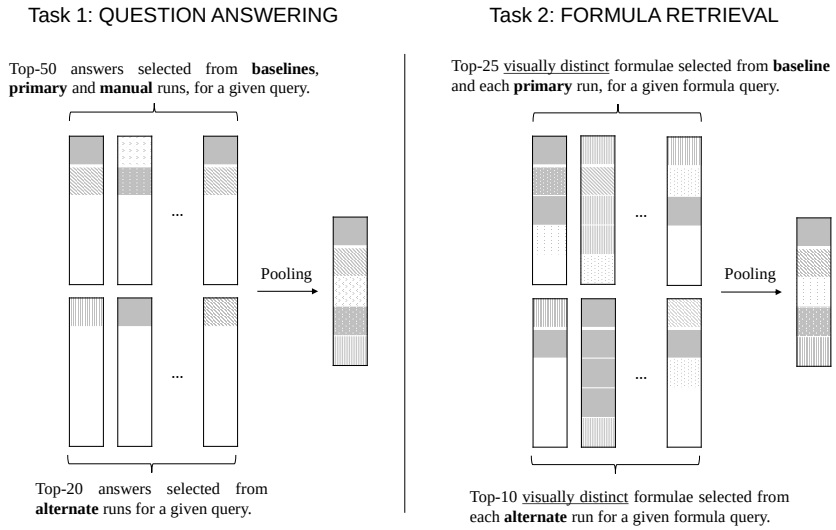

Fig. 2. Pooling Procedures. For Task 1, the pool depth for baselines, primary, and manual runs is 50, and for alternate runs 20. For Task 2 pool depth is the rank at which k visually distinct formulae are observed (25 for primary/baseline, 10 for alternate).

systems were producing reasonable results. Moreover, these relevance judgments were performed before assessor training had been conducted, and thus the definition of relevance used by the fourth author may have differed in subtle ways from the definitions on which the assessors later settled.

Assessment System. Assessments were performed using Turkle[12], a locally installed system with functionality similar to Amazon Mechanical Turk. Turkle uses an HTML task template file, plus a Comma Separate Value (CSV) file to fill HTML templates for each topic. Each row in the CSV file contains the question title, body, and the retrieved answer to be judged. Judgments are exported as CSV files.

As Figure 6 (at the end of this document) illustrates, there were two panels in the Turkle user interface. The question was shown on the left panel, with the Title on top (in a grey bar); below that was the question body. There was also a Thread link, on which assessors could click to look at the MSE post in context, with the question and all of the answers that were actually given for this question (in 2019). This could help the assessor to better understand the question. In the right panel, the answer to be judged was shown at the top. As with the question, there was a thread link where the assessors could click to see the original thread in which the answer post being judged had been present in MSE. This could be handy when the assessors wanted to see details such as the question that had been answered at the time. Finally, the bottom of the right panel (below the answer) was where assessors selected relevance ratings. In addition to four levels of relevance, two additional choices were available. 'System failure' indicated system issues such as unintelligible rendering of formulae, or the thread link not working (when it was essential for interpretation). If after viewing the threads, the assessors were still not able to decide the relevance degree, they were asked to choose 'Do not know'. The organizers asked the assessors to leave a comment in the event of a system failure or a 'Do no know' selection.

Assessor Training. Eight paid undergraduate mathematics students (or, in three cases, recent graduates with an undergraduate mathematics degree) were paid to perform relevance judgments. Four rounds of training were performed before submissions from participating teams had been received. In the first round, assessors met online using Zoom with the organizers, one of whom (the third author) is an expert MSE user and a Professor of mathematics. The task was explained, making reference to specific examples from the small training set. For each subsequent round, a small additional additional training set was created using a similar approach (pooling only answers to duplicate questions) with 8 actual Task 1 topics (for which the actual relevance judgments were not then known). The same 8 topics were assigned to every assessor and the assessors worked independently, thus permitting inter-annotator agreement measures to be computed. Each training round was followed by an online meeting with the organizers using Zoom at which assessors were shown cases in which one or more assessor pairs disagreed. They discussed the reasoning for their choices, with the third author offering reactions and their own assessment. These training judg-

[12] https://github.com/hltcoe/turkle

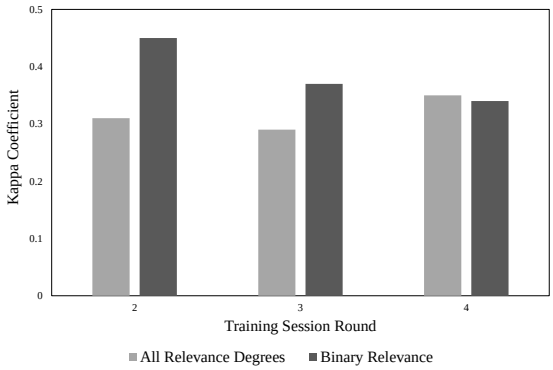

Fig. 3. Inter-annotator agreement (Fleiss' kappa) over 8 assessors during Task 1 training (8 topics per round); four-way classification (gray) and two-way (H+M binarized) classification (black).

ments were not used in the final collection, but the same topic could later be reassigned to one of the assessors to perform judgments on a full pool.

Some of the question topics would not be typically covered in regular undergraduate courses, so that was a challenge that required the assessors to get a basic understanding of those topics before they could do the assessment. The assessors found the questions threads made available in the Turkle interface helpful in this regard (see Figure 6).

