Dowsing for Math Answers: Exploring MathCQA with a Math-aware Search Engine

by

Yin Ki Ng

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2021

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I was the sole author for Chapters 1, 2, 5, 6, and 7 of this thesis, which were written under the supervision of Dr. Frank Tompa. Exceptions to sole authorship of material are as follows:

Research presented in Chapter 3: Dallas Fraser implemented the first version of Tangent-L (Section 3.1). Besat Kassaie implemented many subsequent features of Tangent-L, including SLT tree conversion to math tokens, extraction of repetition tokens, and formula normalizations (Section 3.2 and 3.3). I helped revise ranking formulas to improve Tangent-L's formula matching capability (Section 3.2.2 and 3.4.2). I was the sole author of the descriptions of the work in the chapter, but I relied extensively on documentations and informal descriptions from Dallas Fraser and Besat Kassaie.

Research presented in Chapter 4: Besat Kassaie conducted the study of proximity signals and created the proximity re-ranking run for ARQMath-2 (Section 4.3.2). Andrew Kane suggested capitalizing on proximity by indexing document fragments (Section 4.4.7). Mirette Marzouk, George Labahn, Avery Hiebert, and Kelvin Wang offered insights during our research group's meetings. I contributed to the design and the implementation of the rest of the submissions for ARQMath-1 and ARQMath-2, and the subsequent experimental analyses. Again, I was the sole author of the chapter, but I built upon previously co-authored papers (see below).

Some material forming the core of this thesis was written in three manuscripts for publication. I was the primary author and the publications were written with significant editorial contributions from Dr. Frank Tompa. The publications are:

Yin Ki Ng, Dallas J. Fraser, Besat Kassaie, George Labahn, Mirette S. Marzouk, Frank Wm. Tompa, and Kevin Wang.
Dowsing for Math Answers with Tangent-L,
in: CLEF 2020, volume 2696 of CEUR Workshop Proceedings, 2020

Yin Ki Ng, Dallas J. Fraser, Besat Kassaie, Frank Wm. Tompa.
Dowsing for Answers to Math Questions: Ongoing Viability of Traditional MathIR ,
in: CLEF 2021, volume 2936 of CEUR Workshop Proceddings, 2021

Yin Ki Ng, Dallas J. Fraser, Besat Kassaie, Frank Wm. Tompa.
Dowsing for Math Answers,
in: Candan K.S. et al. (eds) Experimental IR Meets Multilinguality, Multimodality, and Interaction. Lecture Notes in Computer Science, vol 12880. Springer, Cham.

Abstract

Solving math problems can be challenging. It is so challenging that one might wish to seek insights from the internet, looking for related references to understand more about the problems. Even more, one might wish to *actually search for the answer*, believing that some wise people have already solved the problem and shared their intelligence selflessly. However, searching for relevant answers for a math problem effectively from those sites is itself not trivial.

This thesis details how a *math-aware* search engine Tangent-L—which adopts a traditional text retrieval model (Bag-of-Words scored by BM25$^+$) using formulas' symbol pairs and other features as "words"—tackles the challenge of *finding answers to math questions*. Various adaptations for Tangent-L to this challenge are explored, including query conversion, weighting scheme of math features, and result re-ranking.

In a recent workshop series named Answer Retrieval for Questions on Math (ARQ-Math), and with math problems from Math StackExchange, the submissions based on these adaptations of Tangent-L achieved the best participant run for two consecutive years, performing better than many participating models designed with machine learning and deep learning models. The major contributions of this thesis are the design and implementation of the three-stage approach to adapting Tangent-L to the challenge, and the detailed analyses of many variants to understand which aspects are most beneficial. The code repository is available[1], as is a data exploration tool[2] built for interested participants to view the math questions in this ARQMath challenge and check the performance of their answer rankings.

[1] https://github.com/kiking0501/MathDowsers-ARQMath
[2] https://cs.uwaterloo.ca/~yk2ng/MathDowsers-ARQMath

Acknowledgements

First and foremost, to my supervisor Dr. Frank Tompa, for his continuous patience, encouragement, and guidance throughout the time; and to Dr. George Labahn and Dr. Mark Smucker for their valuable feedback as readers of this thesis.

I would like to thank Besat Kassaie, Mirette Marzouk, and Dallas Fraser for providing guidance to work with Tangent-L's implementation; and George Labahn and Avery Hiebert for the stimulating discussions during the research group's regular meetings.

The ARQMath Lab organizers (notably, Behrooz Mansouri) prepared the dataset, the manual translation of the topic questions into formulas and keywords, and offered assistance to participants from time to time. Andrew Kane and anonymous reviewers made valuable suggestions for improving our presentations and for potential future directions.

This research is funded by the Waterloo-Huawei Joint Innovation Lab and NSERC, the Natural Science and Engineering Research Council of Canada.

Dedication

This thesis is dedicated to my parents for their unconditional love and trust; to my fiancé, Pak Hay, for his support in my pursuit, and his affection and care that took me through many times of stress; and to my friends in Hong Kong, for their companionship during their hard time over the last few years.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction: from MathIR to MathCQA

Mathematical Information Retrieval (MathIR) is one of the domain-specific applications of Information Retrieval (IR), the process of retrieving documents relevant to input queries. The goal of MathIR is to develop "math-aware" search engines capable of searching math documents — documents characterized by the presence of formulas in additional to natural language text.

A series of MathIR evaluation workshops were held as part of NTCIR from 2010 to 2012 to encourage the development of math-aware search engines [2, 3, 60]. The NTCIR MathIR task requires participating systems to retrieve documents from a corpus of arXiv or Wikipedia articles with respect to queries that consist of one or more formulas with or

Figure 1: *MathDowsers*: researchers dowsing for math documents.

without keywords. The most recent task data from NTCIR-12 has served as a benchmark for researchers in the MathIR research community to improve their math-aware search engines.

However, the NTCIR MathIR task, which is an ad hoc retrieval task that looks for related documents from given query terms, is not the only task that can be used to evaluate the effectiveness of a math-aware search engine. Generically, the effectiveness of a search engine might be determined by various other tasks, each aiming to satisfy a different information need. One such task is Question Answering. As a cross-disciplinary task between IR and Natural Language Processing (NLP), Question Answering might be reformulated as a retrieval task where the target information to be retrieved is a ranked list of answers with respect to a question posed in a natural language. Similarly, a math domain-specific application of Question Answering can be used as a task that helps evaluate the effectiveness of a math-aware search engine.

The more recent ARQMath Lab series, whose name stands for "Answer Retrieval for Questions on Math", provides such an evaluation platform for math-aware search engines [62, 30]. Running at the Conference and Labs of the Evaluation Forum (CLEF) in 2020 and 2021, and to be held again in 2022, this Lab provides the first Mathematical Community Question Answering (MathCQA) task involving real-life math questions selected from a Community Question Answering (CQA) forum, the Math StackExchange (MSE) site. The MathCQA task asks participating systems to retrieve answers from previous questions in the same forum that might be potential answers to given math questions. The Lab also has an in-context formula retrieval task, in which the participating systems are asked to retrieve useful formulas in the forum with respect to an identified formula of the given math questions. Both tasks have served to encourage the MathIR research community to extend its research together with modern NLP development to design an effective math-aware search engine suited for the tasks.

Motivated by the Lab series, the goal of this thesis is to study MathCQA:

> **Given a math question — expressed in mathematical natural language — how can a math-aware search engine be designed to be effective in retrieving potential answers from a MathCQA forum?**

This thesis describes the research work developed by the MathDowsers team (Figure 1) for the ARQMath Lab series in CLEF 2020 and 2021[1]. A system was designed and implemented for the MathCQA challenge, and the submitted runs from this system achieved the best participant runs in both years. This research work provides the following contributions:

- it proposes an effective three-stage approach which adapts a math-aware search engine to the MathCQA task;
- it details the analysis of many variants of this approach to suggest which aspects are most beneficial;
- it demonstrates the effectiveness of Tangent-L, a math-aware search engine that was first proposed by Fraser et al. [14] and continues to be developed by the team;
- it shows that a traditional MathIR system remains a competitive and viable option for the MathCQA task, even when compared to other systems involving machine learning techniques and deep learning algorithms.