Through this process the formal definition of each relevance level in Table 4 was sharpened, and we sought to help assessors internalize a repeatable way of making self-consistent judgments that were reasonable in the view of the organizers. Judging relevance is a task that calls for interpretation and formation of a personal opinion, so it was not our goal to achieve identical decisions. We did, however, compute Fleiss' Kappa for the three independently conducted rounds of training to check whether reasonable levels of agreement were being achieved. As Figure 3 shows, kappa of 0.34 was achieved by the end of training on the four-way assessment task. Collapsing relevance to be binary by considering high and medium as relevant and low and not-relevant as a not-relevant (henceforth "H+M binarization") yielded similar results.[13]

Assessment. A total of 80 questions were assessed for Task 1. Three judgment pools (for topics A2, A22, and A70) had zero or one posts with relevance ratings of high or medium; these 3 topics were removed from the collection be-

[13] H+M binarization corresponds to the definition of relevance usually used in the Text Retrieval Conference (TREC). The TREC definition is "If you were writing a report on the subject of the topic and would use the information contained in the document in the report, then the document is relevant. Only binary judgments ('relevant' or 'not relevant') are made, and a document is judged relevant if any piece of it is relevant (regardless of how small the piece is in relation to the rest of the document)." (source: `https://trec.nist.gov/data/reljudge_eng.html`)

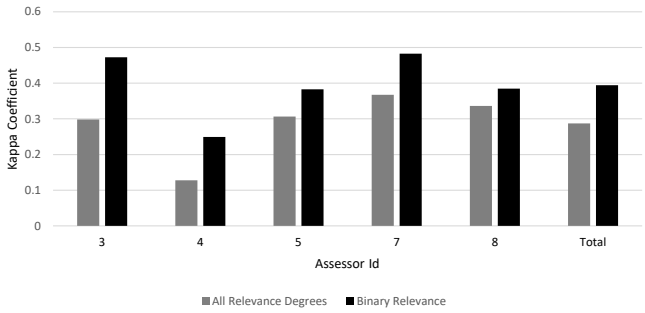

Fig. 4. Inter-annotator agreement (Cohen's kappa) over 5 assessors after Task 1 assessment was completed. Each assessor evaluated two topics that had been scored by two of the other assessors. Shown are results for four-way classification (gray) and two-way (H+M binarized) classification (black). Results are provided for each individual Task 1 assessor (as average kappa score), along with the average kappa values over all assessors at right ('Total').

cause topics with no relevant posts cannot be used to distinguish between ranked retrieval systems, and topics with only a single relevant post result in coarsely quantized values for H+M binarized evaluation measures, and that degree of quantization can adversely affect the ability to measure statistically significant differences. For the remaining 77 questions, an average of 508.5 answers were assessed for each question, with an average assessment time of 63.1 seconds per answer post. The average number of answers labeled with any degree of relevance (high, medium, or low; henceforth "H+M+L binarization") was 52.9 per question, with the highest number of relevant answers being 188 (for topic A.38) and the lowest being 2 (for topic A.96).

Post Assessment. After the official assessments were complete for Task 1, each assessor was assigned two tasks completed by two other assessors to calculate their agreement. As shown in Figure 4, across all five assessors ('Total') an average Cohen's kappa of 0.29 was achieved on the four-way assessment task, and using H+M binarization the average kappa value was 0.39. The individual assessors are reasonably similar (particularly in terms of 4-way agreement) except for Assessor 4. Comparing Figures 3 and 4, we see that agreement was relatively stable between the end of training and after assessments were completed. After assessment was complete, a slightly lower 4-way agreement but higher H+M binarized agreement was obtained relative to the end of training.

4.5 Evaluation Measures

One risk when performing a new task for which rich training data is not yet available is that a larger than typical number of relevant answers may be missed. Measures which treat unjudged documents as not relevant can be used when directly comparing systems that contributed to the judgment pools, but subsequent use of such a first-year test collection (e.g., to train new systems for the

second year of the lab) can be disadvantaged by treating unjudged documents (which as systems improve might actually be relevant) as not relevant. We therefore chose the nDCG′ measure (read as "nDCG-prime") introduced by Sakai and Kando [18] as the primary measure for the task.

The nDCG measure on which nDCG′ is based is a widely used measure when graded relevance judgments are available, as we have in ARQMath, that produced a single figure of merit over a set of ranked lists. Each retrieved document earns a gain value of (0, 1, 2, or 3) discounted by a slowly decaying function of the rank position of each document. The resulting discounted gain values are accumulated and then normalized to [0,1] by dividing by the maximum possible Discounted Cumulative Gain (i.e., from all identified relevant documents, sorted by decreasing order of gain value). This results in normalized Discounted Cumulative Gain (nDCG). The only difference when computing nDCG′ is that unjudged documents are removed from the ranked list before performing the computation. It has been shown that nDCG′ has somewhat better discriminative power and somewhat better system ranking stability (with judgement ablation) than the bpref measure [4] used recently for formula search (e.g., [13]). Moreover, nDCG′ yields a single-valued measure with graded relevance, whereas bpref, Precision@k, and Mean Average Precision (MAP) all require binarized relevance judgments. In addition to nDCG′, we also compute Mean Average Precision (MAP) with unjudged posts removed (thus MAP′), and Precision at 10 posts (P@10).[14] For MAP′ and P@10 we used H+M binarization. We removed unjudged posts as a preprocessing step where required, and then computed the evaluation measures using trec_eval.[15]

4.6 Results

Table A1 in the appendix shows the results, with baselines shown first, and then teams and their systems ranked by nDCG′. nDCG′ values can be interpreted as the average (over topics) of the fraction of the score for the best possible that was actually achieved. As can be seen, the best nDCG′ value that was achieved was 0.345, by the MathDowsers team. For measures computed using H+M binarization we can see that MAP′ and P@10 generally show system comparison patterns similar to those of nDCG′, although with some differences in detail.

5 Task 2: Formula Retrieval

In the formula retrieval task, participants were presented with one formula from a 2019 question used in Task 1, and asked to return a ranked list of up to 1,000 formula instances from questions or answers from the evaluation epoch (2018

[14] Pooling to at least depth 20 ensures that there are no unjudged posts above rank 10 for any primary or secondary submission, and for four of the five baselines. Note, however, that P@10 cannot achieve a value of 1 because some topics have fewer than 10 relevant posts.

[15] `https://github.com/usnistgov/trec_eval`

or earlier). Formulae were returned by their identifiers in `math-container` tags and the companion TSV LaTeX formula index file, along with their associated post identifiers.

This task is challenging because someone searching for math formulae may have goals not evident from the formula itself. For example:

- They may be looking to learn what is known, to form connections between disciplines, or to discover solutions that they can apply to a specific problem.
- They may want to find formulae of a specific form, including details such as specific symbols that have significance in a certain context, or they may wish to find related work in which similar ideas are expressed using different notation. For example, the Schrödinger equation is written both as a wave equation and as a probability current (the former is used in Physics, whereas the latter is used in the study of fluid flow).
- They may be happy to find formulae that contain only part of their formula query, or they may want only complete matches. For example, searching for $\sum_{i=1}^{n} u_i v_i$ could bring up the Cauchy-Schwarz inequality $\sum_{i=1}^{n} u_i v_i \leq \left(\sum_{i=1}^{n} u_i^2\right)^{\frac{1}{2}} \left(\sum_{i=1}^{n} v_i^2\right)^{\frac{1}{2}}$.

For these reasons (among others), it is difficult to formulate relevance judgments for retrieved formulae without access to the context in which the formula query was posed, and to the context in which each formula instance returned as a potentially useful search result was expressed.

Three key details differentiate Task 2 from Task 1. First, in Task 1 only answer posts were returned, but for Task 2 the formulae may appear in answer posts or in question posts. Second, for Task 2 we distinguish visually distinct formulae from instances of those formulae, and evaluate systems based on the ranking of the visually distinct formulae that they return. We call formulae appearing in posts *formula instances*, and of course the same formula may appear in more than one post. By *formula,* we mean a set of formula instances that are visually identical when viewed in isolation. For example, x^2 is a formula, $x * x$ is a different formula, and each time x^2 appears is a distinct instance of the formula x^2. Systems in Task 2 rank formula instances in order to support the relevance judgment process, but the evaluation measure for Task 2 is based on the ranking of visually distinct formulae. The third difference between Task 1 and Task 2 is that in Task 2 the goal is not answering questions, but rather, to show the searcher formulae that might be useful to them as they seek to satisfy their information need. Task 2 is thus still grounded in the question, but the relevance of a retrieved formula is defined by the formula's expected utility, not just the post in which that one formula instance was found.

As with Task 1, ranked lists were evaluated using rank quality measures, making this a ranking task rather than a set retrieval task. Unlike Task 1, the design of which was novel, a pre-existing training set for a similar task (the NTCIR-12 Wikipedia Formula Browsing task test collection [20]) was available to participants. However, we note that the definition of relevance used in Task 2 differs from the definition of relevance in the NTCIR-12 task. This section

describes for Task 2 the search topics, the submissions and baselines, the process used for creating relevance judgments, the evaluation measures, and the results.

5.1 Topics

In Task 2, participating teams were given 87 mathematical formulae, each found in a different question from Task 1 from 2019, and they were asked to find relevant formulae instances from either question or answer posts in the test collection (from 2018 and earlier). The topics for Task 2 were provided in an XML file similar to those of Task 1, in the format shown in Figure 1. Task 2 topics differ from their corresponding Task 1 topics in three ways:

1. **Topic number**: For Task 2, topic ids are in form "B.x" where x is the topic number. There is a correspondence between topic id in tasks 1 and 2. For instance, topic id "B.9" indicates the formula is selected from topic "A.9" in Task 1, and both topics include the same question post (see Figure 1).
2. **Formula_Id**: This added field specifies the unique identifier for the query formula instance. There may be other formulae in the Title or Body of the question post, but the query is only the formula instance specified by this Formula_Id.
3. **LaTeX**: This added field is the LaTeX representation of the query formula instance as found in the question post.

Because query formulae are drawn form Task 1 question posts, the same LaTeX, SLT and OPT TSV files that were provided for the Task 1 topics can be consulted when SLT or OPT representations for a query formula are needed.