The outline of this thesis is as follows: Chapter 2 introduces MathIR and CQA, and provides a background on the ARQMath Lab series; Chapter 3 describes the math-aware search engine Tangent-L and its variants motivated by the ARQMath Lab challenges; Chapters 4 examines the ARQMath MathCQA task, followed by Chapter 5 which focuses instead on the ARQMath In-context Formula Retrieval task. Chapter 6 outlines the design and implementation of an accompanying user interface for the search engine to address the problem of data exploration. Chapter 7 closes with a discussion of what conclusions can be drawn and what future work might be pursued.

[1]By analogy to water dowsing, *MathDowsers* refers to a group of researchers from the University of *Water*loo looking for *math* documents

Chapter 2

Background

2.1 Mathematical Information Retrieval (MathIR)

Formal mathematics and formulas play a crucial role in scientific and engineering documents to express concepts or ideas. Searching for these mathematical expressions and documents with conventional text search engines, however, can be ineffective due to the inability to capture distinctive characteristics inherent only in formulas. Developing specialized search engines that are *math-aware* is thus the research goal of the Mathematical Information Retrieval (MathIR) community.

The development of math-aware search engines has proceeded ever since the first proposal by Miller and Youssef in 2003 [33]. Survey papers and overviews of the field are available [16, 47, 1]. The following subsections outline some of the core details of the research field.

2.1.1 Formula Representations

While formula processing is the key component of math-aware search engines, formulas in digital documents can be encoded in various ways. Other than the LaTeX encoding, MathML is an encoding from the W3C recommendation [34] for exchanging mathematics among software tools. It is a markup language cast as an application of XML and encodes formulas in two tree-like representations (Table 1):

Presentation MathML	*Content MathML*

`<mrow>`
`  <mrow>`
`  <msup>`
`  <mi>x</mi>`
`  <mn>2</mn>`
`  </msup>`
`  <mo>+</mo>`
`  <mrow>`
`  <mn>4</mn>`
`  <mo>&InvisibleTimes;</mo>`
`  <mi>x</mi>`
`  </mrow>`
`  <mo>+</mo>`
`  <mn>4</mn>`
`  </mrow>`
`  <mo>=</mo>`
`  <mn>0</mn>`
`</mrow>`
	`<apply>`
	`  <eq/>`
	`  <apply>`
	`  <plus/>`
	`  <apply>`
	`  <power/>`
	`  <ci>x</ci>`
	`  <cn>2</cn>`
	`  </apply>`
	`  <apply>`
	`  <times/>`
	`  <cn>4</cn>`
	`  <ci>x</ci>`
	`  </apply>`
	`  <cn>4</cn>`
	`  </apply>`
	`  <cn>0</cn>`
	`</apply>`

Table 1: Representations for the formula $x^2 + 4x + 4 = 0$. Presentation MathML uses presentational tags that "... generally start with "m" and then use "o" for operator "i" for identifier "n" for number, and so on. The "mrow" tags indicate organization into horizontal groups." Content MathML uses semantic tags that "... take into account such concepts as "times", "power of" and so on". [34]

Presentation MathML, which captures a Symbol Layout Tree (SLT), is the visual structure of a mathematical expression with a two-dimensional layout;

Content MathML, which captures an Operator Tree (OPT), captures the hierarchy of underlying mathematical concepts.

Fundamental differences exist among these formula representations. LaTeX is a linearized encoding reflecting a formula's syntax, while the two forms of MathML are tree-based encodings. Both LaTeX and Presentation MathML encode visual information, while Content MathML encodes the semantic mathematical meaning of formulas. Common tools are available for converting LaTeX to MathML (such as LaTeXML[1]), however, conversion failure might occur, especially for Content MathML, which requires an interpretation of semantics from visual information.

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

Each of these differences brings pros and cons to math-aware search engines. For instance, the LaTeX encoding is already linearized and thus can be easily incorporated into existing text search approaches; however its representation only implicitly reflects the structural information embedded in tree-like encodings. Presentation MathML provides a richer visual information and has great compatibility with LaTeX, but it does not encode the semantic mathematical meaning of the formula. A single or parallel markup with Content MathML, however, has to deal with a higher degree of ambiguity encountered during the conversion for this encoding. The choice of formula representation(s) and the associated input processing thus dictate the capability of a math-aware search engine.

2.1.2 Retrieval Models

Major approaches to design a retrieval model of math-aware search engines can be broadly categorized as:

1. structure-based searching via trees (tree search); or
2. reduction to text retrieval model (text search); or
3. a hybrid combination of both (hybrid).

While the actual models vary, the remainder of this section describes some commonly-seen techniques in the latest math-aware search engines developed for the ARQMath Lab series (Section 2.5).

Tree Search: Operations on Tree Structures

With tree-like encodings, each complete tree and all partial trees of a formula can be used for formula matching. Two trees can be compared by:

Tree-Edit Distance: The tree-edit distance refers to the minimal-cost sequence of node edit operations that transforms one tree to another. Common node edit operations in consideration are insertion, deletion, and substitution. A tree-edit distance score can be defined accordingly, with custom weights assigned for each node edit operation.

Alternatively, a tree-like encoding can be linearized by decomposing into a set of *leaf-root* paths, or by its traversal path in the *infix, prefix*, or *postfix* order.

Text Search: Bag-of-Words Model

A bag-of-words model represents a textual document by the bag of its words that disregards the word order but keeps the word counts. Under this model, a document's score with respect to a query can be represented by a weighted sum of the scores of the terms making up the document. Two traditional scoring functions are:

TF-IDF [23, 18]: The Term Frequency-Inverse Document Frequency (TF-IDF) estimates document relevancy with numerical statistics of the query term. A query term contributes more weight to the document that contains it if either it frequently appears within the document (a higher Term-Frequency) or if it is rarer in the overall collection of documents (a lower Document-Frequency). A simple form of TF-IDF for a collection of documents D, a query q, and a document $d \in D$ is given by

$$\text{TF-IDF}(q, d) = \sum_{w \in q} \frac{tf_{w,d}}{|d|} \log\left(\frac{|D|}{|D_w|}\right) \tag{2.1}$$

where $tf_{w,d}$ is the term frequency of w in document d, $|d|$ is the document length, and $|D_w|$ is the document frequency of w—the number of documents in D containing w.

Okapi BM25 [52]: Introduced in 1994, Okapi BM25 is a scoring function inspired by the Probability Ranking Principle [51], and it scores documents by a decreasing *estimated*-probability of document relevancy. Similar to TF-IDF, Okapi BM25 also has a Term-Frequency component to reward term frequency and an Inverse-Document-Frequency component to penalize document frequency for each query term in a document, but the components are designed to also account for term frequency saturation and document length normalization. More specifically, it is defined as

$$\text{BM25}(q, d) = \sum_{w \in q} \left(\frac{(k+1)tf_{w,d}}{k\left(1.0 - b + b\frac{|d|}{\overline{d}}\right) + tf_{w,d}}\right) \log\left(\frac{|D| - |D_w| + 0.5}{|D_w| + 0.5}\right) \tag{2.2}$$

where $\overline{d}$ is the average document length, and k and b are constants that are commonly set to 1.2 and 0.75, respectively, without specific data training.

Both score functions are "traditional" text ranking functions that have a long history and have been widely implemented in popular open-source text search platforms such as Lucene, Solr, and ElasticSearch.[2]

[2]Okapi BM25 has replaced TF-IDF to become the default ranking since Lucene version 6.0 in 2016.

Text Search: Vector Space Model

A vector space model encodes a textual document as a vector, where each dimension corresponds to a word in the document. For example, a document can be simply encoded by a one-hot vector of the length of total vocabularies, where each dimension of the one-hot vector has a value of one if its corresponding word exists in the document, otherwise zero.

With this model, the document relevancy with respect to a query can be represented by a similarity score between the document vector $\mathbf{d}$ and the query vector $\mathbf{q}$, which is usually measured by:

Cosine Similarity:

$$\frac{\mathbf{d} \cdot \mathbf{q}}{\|\mathbf{d}\|\|\mathbf{q}\|} \tag{2.3}$$

A common vector space model encodes a document by a TF-IDF vector. That is, each dimension of the vector is the TF-IDF score of its corresponding word.