Formulae for Task 2 were manually selected using a heuristic approach to stratified sampling over three criteria: complexity, elements, and text dependence. Formulae complexity was labeled low, medium or high by the fourth author. For example, $\frac{df}{dx} = f(x+1)$ is low complexity, $\sum_{k=0}^{n} \binom{n}{k} k$ is medium complexity, and $x - \frac{x^3}{3\times 3!} + \frac{x^5}{5\times 5!} - \frac{x^7}{7\times 7!} + \cdots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{(2n+1)}}{(2n+1)\times(2n+1)!}$ is high complexity. Mathematical elements such as limit, integral, fraction or matrix were manually noted by the fourth author when present. Text dependence reflected the fourth author's opinion of the degree to which text in the Title and Question fields were likely to yield related search results. For instance, for one Task 2 topic, the query formula is $\frac{df}{dx} = f(x+1)$ whereas the complete question is: "How to solve differential equations of the following form: $\frac{df}{dx} = f(x+1)$." When searching for this formula, perhaps the surrounding text could safely be ignored. At most one formula was selected from each Task 1 question topic to produce Task 2 topics. In 12 cases, it was decided that no formula in a question post would be a useful query for Task 2, and thus 12 Task 1 queries have no corresponding Task 2 query.

5.2 Runs Submitted by Participating Teams

A total of 11 runs were received for Task 2 from a total of 4 teams, as shown in Table 2. All were automatic runs. Each run contains at most 1,000 formula

instances for each query formula, ranked in decreasing order of system-estimated relevance to that query. For each formula instance in a ranked list, participating teams provided the formula_id and the associated post_id for that formula. Please see the participant papers in the working notes for descriptions of the systems that generated these runs.

5.3 Baseline Runs

We again used Tangent-S [5] as our baseline. Unlike Task 1, a single formula is specified for each Task 2 query, so no formula selection step was needed. This Tangent-S baseline makes no use of the question text.

Performance. For the Tangent-S baseline, the minimum retrieval time was 0.238 seconds for topic B.3, and the maximum retrieval time was 30.355 seconds for topic B.51. The average retrieval time for all queries was 3.757 seconds, with a standard deviation of 5.532 seconds. The same system configuration was used as in Task 1.

5.4 Assessment

Pooling. The retrieved items for Task 2 are formula instances, but pooling was done based on visually distinct formulae, not formula instances (see Figure 2). This was done by first clustering all formula instances from all submitted runs to identify visually distinct formulae, and then proceeding down each list until at least one instance of some number of different formulae had been seen. For primary runs and for the baseline run, the pool depth was the rank of the first instance of the 25th visually distinct formula; for secondary runs the pool depth was the rank of the first instance of the 10th visually distinct formulae. Additionally, a pool depth of 1,000 (i.e., all available formulae) was used for any formula for which the associated answer post had been marked as relevant for Task 1.[16] This was the only way in which the post ids for answer posts was used.

Clustering visually distinct formulae instances was performed using the SLT representation when possible, and the LaTeX representation otherwise. We first converted the Presentation MathML representation to a string representation using Tangent-S, which performed a depth-first traversal of the SLT, with each SLT node generating a single character of the SLT string. Formula instances with identical SLT strings were considered to be the same formula; note that this ignores differences in font. For formula instances with no Tangent-S SLT string available, we removed the white space from their LaTeX strings and grouped formula instances with identical strings. This process is simple and appears to have been reasonably robust for our purposes, but it is possible that some visually identical formula instances were not captured due to LaTeXML conversion failures, or where different LaTeX string produce the same formula (e.g., if subscripts and superscripts appear in a different order).

[16] One team submitted incorrect post id's for retrieved formulae; those post id's were not used for pooling.

Assessment was done on formula instances, so for each formula we selected at most five instances to assess. We selected the 5 instances that were contributed to the pools by the largest number of runs, breaking ties randomly. Out of 5,843 visually distinct formulae that were assessed, 93 (1.6%) had instances in more than 5 pooled posts.

Relevance definition. The relevance judgment task was defined for assessors as follows: for a formula query, if a search engine retrieved one or more instances of this retrieved formula, would that have been expected to be useful for the task that the searcher was attempting to accomplish?

Assessors were presented with formula instances, and asked to decide their relevance by considering whether retrieving either that instance or some other instance of that formula would have been useful, assigning each formula instance in the judgment pool one of four scores as defined in Table 4.

For example, if the formula query was $\sum \frac{1}{n^{2+\cos n}}$, and the formula instance to be judged is $\sum_{n-1}^{\infty} \frac{1}{n^2}$, the assessors would decide whether finding the second formula rather than the first would be expected to yield good results. To do this, they would consider the content of the question post containing the query (and, optionally, the thread containing that question post) in order to understand the searcher's actual information need. Thus the question post fills a role akin to Borlund's simulated work task [3], although in this case the title, body and tags from the question post are included in the topic and thus can optionally be used by the retrieval system. The assessor can also consult the post containing a retrieved formula instance (which may be another question post or an answer post), along with the associated thread, to see if in that case the formula instance would indeed have been a useful basis for a search. Note, however, that the assessment task is not to determine whether the specific post containing the retrieved formula instance is useful, but rather to use that context as a basis for estimating the degree to which useful content would likely be found if this or other instances of the retrieved formula were returned by a search engine.