Text Search: Word Embeddings

A document vector in a vector space model can also be computed (say, taking the average) from a set of vectors where each vector represents a word that makes up the document. The set of vectors is referred to as *word vectors*. Word vectors can be learned effectively from a text corpus, such that words having a similar meaning would have similar vectors [31], in which case, the word vectors are more often referred to as *word embeddings*. Common word embedding algorithms include:

word2Vec [32] (2013): trained with shallow neural networks by either taking the context of each word as the input and to predict the word corresponding to the context (the Continuous Bag-of-Word (CBOW) model), or by predicting the context words of an input word (the Skip Gram model).

GloVe [46] (2014): while word2Vec "sees" only a local context window each time during training, GloVe first builds a word co-occurrence matrix for the entire corpus, followed by matrix factorization to yield a lower-dimensional matrix that contains the word embeddings.

fastText [8] (2016): incorporates n-gram subwords (that is, characters in the original word with a window size n) when training word2vec, so that rare or unseen words are represented more appropriately.

Text Search: Transformers and the BERT Family

One drawback of word embeddings is that each word can only have one fixed meaning (one vector) across the whole corpus, regardless of its context[3]. The Transformer architecture by Vaswani et al. [58] in 2017, on the other hand, is a deep learning model architecture that handles sequential input of words, with an Encoder component that adopts a *self-attention* mechanism to learn the weight of a word considering its surrounding words at input time.

Bidirectional Encoder Representations from Transformers (BERT) proposed by Devlin et al., [13] in 2018, as its name suggests, is built on top of Transformer Encoders. Given a large corpus, it trains on the Masked Language Model (MLM) task and the Next Sentence Prediction (NSP) task with bidirectional layers to learn word representations in context. The pre-trained neural networks can then be *transferred*, that is, they can be equipped with additional layers and fine-tuned to apply to various downstream tasks, such as IR applications as detailed by Lin et al. [21] in 2020.

Although BERT achieves state-of-the-art performance on many natural language understanding tasks, it suffers from being heavily compute-intensive. Variants of BERT have been proposed to improve its efficiency on top of effectiveness. Some of the variants are:

RoBERTa [22] (2019): A Robustly optimized BERT approach that proposes modifications to the BERT pre-training procedure to improve end-task performance.

ALBERT [20] (2019): A Lite BERT approach with parameter-reduction techniques to lower memory consumption and to increase the training speed.

SentenceBERT [49] (2020): An approach to optimize the computation overhead specifically for finding similar sentences in a corpus (or any pair of textual input that wants to be compared by cosine similarity), by deriving semantically meaningful sentence embeddings.

ColBERT [19] (2020): The Contextualized Late Interation over BERT approach speeds up query processing in document retrieval, by first introducing a late interaction architecture that independently encodes the query and the document using BERT; followed by an interaction step that can effectively model their fine-grained similarity.

[3]Think of "Apple"—is it food or a technology company?

Hybrid: Ensembling Ranked Results

Finally, results from different models can be ensembled to create a single ranked list with:

The Reciprocal Rank Fusion (RRF) [10]:

$$RRF(d \in D) = \sum_{m \in M} \frac{1}{k + r_m(d)} \tag{2.4}$$

where D is the collection of documents, M is the set of models, r_m is the rank, and k is a parameter determined by the dataset.

2.1.3 Effectiveness Measures

The effective measures in MathIR share common measures across the IR community as follows:

Precision and Recall: Precision is the fraction of relevant documents among all retrieved documents, and recall is the fraction of relevant documents that have been retrieved among all relevant documents. It is, in general, a challenge to get good performance on both measures, since an increase in recall—usually achieved by retrieving more documents to get the relevant documents—lowers the precision when more irrelevant documents are recovered as well.

Another form of precision is Precision@k (P@k), the calculation of which is based only on the top-k documents retrieved. P@k is closely related to the user experience since usually only top-k but not all presented documents might actually be examined by a user during searching.

MAP: The Mean Average Precision (MAP) is the mean of Average Precision (AP). For each query with n retrieved documents, AP is defined as:

$$\text{AP} = \frac{\sum_{k=1}^{n} \Big(\text{P}(k) \cdot rel(k) \Big)}{\text{number of relevant documents}} \tag{2.5}$$

where P(k) is the precision calculated up to the top-k retrieved documents, and $rel(k)$ is one if a document at rank k is relevant and zero otherwise. MAP is then computed as the average of AP over all queries. Compared to precision and recall, MAP considers also the order of the relevant documents in the retrieved list.

bpref: The binary preference-based measure (bpref) is a measure that takes into account the existence of *unjudged* documents in a ranked list. With the previous measures, the unjudged documents are deemed to be irrelevant and thus the evaluation power of those measures are sometimes misleading. On the other hand, bpref considers only judged documents for evaluation. For a given collection of documents R judged as relevant and a given collection of documents N judged as irrelevant, bpref is defined as,

$$\text{bpref} = \frac{1}{|R|} \sum_{r \in R} \left(1 - \frac{|n \in N \text{ that ranked higher than } r|}{\min(|R|, |N|)} \right) \tag{2.6}$$

nDCG: The normalized Discount Cumulative Gain (nDCG) is a measure for graded relevance judgements. It is computed as follows: each retrieved document first earns a gain value from the graded relevance judgement, discounted by a decaying function of the rank position of each document. The summation of the gain values is the Discount Cumulative Gain (DCG):

$$\text{DCG}_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)} \tag{2.7}$$

where rel_i is the graded relevance of the result at position i. nDCG is then calculated by dividing by the *ideal* Discounted Cumulative Gain:

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p}, \text{ where } \quad \text{IDCG}_p = \sum_{i=1}^{REL_p} \frac{rel_i}{\log_2(i+1)} \tag{2.8}$$

with REL_p representing the list of relevant documents (in decreasing order of their relevance) in the whole collection up to position p.

The nDCG′ [55] (read as "nDCG"-prime) extends the nDCG measure. The only difference when computing the nDCG′ is that unjudged documents are removed from the ranked list before performing the computation. It is shown that nDCG′ has somewhat better discriminating power and a better system ranking stability than the bpref measure with judgement ablation.

Similar to nDCG′, MAP′ and P′@k are identical to MAP and P@k, respectively, but with unjudged documents removed from the list of retrieved documents before computing the score.

P@k, MAP, and bpref are measures based on binary relevance judgements. If relevance judgements are instead graded, binarization (in which the relevance levels are collapsed into binary judgements by considering only one or more of the highest relevance scores as relevant) must be applied to the relevance judgements before the calculation.

2.2 The NTCIR MathIR benchmark

Evaluation of math-aware search engines requires not only large corpora of mathematical documents but also sets of real-world, interesting queries with appropriate evaluation of the results [16]. Such an evaluation benchmark was lacking for the research community until 2010, when the first MathIR-focused task was created at NTCIR-10 (the 10th NII Testbeds and Community for Information access Research) [2]. NTCIR MathIR tasks were held in subsequent years in NTCIR-11 and NTCIR-12 ([3, 60]), serving to introduce an IR evaluation framework to math-aware search.

Several subtasks were developed in NTCIR, summarized in Tables 2 and 3 according to the types of queries and the target corpus in search. Each query consists of one or more formulas with or without keywords, and wildcard operators might occur in the query formulas. The target corpora to be searched are mathematical documents from arXiv[4] and Wikipedia[5]: the arXiv corpus contains paragraphs from technical articles with arXiv categories `math`, `cs`, `physics:math-ph`, `stat`, `physics:hep-th`, and `physics:nlin`; and the Wikipedia corpus contains complete articles from English Wikipedia that explicitly contain formulas, plus articles sampled from across the rest of Wikipedia. The two corpora are chosen to simulate the search needs for two groups of people: technical experts presumed to have a high level of mathematical sophistication (arXiv) and non-experts (Wikipedia).

Evaluation of the NTCIR MathIR tasks was pooling-based: the top-20 ranked documents were selected from each run and evaluated by human assessors with a graded judgment 0 (not-relevant), 1 (partially relevant), or 2 (relevant). In most NTCIR MathIR tasks, the primary measure was P@k for $k = \{5, 10, 15, 20\}$ for each of the three types of relevant hits. The relevance judgements along with the queries provided in the NTCIR MathIR tasks have supported the development of math-aware search engines over the years before the availability of the new benchmark in 2020, the ARQMath Lab series (Section 2.4).

[4] https://kwarc.info/projects/arXMLiv/
[5] http://www.cs.rit.edu/~rlaz/NTCIR12_MathIR_WikiCorpus_v2.1.0.tar.bz2

NTCIR subtasks		*NTCIR-10*	*NTCIR-11*	*NTCIR-12*
MathIR tasks with ArXiv corpus	Formula Search	O		
	Formula + Keyword Search	O	O	O (arXiv Main Task)
	Formula + Keyword Search with "simto"			O
	Free-form query search	O		
MathIR tasks with Wikipedia corpus	Formula Search		O (Wikipedia Open Subtask)	O (MathWiki Formula)
	Formula + Keyword Search			O (MathWiki)
	Formula + Keyword Search with "simto"			
Math understanding subtask		O		

Table 2: Summary of NTCIR tasks by Aizawa and Kohlhase [1].

Task	*# of Queries*	*# of Queries with a Wild card*	*Includes Text Keywords*
NTCIR-11 Wikipedia Open Subtask	100	43	No
NTCIR-12 MathWiki Task	30	10	Yes
NTCIR-12 arXiv Main Task	29	25	Yes
NTCIR-12 MathWikiFormula Task	40	20	No

Table 3: Summary of NTCIR tasks by Fraser [14].

2.3 Math Community Question Answering (MathCQA)

Community Question Answering (CQA) sites, broadly speaking, refer to web-based services where people can ask questions online or share their knowledge by providing answers to questions asked by the rest of the community on those sites. The process of people asking questions on CQA sites might be initiated by unsuccessful keyword searches using web search engines. They thus seek help from the CQA sites where they might input a more precise question description that is formulated in a natural language.

As a research field, CQA might refer to a large family of tasks characterized by the presence of three domain entities: a question, an answer, and a pair of users who act as an asker and an answerer. For instance, CQA might involve the tasks:

Question Retrieval: retrieving relevant old questions from the sites given a new question;

Answer Retrieval: retrieving relevant old answers from the sites given a new question;

User Ranking: estimating user expertise to rank a user;

or tasks such as topic classification, best answer prediction, answer ranking, asker satisfaction prediction, and so on. Srba and Bielikova have written a survey with a comprehensive classification of the tasks [56]. Usually, a CQA task refers to the process of question answering, that is, as a task of answer retrieval followed by answer ranking—and thus an IR task essentially that might be accomplished with text retrieval modelling. Additionally, the sophisticated structure in CQA allows many features to be extracted for modelling the task, as shown in Figure 2.

More recent CQA challenges were held at SemEval-2015, SemEval-2016, and SemEval-2017 (Task 3) [36, 37, 35] with data provided from the Qatar Living Forum[6] and the Fatwa site[7] (in Arabic). Domain-specific CQA challenges, on the other hand, can be diverse depending on the availability of such sites in real life. For mathematics—which is the focus of this research work—MathCQA sites are available, including the popular Math StackExchange[8] and Math Overflow[9], and the first MathCQA challenge was held using data from the former and to be introduced in the next section.

[6] https://www.qatarliving.com/forum/
[7] https://www.islamweb.net/ar/fatwa/
[8] https://math.stackexchange.com
[9] https://mathoverflow.net

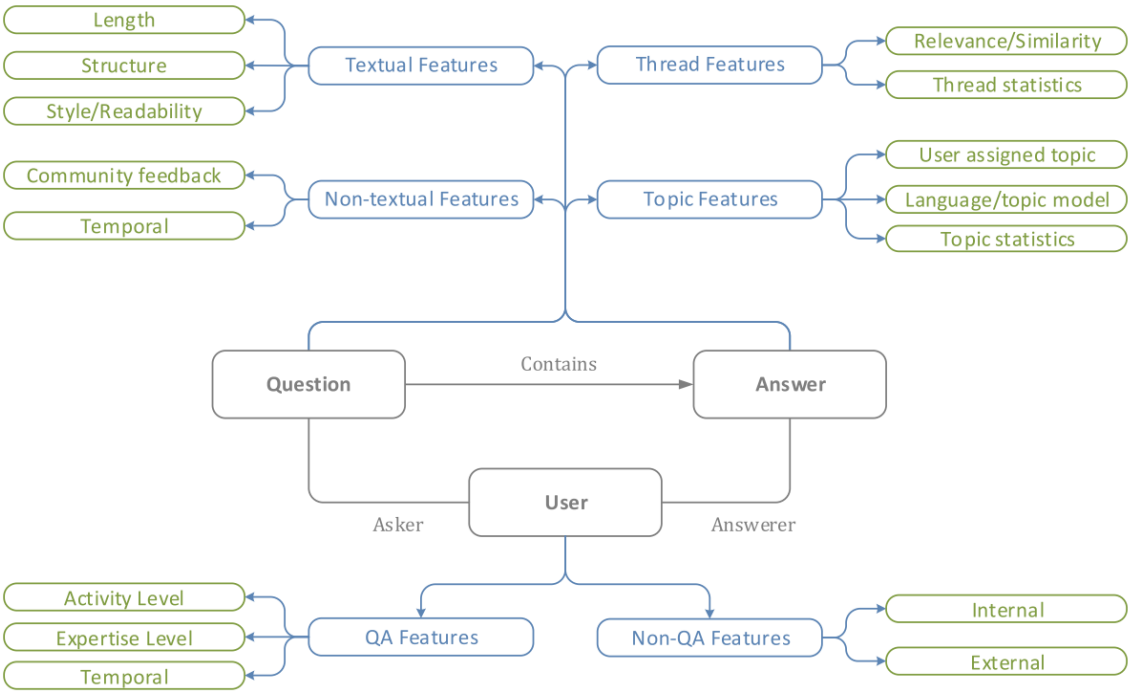


Figure 2: CQA Features categorized by Srba and Bielikova [56].

It is important to recognize that MathCQA differs from the research field of Math Question Answering (MathQA). While the former usually focuses on retrieving relevant answers from an existing dataset (a CQA site), the latter deals with actual problem-solving, that is, to compute a numeric answer from equations or to parse and solve an algebra problem with symbolic computing, for example. A recent MathQA task was held at SemEval-2019 [17], which considered a math question set that was derived from Math SAT practice exams. That task's objective is to identify one uniquely correct answer by multiple-choice selection or by numerical computation, involving no direct retrieval subtask. These types of MathQA tasks are thus not the focus of this research. More differences between CQA and QA in a general domain are discussed in a survey paper by Patra [44]. Olvera-Lobo, María Dolores and Gutiérrez-Artacho have prepared a list of previous QA challenges [42].

2.4 The ARQMath Lab Series

The ARQMath Lab series [62, 30], whose name stands for Answer Retrieval for Questions on Math, is the core evaluation platform used in this research work. Running at the Conference and Labs of the Evaluation Forum (CLEF)[10] in 2020 and 2021, and to be held again in 2022, the ARQMath Lab series provides the first Community Question Answering (CQA) platform with questions involving math data (and thus the first MathCQA). The Lab uses a collection of questions and answers from the Math StackExhange (MSE)[11] site and poses two tasks—an answer retrieval task and a formula retrieval task—with an aim to advance math-aware search and the semantic analysis of mathematical notation and texts.

The following subsections further describe the dataset and the two tasks in this Lab series. Hereafter, the Lab series in 2020 and 2021 are referred to as *ARQMath-1* and *ARQMath-2* respectively, and the two tasks as *the MathCQA Task* and *In-Context Formula Retrieval*, respectively.

[10] http://www.clef-initiative.eu/
[11] https://math.stackexchange.com/

2.4.1 Dataset: The MSE Collection and Formula Files

The Math StackExchange Forum

Math StackExchange (MSE) is a CQA forum specialized in mathematics, where users can *post* math questions (thus create a question post) or provide an answer to a math question (thus an answer post). A question post and its associated answer posts together form a thread. A question post in the thread consists of a title that summarizes the question, a body text that includes all necessary information, and up to 5 *tags* that describe the question. An answer post contains a proposed answer to its associated question post in the same thread, and might receive *votes*—up-votes or down-votes—from other users. Users might also *edit* their own posts, leave *comments* to any posts, or mark an existing question post as a *duplicate* post or a *related* post of other question posts. *Badges* might be assigned to users to reflect their reputation according to their activeness in the forum.