We then defined the relevance score for a formula to be the maximum relevance score for any judged instance of that formula. This relevance definition essentially asks "if instances of this formula were returned, would we reasonably expect some of those instances to be useful?" This definition of relevance might be used by system developers in several ways. One possibility is using Task 2 relevance judgments to train a formula matching component for use in a Task 1 system. A second possibility is using these relevance judgments to train and evaluate a system for interactively suggesting alternative formulae to users.[17]

Assessment System. As in Task 1, we used Turkle to build the assessment system. As shown in Figure 6 (at the end of this document), there are two main panels. In the left panel, the question is shown as in Task 1, but now with the formula query highlighted in yellow. In the right panel, up to five retrieval posts (question posts or answer posts) containing instances of the same retrieved formula are displayed, with the retrieved formula instance highlighted in each

[17] See, for example, MathDeck [16], in which candidate formulae are suggested to the users during formula editing.

case. For example, the formula $\sum_{n=1}^{\infty} a_n$ shown in Figure 6 was retrieved both in an answer post (shown first) and in a question post (shown second). As in Task 1, buttons are provided for the assessor to record their judgment; unlike Task 1, judgments for each instance of the same retrieved formula (up to 5) are recorded separately, and later used to produce a *single* (max) score for each visually distinct formula.

Assessor training. After some initial work on assessment for Task 1, 3 assessors were reassigned to to perform relevance judgements for Task 2, with the remaining 5 continuing to do relevance judgments for Task 1. Two rounds of training were performed.

In the first training round, the assessors were familiarized with the task. To illustrate how formula search might be used, we interactively demonstrated formula suggestion in MathDeck [16] and the formula search capability of Approach0 [23]. Then the task was defined using examples, showing a formula query with some retrieved results, talking through the relevance definitions and how to apply those definitions in specific cases. During the training session, the assessors saw different example results for topics and discussed their relevance based on criteria defined for them with the organizers. They also received feedback from the third author, an expert MSE user. To prepare the judgment pools used for this purpose, we pooled actual submissions from participating teams, but only to depth 10 (i.e., 10 different formulae) for primary runs and the baseline run, and 5 different formulae for alternate runs. The queries used for this initial assessor training were omitted from the final Task 2 query set on which systems were evaluated because they were not judged on full-sized pools.

All three assessors were then assigned two complete Task 2 pools (for topics B.46 and B.98) to independently assess; these topics were not removed from the collection. After creating relevance judgments for these full-sized pools, the assessors and organizers met by Zoom to discuss and resolve disagreements. The assessors used this opportunity to refine their understanding of the relevance criteria, and the application of those criteria to specific cases. Annotator agreement was found to be fairly good (kappa=0.83). An adjudicated judgment was recorded for each disagreement, and used in the final relevance judgment sets for these two topics.

The assessors were then each assigned complete pools to judge for four topics, one of which was also assessed independently by a second assessor. The average kappa on the three dual-assessed topics was 0.47. After discussion between the organizers and the assessors, adjudicated disagreements were recorded and used in the final relevance judgments. The assessors then performed the remaining assessments for Task 2 independently.

Assessment. A total of 47 topics were assessed for Task 2. Two queries (B.58 and B.65) had fewer than two relevant answers after H+M binarization and were removed. Of the remaining 45 queries, an average of 125.0 formulae were assessed per topic, with an average assessment time of 38.1 seconds per formulae. The average number of formulae instances labeled as relevant after

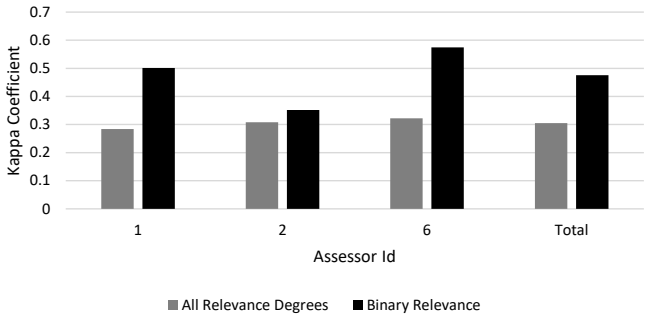

Fig. 5. Inter-annotator agreement (Cohen's kappa) over 3 assessors after official Task 2 assessment. Each annotator evaluated two tasks completed by the other two annotators. Shown are four-way classification (gray) and two-way (H+M binarized) classification (black). Results are provided for each individual Task 1 assessor (as average kappa score), along with the average kappa values over all assessors at right ('Total').

H+M+L binarization was 43.1 per topic, with the highest being 115 for topic B.60 and the lowest being 7 for topics B.56 and B.32.

Post Assessment. As we did for Task 1, after assessment for Task 2 was completed, each of the three assessors were given two topics, one completed by each of the other two annotators. Figure 5 shows the Cohen's kappa coefficient values for each assessor and total agreement over all of them. A kappa of 0.30 was achieved on the four-way assessment task, and with H+M binarization the average kappa value was 0.48. Interestingly, the post-assessment agreement between assessors is about the same as Task 1 for four-way agreement (0.29), but H+M binarized agreement is almost 10% higher than Task 1 (0.39). When asked, assessors working on Task 2 (who had all been previously trained on Task 1) reported finding Task 2 assessment to be easier. We note that there were fewer assessors working on Task 2 than Task 1 (3 vs. 5 assessors).

Additional Training Topics. After the official assessment, to increase the size of the available dataset, an additional 27 topics were annotated. These are available in the ARQMath dataset, and can be used for training models. As a result, 74 topics have been published for Task 2.

5.5 Evaluation Measures

As for Task 1, the primary evaluation measure for Task 2 is nDCG′, and MAP′ and P@10 were also computed. Participants submitted ranked lists of formula instances, but we computed these measures over visually distinct formulae. To do this, we replaced each formula instance with its associated visually distinct formula, then deduplicated from the top of the list downward to obtain a ranked list of visually distinct formulae, and then computed the evaluation measures. As explained above, the relevance score for each visually distinct formula was computed as the maximum score over each assessed instance of that formula.

5.6 Results

Table A2 in the appendix shows the results, with the baseline run shown first, and then teams and their systems ranked by nDCG′. No team did better than the baseline system as measured by nDCG′ or MAP′, although the DPRL team did achieve the highest score for P@10.

6 Conclusion

The ARQMath lab is the first shared-task evaluation exercise to explore Community Question Answering (CQA) for mathematical questions. Additionally, the lab introduced a new formula retrieval task in which both the query and retrieved formulae are considered within the context of their question or answer posts, and evaluation is performed using visually distinct formulas, rather than all formulas returned in a run. For both tasks, we used posts and associated data from the Math Stack Exchange (MSE) CQA forum.

To reduce assessor effort and obtain a better understanding of the relationship between mathematical CQA and formula search, the formulae used as formula search topics were selected from the Task 1 (CQA) question topics. This allowed us to increase coverage for the formula retrieval task by using relevant posts found in the CQA evaluations as candidates for assessment. To enrich the judgments pools for the CQA task, we added answer posts from the original topic question thread and threads identified as duplicate questions by the MSE moderators.

In total, 6 teams submitted 29 runs: 5 teams submitted 18 runs for the CQA task (Task 1), and 4 teams submitted 11 runs for the formula retrieval task (Task 2). We thus judge the first year of the ARQMath lab to be successful. Each of these teams had some prior experience with math-aware information retrieval; in future editions of the lab we hope to further broaden participation, particularly from the larger IR and NLP communities.

Our assessment effort was substantial: 8 paid upper-year or recently graduated undergraduate math students worked with us for over a month, and underwent training in multiple phases. Our training procedure provided our assessors with an opportunity to provide feedback on relevance definitions, the assessment interface, and best practices for assessment. In going through this process, we learned that 1) the CQA task is much harder to assess than the formula retrieval task, as identifying non-relevant answers requires more careful study than identifying non-relevant formulae, 2) the breadth of mathematical expertise needed for the CQA task is very high; this led us to having assessors indicate which questions they wished to assess and us assigning topics according to those preferences (leaving the 10 topics that no assessor requested unassessed), and 3) having an expert mathematician (in this case, a math Professor) involved was essential for task design, clarifying relevance definitions, and improving assessor consistency.

To facilitate comparison with systems using ARQMath for benchmarking in the future, and to make use of our graded relevance assessments, we chose

nDCG′ [18] as the primary measure for comparing systems. Additional metrics (MAP′ and Precision at 10) are also reported to provide a more complete picture of system differences.

Overall, we found that systems submitted to the first ARQMath lab generally approached the task in similar ways, using both text and formulae for Task 1, and (with two exceptions) operating fully automatically. In future editions of the task, we hope to see a greater diversity of goals, with, for example, systems optimized for specific types of formulae, or systems pushing the state of the art for the use of text queries to find math. We might also consider supporting a broad range of more specialized investigations by, for example, creating subsets of the collection that are designed specifically to formula variants such as simplified forms or forms using notation conventions from different disciplines. Our present collection includes user-generated tags, but we might also consider defining a well-defined tag set to indicate which of these types of results are desired.

Acknowledgements. Wei Zhong suggested using Math Stack Exchange for benchmarking, made Approach0 available for participants, and provided helpful feedback. Kenny Davila helped with the Tangent-S formula search results. We also thank our student assessors from RIT: Josh Anglum, Wiley Dole, Kiera Gross, Justin Haverlick, Riley Kieffer, Minyao Li, Ken Shultes, and Gabriella Wolf. This material is based upon work supported by the National Science Foundation (USA) under Grant No. IIS-1717997 and the Alfred P. Sloan Foundation under Grant No. G-2017-9827.

References

1. Aizawa, A., Kohlhase, M., Ounis, I.: NTCIR-10 math pilot task overview. In: NTCIR (2013)
2. Aizawa, A., Kohlhase, M., Ounis, I., Schubotz, M.: NTCIR-11 Math-2 task overview. In: NTCIR. vol. 11, pp. 88–98 (2014)
3. Borlund, P.: The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. Information Research **8**(3), 8–3 (2003)
4. Buckley, C., Voorhees, E.M.: Retrieval evaluation with incomplete information. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 25–32 (2004)
5. Davila, K., Zanibbi, R.: 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. pp. 1165–1168 (2017)
6. Guidi, F., Coen, C.S.: A survey on retrieval of mathematical knowledge. In: CICM. Lecture Notes in Computer Science, vol. 9150, pp. 296–315. Springer (2015)
7. Hopkins, M., Le Bras, R., Petrescu-Prahova, C., Stanovsky, G., Hajishirzi, H., Koncel-Kedziorski, R.: SemEval-2019 Task 10: Math Question Answering. In: Proceedings of the 13th International Workshop on Semantic Evaluation (2019)
8. Kaliszyk, C., Brady, E.C., Kohlhase, A., Coen, C.S. (eds.): Intelligent Computer Mathematics - 12th International Conference, CICM 2019, Prague, Czech Republic, July 8-12, 2019, Proceedings, Lecture Notes in Computer Science, vol. 11617. Springer (2019)