The MSE Collection Format

The ARQMath Lab collection is a processed MSE snapshot as of 01-March-2020 from the Internet Archive[12]. With respect to the forum activities mentioned above, the collection is stored as separate XML[13] files: Posts, Tags, Votes, PostHistory, Comments, PostLinks (that records the duplicate posts and related posts), Badges and Users. The final MSE collection that is provided to the Lab participants accounts for data from 2010 to 2018 and contains roughly 1.1 million math question posts and 1.4 million answer posts.

Formula Annotations and Representation Files

Formulas present in the MSE collection (specifically, in the Posts file and the Comments file) are annotated by the Lab organizers. Each identified instance of a formula is assigned a unique formula ID, 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. While the raw formulas in the files are presented as LaTeX, the ARQMath Lab also provides other formula representations—formulas in Presentation

[12] https://archive.org/download/stackexchange
[13] https://www.w3.org/TR/1998/REC-xml-19980210.html

MathML and Content MathML (See Section 2.1.1) generated using LaTeXML[14] to facilitate participants' systems development. These formula representation files are stored as Tab Separated Value (TSV) files, where each line of a TSV file represents a single instance of a formula that includes a formula id (and a visual id, see below) and its formula representation in either LaTeX, Presentation MathML, or Content MathML. Roughly 28 million formulas are annotated accordingly.

Difference between ARQMath-1 and ARQMath-2

ARQMath-1 and ARQMath-2 provide the same MSE collection, but with different versions of the formula representation files as follows:

- In the ARQMath-1 version, a total of 8% of Presentation MathML and 10% of Content MathML are missing in the formula representation files due to conversion failures of some malformed LaTeX formulas and the processing limitation of LaTeXML. Improvement were made in ARQMath-2 by the Lab organizers so that only 0.14% of both the Presentation MathML and Content MathML are missing in the files due to conversion failures.
- ARQMath-2 introduces, the concept of *visually distinct formula* (see Section 2.4.3). In ARQMath-2, around 9.3 million visually distinct formulas are recognized by the Lab organizers, and a *visual id* is included for each formula instance in the ARQMath-2 formula representation files. Repeated occurrences of a visually identical formula would have different formula ids but the same visual id in these formula representation files.

ARQMath-1 provides only sample data but not any training data for the tasks, due to it being the first of its series. Starting from ARQMath-2, researchers might use the relevance assessment released in ARQMath-1, which is referred hereafter as the *ARQMath-1 benchmark*, as their training data. Similarly, the ARQMath-2 relevance assessment creates an *ARQMath-2 benchmark*.

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

Topic-ID: A.75

Title: Prove that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$

Body: I'm unsure how to show that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$. Looking at the solutions it starts with $e^u > \frac{u^{m+1}}{(m+1)!}$ but not sure how this is a logical step.

Tags: real-analysis, calculus, limits

Figure 3: A math question, framed as a *topic*, with topic-ID, title, body, and tags.

2.4.2 Task 1: The MathCQA Task

Task Definition

Task 1 of the Lab, *the MathCQA Task*, is defined as follows:

> Given a posted question as a query, search all answer posts and return relevant answer posts.[15]

The query questions, represented as *topics*, are selected from question posts from the part of MSE collection from 2019 (for ARQMath-1) and 2020 (for ARQMath-2), which are not accessible by participants. Each topic includes a topic-ID, title, body text, and list of tags, as shown in Figure 3.

Participants need to submit a *run* which includes a ranked list of at most 1,000 answer posts—represented by their post ids, referred as *answer ids*—for each topic retrieved from the provided MSE collection. Each participant might submit one *primary* run—the major run to be assessed—and up to four *alternate* runs. The participants declare whether each submitted run is an *automatic* run, meaning the result is produced without human intervention, or a *manual* run, meaning that there is some human involvement during the generation of the ranked lists.

Relevance Assessment

During the assessment, top-k pooling is performed on all participants' submitted runs to create pools of answer posts to be judged for relevance to each topic. For ARQMath-1, the

[15] https://www.cs.rit.edu/~dprl/ARQMath/Task1-answers.html

top-50 results in all primary runs and baseline runs, and the top-20 results for all alternate runs are judged to be of high (H), medium (M), or low (L) relevance or to be not relevant (NR). For ARQMath-2, the top-45 results and the top-15 results are selected, respectively.

The primary evaluation metric for the task is nDCG′. The Lab organizers also compute other common IR metrics: P′@10 and MAP′, with H+M binarization, meaning that all answer posts with H or M judgements are deemed to be relevant and those with L or NR judgements are deemed to be irrelevant (Section 2.1.3). For ARQMath-2, all submitted runs are run against the ARQMath-1 benchmark as an alternative to compare their performance.

Baseline Systems

The Lab organizers provide five baselines systems in ARQMath-1 and four baseline systems in ARQMath-2 for Task 1. One of the baselines, *Linked MSE posts*, uses privately-held data, which is not available to Lab participants, as described as follows:

Linked MSE posts: a model "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." [62]

Other baseline systems are: *TF-IDF*, *Tangent-S*, *TF-IDF + Tangent-S*, and *Approach0*. The *Approach0* system is a baseline system in ARQMath-1 only; in ARQMath-2 it is one of the participant systems. These systems are further described in Section 2.5.1 and 2.5.2.

Results and Released Benchmark

A summary of the task participation and task topics for ARQMath-1 and ARQMath-2 can be found in Table 4, and the official results are attached in Appendix A.1 and A.2.

In ARQMath-1, runs from the MathDowsers team placed in the top three positions out of all runs (including the baselines) with respect to the primary measure nDCG′. For P′@10 and MAP′, the top run was achieved by the baseline run *Linked MSE posts*. In ARQMath-2, one of the runs from the MathDowsers team placed first out of all runs (including the baselines) with respect to the primary measure nDCG′ as well as another

Count of	Task 1: The MathCQA Task *ARQMath-1*	*ARQMath-2*
Total Participant Teams	5	9
Total Participant Runs	18	36
Total Runs with Baselines	23	40
MathDowsers' Runs	5	2
Total Topics	98	100
Assessed Topics	77	71
Extra Assessed Topics	0	18

Table 4: Task summary for the MathCQA task. Extra assessed topics are released after the Lab period and do not contribute to the performance of the submitted runs.

Count of Topic Labels	Task 1: The MathCQA Task *ARQMath-1 (77 Topics)*	*ARQMath-2 (71 topics)*
Dependency		
Text	13	10
Formula	32	21
Both	32	40
Topic Type		
Computation	26	25
Concept	10	19
Proof	41	27
Difficulty		
Easy	32	32
Medium	21	20
Hard	24	19

Table 5: Labels for the assessed topics in the MathCQA task.

measure MAP′. Concerning the remaining measure P′@10, the top run was again achieved by the baseline run *Linked MSE posts*.

For both ARQMath-1 and ARQMath-2, assessed topics are further labelled by the Lab organizers as depicted in Table 5.

2.4.3 Task 2: In-Context Formula Retrieval

Task Definition

Task 2 of the Lab, *In-Context Formula Retrieval*, is defined as follows:

> Given a question post with an identified formula as a query, search all question and answer posts and return relevant formulas with their posts.[16]

The *topics* in this task are identical to the topics defined in Task 1, with the addition of one identified formula selected from the title or the body text of the topic.

Unlike a regular formula retrieval task, of which the relevance of a retrieved formula is determined in isolation, in this *in-context* formula retrieval, the relevance of a formula is defined by its *expected utility* given its associated question. More specifically, the relevance judgement task for the Lab assessors is defined 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? [30]

Participants need to submit a *run* which includes a ranked list of at most 1,000 formula instances—represented by *formula ids*—along with the question or answer post that they appear in (again, represented by their post ids) from the provided MSE collection. Similar to Task 1, each participant might submit one *primary* run——the major run to be assessed——and up to four *alternate* runs. The participants also declare whether each submitted run is an *automatic* run, meaning the result is produced without human intervention, or a *manual* run, meaning that there is some human involvement during the generation of the ranked lists.