9. Kincaid, J.P., Fishburne Jr, R.P., Rogers, R.L., Chissom, B.S.: Derivation of new readability formulas (automated readability index, fog count and Flesch reading ease formula) for Navy enlisted personnel. Tech. rep., Naval Technical Training Command Millington TN Research Branch (1975)
10. Kushman, N., Artzi, Y., Zettlemoyer, L., Barzilay, R.: Learning to automatically solve algebra word problems. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (2014)
11. Ling, W., Yogatama, D., Dyer, C., Blunsom, P.: Program induction by rationale generation: Learning to solve and explain algebraic word problems. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (2017)
12. Mansouri, B., Agarwal, A., Oard, D., Zanibbi, R.: Finding old answers to new math questions:the ARQMath lab at CLEF 2020. In: European Conference on Information Retrieval (2020)
13. Mansouri, B., Rohatgi, S., Oard, D.W., Wu, J., Giles, C.L., Zanibbi, R.: Tangent-CFT: An embedding model for mathematical formulas. In: Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval (ICTIR). pp. 11–18 (2019)
14. Mansouri, B., Zanibbi, R., Oard, D.W.: Characterizing searches for mathematical concepts. In: Joint Conference on Digital Libraries (2019)
15. Newell, A., Simon, H.: The logic theory machine–a complex information processing system. IRE Transactions on information theory (1956)
16. Nishizawa, G., Liu, J., Diaz, Y., Dmello, A., Zhong, W., Zanibbi, R.: MathSeer: A math-aware search interface with intuitive formula editing, reuse, and lookup. In: European Conference on Information Retrieval. pp. 470–475. Springer (2020)
17. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier information retrieval platform. In: European Conference on Information Retrieval. pp. 517–519. Springer (2005)
18. Sakai, T., Kando, N.: On information retrieval metrics designed for evaluation with incomplete relevance assessments. Information Retrieval **11**(5), 447–470 (2008)
19. Schubotz, M., Youssef, A., Markl, V., Cohl, H.S.: Challenges of mathematical information retrieval in the NTCIR-11 Math Wikipedia Task. In: SIGIR. pp. 951–954. ACM (2015)
20. Zanibbi, R., Aizawa, A., Kohlhase, M., Ounis, I., Topic, G., Davila, K.: NTCIR-12 MathIR task overview. In: NTCIR (2016)
21. Zanibbi, R., Blostein, D.: Recognition and retrieval of mathematical expressions. International Journal on Document Analysis and Recognition (IJDAR) **15**(4), 331–357 (2012)
22. Zhong, W., Rohatgi, S., Wu, J., Giles, C.L., Zanibbi, R.: Accelerating substructure similarity search for formula retrieval. In: ECIR (1). Lecture Notes in Computer Science, vol. 12035, pp. 714–727. Springer (2020)
23. Zhong, W., Zanibbi, R.: Structural similarity search for formulas using leaf-root paths in operator subtrees. In: European Conference on Information Retrieval. pp. 116–129. Springer (2019)