[16] https://www.cs.rit.edu/∼dprl/ARQMath/task2-formulas.html

Visually Distinct Formulas

While retrieved items in the runs are formula instances, *visually distinct formulas* are considered during assessment. Visually distinct formulas refer to formulas distinguishable by their appearance. They are determined by the Lab organizers through clustering formula instances using their Presentation MathML representations when possible, and otherwise the LaTeX representations. Around 9.3 million visually distinct formulas partition the 28 million formula instances in the MSE collection.

Relevance Assessment

During the assessment, simple top-k pooling is not performed on the retrieved formula instances, but instead all formula instances in a submitted run are first clustered by visually distinct formulas. The process is to proceed down each list of formula instances until some threshold of visually distinct formulas has been seen. In ARQMath-1, the pool depth is the rank of the first instance of the 25th visually distinct formula for primary runs and for the baseline run; for alternate runs, the pool depth is the rank of the first instance of the 10th visually distinct formula. In ARQMath-2, it is 20th and 10th, respectively.[17]

After pooling, assessment is then performed on formula instances: for each visually distinct formula, at most five instances are judged by the Lab assessors. In ARQMath-1, the five instances that contribute to the pools by the largest number of runs are selected, breaking ties randomly. In ARQMath-2, the five instances are chosen by a voting protocol to prefer highly-ranked instances in addition to instances returned in multiple runs: each instance vote is weighted by the sum of its reciprocal ranks within each run, breaking ties randomly.

Assessors are presented with the query formula within its associated question post, and also the retrieved formula instances *in context* with its associated post. Each formula instance is graded according to the relevance judgement task defined previously from 0 (not expected to be useful) to 3 (just as good as finding an exact match to the query formula would be).

The evaluation metric for Task 2 follows Task 1, where the primary metric is nDCG′ and the other two metrics, P′@10 and MAP′, are provided as well. All submitted runs for ARQMath-2 are also run against the ARQMath-1 benchmark as an alternative means to compare their performance.

[17] Additionally, a pool depth of up to 1,000 is used in ARQMath-1 for any formula having its associated answer marked as relevant in Task 1; but this is not used in ARQMath-2.

Count of	Task 2: In-Context Formula Retrieval *ARQMath-1*	*ARQMath-2*
Total Participant Teams	4	6
Total Participant Runs	11	17
Total Runs with Baselines	12	18
MathDowsers' Runs	0	2
Total Topics	87	100
Accessed Topics	45	58
Extra Accessed Topics	27	12

Table 6: Task summary for In-Context Formula Retrieval. Extra accessed topics are released after the Lab period and do not contribute to the performance of the submitted runs.

Baseline Systems

The Lab organizers provide a baseline system, *Tangent-S* for Task 2. This system is further described in Section 2.5.1.

Result and Released Benchmark

A summary of the task participation and task topics for ARQMath-1 and ARQMath-2 can be found in Table 4, and the official results are attached in Appendices A.3.1 and A.4.

In ARQMath-1, no systems had a better result than the baseline system *Tangent-S* with respect to the primary measure nDCG′ and also MAP′. Concerning P′@10, a run from the DPRL team placed first among all teams. The MathDowsers team did not participate for the task during ARQMath-1. In ARQMath-2, a manual run from the *Approach0* team (Section 2.5.2) placed first out of all runs (including the baseline) with respect to all three measures. In regards to the primary measure nDCG′, an automatic run from the MathDowsers team achieved an indistinguishable result to the manual run from the *Approach0* team, placing first out of all 13 automatic runs.

For both ARQMath-1 and ARQMath-2, assessed topics are further labelled by the Lab organizers with formula complexity as low, medium, or high, and in ARQMath-1 the topics are also labelled with the major math element (such as limit, integral, fraction, etc.).

Difference in Assessment between ARQMath-1 and ARQMath-2

In ARQMath-2, the formula representation files are updated with a better quality of formula representations (Section 2.4.1), in particular, the Presentation MathML representations that are used to cluster the visually distinct formulas. Also, the pooling for ARQMath-2 is based on visually distinct formulas clustering from formula instances among the whole collection, instead of clustering among only the submitted runs as in ARQMath-1. During ARQMath-2, the Lab organizers re-evaluate—as an unofficial result—those runs that were submitted back in ARQMath-1, with the retrieved formula instances clustered to visually distinct formulas using the updated formula representation files. This results in a change of the evaluation measures to most ARQMath-1 participants' runs (Appendix A.3.2). In particular, P′@10 changes for several different runs, and the baseline system *Tangent-S* gets the best result for P′@10 that was previously reported to be exceeded by one of the participant runs.

On the other hand, in ARQMath-2, the relevance assessment of the formula instances must be done based on the context in which they appear, but in ARQMath-1, this is not strictly enforced. For example, in ARQMath-2, there is one formula query $x^n + y^n + z^n$ (topic-id B.289) with its associated question post stating that x, y, and z could be any real numbers. The assessors consider as irrelevant all exact matches in the pooled posts where x, y, and z are explicitly referred to as integers instead of real numbers. In contrast, in ARQMath-1, the assessors are instructed that if the query and candidate formulas have the same appearance, then the candidate is highly relevant.

These differences suggest that while the ARQMath-1 data is available for Task 2, it should be considered with caution; the ARQMath-2 data might be a better choice as training data and benchmark data for Task 2.

2.5 Math-aware Search Engines at ARQMath-2

This section provides a short description of the baselines and participant systems during ARQMath-2 (except for MathDowsers, which is the subject of Chapter 3). A brief table outlining each system's major techniques used (Section 2.1.2), and an annotation of their use of formula representations (Section 2.1.1) is displayed in Figure 4.

2.5.1 TF-IDF and Tangent-S

These serve as baselines [30], with each system either:

- a Bag-of-Words model built on LaTeX and text tokens, scored by TF-IDF (via an IR platform Terrier [43]);
- a three-stage formula retrieval system (Tangent-S [12]) that first retrieves top formula candidates using symbol pairs built from SLT and OPT, followed by subtree-structure matching, and finally ranking of formulas by a linear regression model trained with internal features;
- or a linear combination of both.

2.5.2 Approach0

This is a combined system [65] consisting of two search engines:

- Approach Zero [64, 63], which is a tree-structure search system built on leaf-to-root paths extracted from OPT, with a combined subtree similarity and symbol similarity scoring on formula terms. It also indexes text terms, in which case the scoring function is BM25$^+$;
- a Bag-of-Words model built on leaf-to-root paths, scored by BM25 (via the IR toolkit Anserini [59]).

Each submission result is either an interpolated scoring between the two systems, or a concatenation where the final ranking is first contributed by Approach Zero and then contributed by Anserini.

2.5.3 XY-PHOC-DPRL

This system submitted runs for for the in-context formula retrieval task only. It converts each formula first into its *SVG image*, then to bounding boxes followed by multiple symbol location embeddings according to the occurrence of symbols in those bounding boxes; followed by a scoring function using cosine similarity. [4, 5]

2.5.4 MIRMU and MSM

Several systems were behind a related set of submissions for the MathCQA task only [41]:

- Several simple models built on LaTeX and text tokens with BM25-like and TF-IDF-like scoring functions from student course projects (the MSM-team submissions);
- A *soft* Vector Space Model that incorporates TF-IDF vectors, with term similarity adopted from a text-and-math joint embedding trained on MSE and ArXMLiv[18] by fastText. The model considers formulas with a tokenized prefix-notation of OPT (the SCM submission);
- a SentenceBERT model that trains with the MSE Question-Answer Pairs. The model considers the LaTeX encoding of formulas (the compuBERT submission);
- Ensembles of ten systems including the above systems, using different weighting schemes (the IBC, WIBC, and RBC submissions);

2.5.5 DPRL

Various systems [27] are submitted for the two tasks. For the MathCQA task, the answer ranking score is a multiplication of two similarity scores as follows:

- First a Question-Question similarity score, is computed by fine-tuning a pre-trained SentenceBERT model with MSE Question-Question pairs. The pre-training dataset is the Quora question pairs dataset[19].
- Then a Question-Answer similarity score is computed by three approaches:

[18] https://sigmathling.kwarc.info/resources/arxmliv-dataset-082019/
[19] https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs

- either fine-tune another pre-trained SentenceBERT model with MSE Question-Answer pairs, where the pre-training task is the MS Marco Passage Reranking task [40] (the QASim submission);
- or get the normalized MSE vote score (the MSE submission);
- or compute a combined score of the above by an adjusted Reciprocal Rank Function (adjusted RRF) as follows:

$$adjustedRRF(d \in D) = \sum_{m \in M} \frac{s_m(d)}{k + r_m(d)} \tag{2.9}$$

  Compared to the regular Reciprocal Rank Function (Section 2.1.2), this adjusted version incorporates also s_m which is the score of the document (the RRF submission).