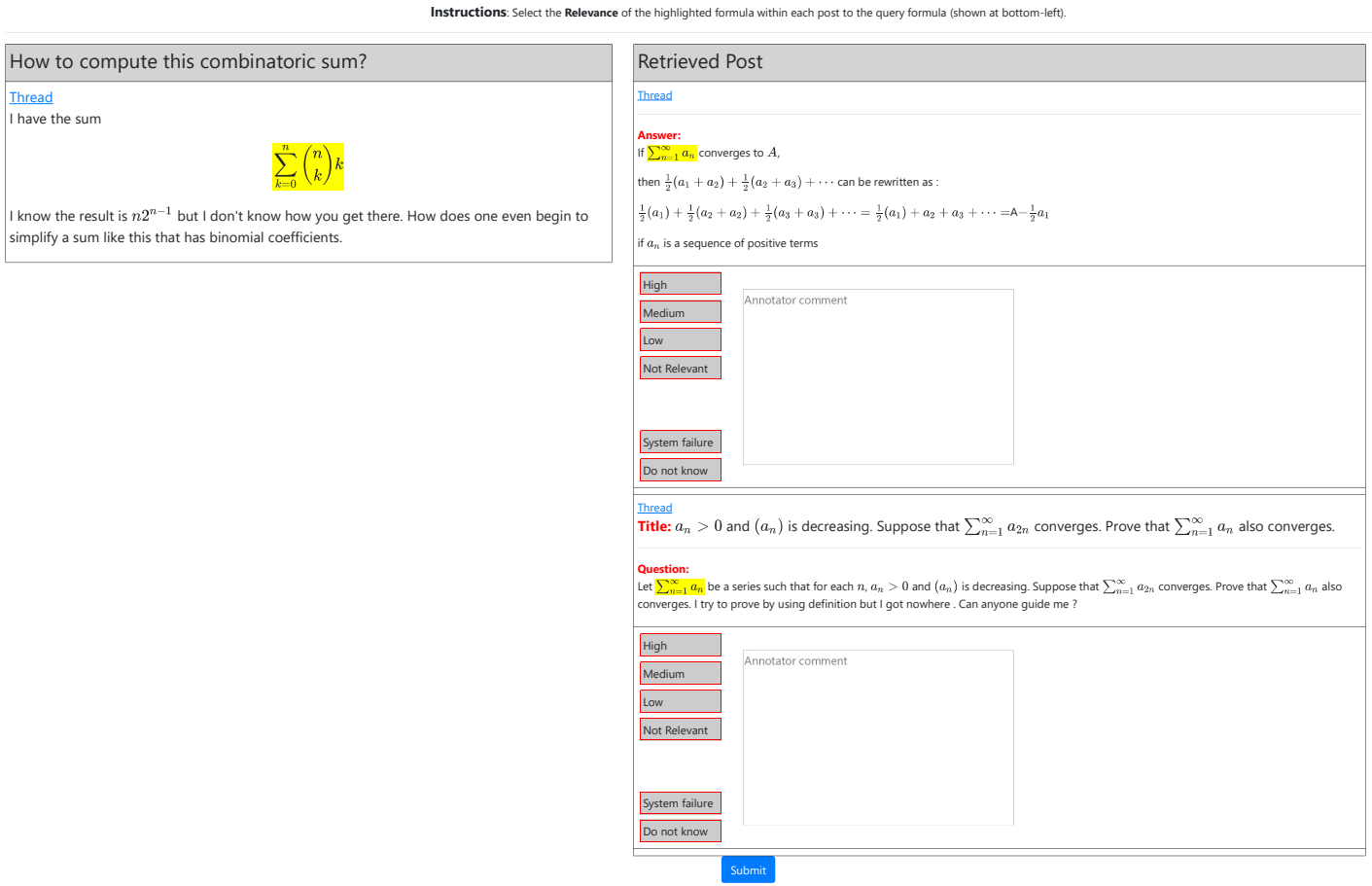

Fig. 6. Turkle Assessment Interface. Shown are hits for Formula Retrieval (Task 2). In the left panel, the formula query is highlighted. In the right panel, one answer post and one question post containing the same retrieved formula are shown. For Task 1, a similar interface was used, but without formula highlighting, and just one returned answer post viewed at a time.

A Appendix: Evaluation Results

Table A1. Task 1 (CQA) results, averaged over 77 topics. **P** indicates a primary run, **M** indicates a manual run, and **(✓)** indicates a baseline pooled at the primary run depth. For Precision@10 and MAP, H+M binarization was used. The best baseline results are in parentheses. * indicates that one baseline did not contribute to judgment pools.

Run	Data	Run Type		Evaluation Measures		
		P	M	nDCG′	MAP′	P@10
Baselines						
Linked MSE posts	n/a	(✓)		(**0.279**)	(**0.194**)	(**0.384**)
*Approach0**	Both		✓	0.250	0.099	0.062
TF-IDF + Tangent-S	Both	(✓)		0.248	0.047	0.073
TF-IDF	Text	(✓)		0.204	0.049	0.073
Tangent-S	Math	(✓)		0.158	0.033	0.051
MathDowsers						
alpha05noReRank	Both			**0.345**	**0.139**	**0.161**
alpha02	Both			0.301	0.069	0.075
alpha05translated	Both		✓	0.298	0.074	0.079
alpha05	Both	✓		0.278	0.063	0.073
alpha10	Both			0.267	0.063	0.079
PSU						
PSU1	Both			0.263	0.082	0.116
PSU2	Both	✓		0.228	0.054	0.055
PSU3	Both			0.211	0.046	0.026
MIRMU						
Ensemble	Both			0.238	0.064	0.135
SCM	Both	✓		0.224	0.066	0.110
MIaS	Both	✓		0.155	0.039	0.052
Formula2Vec	Both			0.050	0.007	0.020
CompuBERT	Both	✓		0.009	0.000	0.001
DPRL						
DPRL4	Both			0.060	0.015	0.020
DPRL2	Both			0.054	0.015	0.029
DPRL1	Both	✓		0.051	0.015	0.026
DPRL3	Both			0.036	0.007	0.016
zbMATH						
zbMATH	Both	✓	✓	0.042	0.022	0.027

Table A2. Task 2 (Formula Retrieval) results, averaged over 45 topics and computed over deduplicated ranked lists of visually distinct formulae. **P** indicates a primary run, and **(✓)** shows the baseline pooled at the primary run depth. For MAP and P@10, relevance was thresholded H+M binarization. All runs were automatic. Baseline results are in parentheses.

			Evaluation Measures		
Run	Data	P	nDCG′	MAP′	P@10
Baseline					
Tangent-S	Math	(✓)	**(0.506)**	**(0.288)**	**(0.478)**
DPRL					
TangentCFTED	Math	✓	**0.420**	**0.258**	**0.502**
TangentCFT	Math		0.392	0.219	0.396
TangentCFT+	Both		0.135	0.047	0.207
MIRMU					
SCM	Math		0.119	0.056	0.058
Formula2Vec	Math	✓	0.108	0.047	0.076
Ensemble	Math		0.100	0.033	0.051
Formula2Vec	Math		0.077	0.028	0.044
SCM	Math	✓	0.059	0.018	0.049
NLP_NITS					
formulaembedding	Math	✓	0.026	0.005	0.042