All SentenceBERT models are applied to the LaTeX encoding of formulas.

For the in-context formula retrieval task, the submissions are:

- a Tangent-CFT2 system, which is an improved version of the Tangent-CFT formula retrieval system [29]. It trains formula embeddings by fastText with symbol pairs from the SLT and OPT representations, followed by ensembling with other internal features using the adjusted RRF at Equation 2.9 (the Tangent-CFT2 submission);
- a Tangent-CFT2TED formula retrieval system [28] which extends the above with tree-edit distance (the Tangent-CFT2-TED submission);
- and a combined result of the above, using Learning-to-Rank with partial or full ARQMath-1 benchmark data (the ltr29 and ltrall submissions).

2.5.6 TU_DBS

The submissions [50] use either:

- a system that first trains an ALBERT model with the MSE dataset, followed by training a classifier with MSE Question-Answer pairs or in-context MSE Formula-Answer pairs;
- or a system that trains a ColBERT model with MSE Question-Answer pairs. The ColBERT model is on top of a pre-trained SciBERT model [6] (the T_DBS_A4 submission only).

All systems use the LaTeX encoding of formulas.

2.5.7 NLP_NITS

The system [11] is a formula retrieval system and thus applicable for the in-context formula retrieval task only. It first trains BERT on the MSE dataset, then uses BERT embeddings to represent formulas and uses cosine similarity for formula matching. The system uses the LaTeX encoding of formulas.

2.5.8 PSU

This system [54] is only for the MathCQA task. It first ensembles two results by the Reciprocal Rank Function, one from the BM25 scoring and another from the TF-IDF cosine similarity scoring. The result is then re-ranked by a RoBERTa model that has been trained on the MSE dataset. Finally, the result is ensembled by the Reciprocal Rank Function with results from three other optimizations to produce the final ranking. The system also uses the LaTeX encoding of formulas.

	Tree-Structure Search	*Bag-of-Words Model*	*Vector Space Model*	*BERT-related*	*Ranking or Re-ranking*	*Ensembling*
TF-IDF (MathCQA)		TF-IDF (via Terrier) LaTeX				
Tangent-S (Formula Retrieval)	2. Top formula candidates ranked by MSS	1. Symbol Pair retrieval by Dice Coefficient for top formula candidates (via Lucene) SLT+OPT			3. Linear regression of features from (2)	
TF-IDF + Tangent-S (MathCQA)						Linearly combination of TF-IDF + Tangent-S
Approach0 (Both)	Ranking by subtree + symbol similarity (via Approach0) OPT	Ranking leaf-to-root paths by BM25 (via Anserini) OPT			(+ Linear regression / LambdaLM)	Interpolation or merging of Approach0 + Anserini
MathDowsers (Both)		**[primary]** Symbol-pair retrieval by BM25+ (via Lucene) SLT			**[proximityReRank]** + Proximity	
XY-PHOC-DPRL (Formula Retrieval)			XY- PHOC Symbol Location Embedding Image			
MSM (MathCQA)		**[MG]** BM25+ **[PZ]** TF-IDF (via Anserini) [...] LaTeX				
MIRMU (MathCQA)			**[SCM]** fastText embeddings + Soft Cosine Similarity OPT-Prefix	**[compuBERT]** SentenceBERT trained with QA pairs LaTeX		**[IBC]** Equal **[WIBC]** Weighted **[RBC]** Regression (10 systems)
DPRL (MathCQA)				1. SentenceBERT trained with Question pairs LaTeX	2 + **[QASim]** SentenceBERT trained with QA Pairs + **[MSE]** MSE answer scores	**[RRF]** Ensemble of QASim + MSE by RRF
DPRL (Formula Retrieval)			**[Tangent-CFT2]** fastText embeddings trained from Symbol Pairs SLT+OPT		**[Tangent-CFT2ED2]** + Tree Edit Distance **[ltr20, ltrall]** SVM-Rank with Tangent-S, Tangent-CFT2, Tangent-CFT2ED2	
TU_DBS (Both)				(AL)BERT Pretraining + ALBERT/ColBERT Classifier LaTeX		
NLP_NITS (Formula Retrieval)			BERT Pretraining + BERT embeddings LaTeX			
PSU (MathCQA)		1. BM25 + TF-IDF (Cosine Similarity) ensembled by RRF LaTeX		2. BPE + RoBERTa training and re-ranking		3. Ensemble of (2) and others by RRF

Figure 4: A summary of the baselines and the participant systems at ARQMath-2.

Chapter 3

Tangent-L: the Math-aware Search Engine

A math-aware search engine, when compared to a normal text search engine, requires specialized capabilities to address the following challenges:

1. How to handle input with formulas, in addition to natural language text?
2. How to combine formula retrieval and text retrieval to output a desired ranking result?

Proposed by Fraser et al., Tangent-L is a math-aware search engine built on the popular Lucene text search platform [14, 15]. Tangent-L tackles the above challenges by first adopting methods from its origin Tangent-3 [61] to create math features that represent input formulas; then combining formula retrieval and text retrieval as a whole with a traditional text retrieval ranking to achieve results that are comparable to using expensive math-specific scoring functions [15].

Evaluated using the NTCIR-12 benchmark, Tangent-L is shown to be competitive with the participating systems in that workshop [15]. Its math-aware capability is further consolidated when adapted to be the core component of the MathDowsers' participating system for the ARQMath Lab challenges [39, 38].

The remainder of this chapter describes Tangent-L in greater detail. The vanilla version, which is the initially proposed version in 2017, will be introduced first, followed by some system variants developed for the ARQMath Lab series in 2020 and 2021.

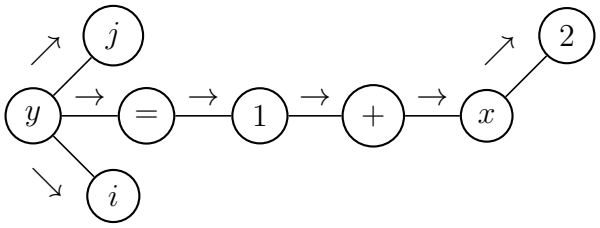


Figure 5: Symbol Layout Tree for $y_i^j = 1 + x^2$.

3.1 The Vanilla Version

3.1.1 From Formulas to Math Tokens

Adopted from Tangent-3, a key characteristic of Tangent-L is replacing formulas by a bag of math tokens in preparation for indexing and searching—just like the replacement of natural language text by text tokens in a text search engine.

Tangent-L takes as input a formula in Presentation MathML format (Section 2.1.1). At first, Tangent-L parses an input formula presentation into a Symbol Layout Tree (SLT), where nodes represent the math symbols, and edges represent the spatial relationship between these symbols (Figure 5). Thereafter, this tree-like representation is traversed to extract a set of features, or math tokens, to capture local characteristics of the *appearance* of a math formula. While Tangent-3 only considers one type of math tokens (*symbol pairs*) [61], Tangent-L recommends three additional types of math tokens (*terminal symbols*, *compound symbols* and *augmented locations*) to represent a formula as depicted in Table 7.

In preparation for indexing (or a search), the math tokens replace the formula itself in the document (or the query) and then they are considered by Tangent-L as if each were a keyword term in the text to be matched.

Feature Type	*Definition and Extracted Features*		
Symbol pairs	For each edge, start and end symbols and edge label		
	$(y, j, \nearrow)$	$(y, =, \rightarrow)$	$(y, i, \searrow)$
	$(=, 1, \rightarrow)$	$(1, +, \rightarrow)$	$(+, x, \rightarrow)$
	$(x, 2, \nearrow)$		
Terminal symbols	List of symbols with no outedges		
	$(j, \triangle)$	$(i, \triangle)$	$(2, \triangle)$
Compound symbols	List of outedge labels for nodes with more than one outedge		
	$(y, \nearrow\searrow\rightarrow)$		
Augmented locations	For each feature of the first three types, that feature together with the path to the feature's (first) symbol		
	$(y, j, \nearrow, \emptyset)$	$(y, =, \rightarrow, \emptyset)$	$(y, i, \searrow, \emptyset)$
	$(=, 1, \rightarrow, \rightarrow)$	$(1, +, \rightarrow, \rightarrow\rightarrow)$	$(+, x, \rightarrow, \rightarrow\rightarrow\rightarrow)$
	$(x, 2, \nearrow, \rightarrow\rightarrow\rightarrow\rightarrow)$	$(j, \triangle, \nearrow)$	$(i, \triangle, \searrow)$
	$(2, \triangle, \rightarrow\rightarrow\rightarrow\rightarrow\nearrow)$	$(y, \nearrow\searrow\rightarrow, \emptyset)$	

Table 7: Extracted features (math tokens) to represent the formula in Figure 5. Each token is a "tuple" that records local characteristic of a symbol layout tree representation.

3.1.2 Single Retrieval Model with BM25$^+$ Ranking

Another key characteristic of Tangent-L is its ranking methodology: creating a single retrieval model by indexing text tokens and math tokens altogether, followed by BM25$^+$ [24] for the final retrieval result.

This ranking methodology tightly follows a traditional text search approach. The origin of BM25$^+$ is Okapi BM25 [52], a commonly used scoring function in traditional text search (Section 2.1.2). BM25$^+$ [24], which is used by Tangent-L, further extends Okapi BM25 by setting a proper lower bound in the Term-Frequency component to handle deficiency in document-length normalization. More specifically, given a collection of documents D containing $|D|$ documents and a query q consisting of a set of query terms, the BM25$^+$ score for a document $d \in D$ is defined as the sum of scores for each query term as follows:

$$\text{BM25}^+(q, d) = \sum_{w \in q} \left(\frac{(k+1) tf_{w,d}}{k \left(1.0 - b + b \frac{|d|}{\bar{d}}\right) + tf_{w,d}} + \delta \right) \log \left(\frac{|D| + 1}{|D_w|} \right) \tag{3.1}$$

where

$$\log\left(\frac{|D|+1}{|D_w|}\right) \tag{3.2}$$

is the Inverse-Document-Frequency component with $|D_w|$ being the document frequency for w—the number of documents in D containing term w; and

$$\frac{(k+1)tf_{w,d}}{k\left(1.0-b+b\frac{|d|}{\overline{d}}\right)+tf_{w,d}}+\delta \tag{3.3}$$

is the Term-Frequency component with $tf_{w,d}$ being the term frequency of w in document d; $|d|$ being the document length; $\overline{d}$ being the average document length; and k, b, and δ are constants detailed below:

- the constant k models term frequency saturation inspired by the function $\frac{tf_{w,d}}{k\ +\ tf_{w,d}}$, such that the reward for term frequency is limited when the term frequency grows very large;
- the constant b addresses document length normalization by applying the multiplier $\left(1.0-b+b\frac{|d|}{\overline{d}}\right)$ on k in the denominator of the term frequency saturation function; and
- the constant δ sets a lower bound for the whole Term-Frequency component, such that the score between the presence of a query term in a long document against the absence of a query term in a short document is properly distinguished.

In general, the constants k, b, and δ are chosen to be 1.2, 0.75, and 1, respectively, without specific data training.

Tangent-L adopts the BM25$^+$ ranking with slight variations. For a bag of query terms with term repetition, the score for the repeated query term is accumulated multiple times. Also, Tangent-L allows for math tokens to be given a weight that differs from the keyword tokens by the following equation:

$$\mathrm{BM25}^+_{\mathrm{w}}(q_t \cup q_m, d) = \mathrm{BM25}^+(q_t, d) + \alpha \cdot \mathrm{BM25}^+(q_m, d) \tag{3.4}$$

where q_t is the set of keyword tokens in a query, q_m is the set of math tokens in that query, and α is a parameter to adjust the relative weight applied to math tokens. In the NTCIR-12 benchmark, a gain in precision is observed as α increases in the range $0.05 \leq \alpha \leq 0.50$, followed by a gradual decline for larger values of α [15].

Overall, this vanilla version of Tangent-L—which follows a traditional text retrieval approach with appropriately chosen math tokens—is simple yet effective enough to have a comparable performance with other systems in the NTCIR-12 benchmark. Starting from the next section, variations of Tangent-L developed for the ARQMath Lab series are introduced.

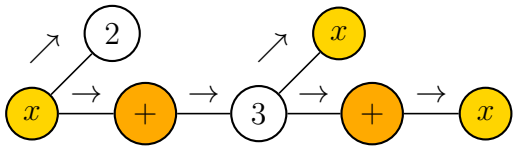


Figure 6: Symbol Layout Tree for $x^2 + 3^x + x$ with repetitions highlighted.

Feature Type	*Tokens Generated*		*Remark*
Repeated symbols	$\{x, \rightarrow\rightarrow\nearrow\}$	$\{x, \rightarrow\rightarrow\rightarrow\rightarrow\}$	The first occurrence of x resides on the same path as each of the second and third occurrences.
	$\{+, \rightarrow\rightarrow\}$		Similarly for $+$.
	$\{x, \nearrow, \rightarrow\rightarrow\}$		The second and third occurrences of x lie on difference root-to-leaf paths and share a closest common ancestor.
Augmented locations	$\{x, \rightarrow\rightarrow\nearrow, \emptyset\}$	$\{x, \rightarrow\rightarrow\rightarrow\rightarrow, \emptyset\}$	Augmented with the path from the root to the first occurrence.
	$\{+, \rightarrow\rightarrow, \rightarrow\}$		Similarly for $+$.
	$\{x, \nearrow, \rightarrow\rightarrow, \rightarrow\rightarrow\}$		Augmented with the path from the root to the closest common ancestor.

Table 8: Generated repetition tokens for the formula in Figure 6.

3.2 Incorporating Repeated Symbols

3.2.1 From Repeated Symbols to Repetition Tokens

Repetitions of symbols are commonplace in a formula; for instance, x repeats in the formula $x^2 + 3^x + x$, as does the operator $+$ (Figure 6). Ideally, a search for either $y^x - x$ or $6x^3 - y + x$ could match that formula because of the pattern of repetitions for x, and a search for $2y^3 + y + 5$ could also match because of the repeated symbol $+$.

With this motivation, a variant of Tangent-L is developed with a new type of math features—*repetition tokens*—to capture this characteristic. Repetition tokens are generated based on the relative positions of the repeated symbols in a formula's SLT representation.

For every pair of repeated symbols:

1. if the pair of repeated symbols reside on the same root-to-leaf path of the SLT (that is, one is an ancestor of the other), then a repetition token $\{symbol, p\}$ is generated, where p represents the path between the repeated symbols;

2. otherwise, a repetition token $\{symbol, p_1, p_2\}$ is generated where p_1 and p_2 represent the paths from the closest common ancestor in the SLT to each repeated symbol.

If a symbol repeats k times where $k > 1$, $\binom{k}{2}$ repetition tokens are generated for that symbol following the above procedure. For each of these tokens, an additional location token is generated with the augmentation of the path traversing from the root to the closest common ancestor of the pair. As such, a total of $2 \cdot \binom{k}{2}$ repetition tokens are generated and indexed. Table 8 shows the repetition tokens that would be indexed for the formula $x^2 + 3^x + x$ in Figure 6.

Notice that this choice of encoding for the repetition tokens contains ambiguity. For instance, the repetition token $\{x, \nearrow, \rightarrow\rightarrow\}$ might represent either a repeated symbol in $3^x + x$, or an augmented location tuple in $1 + y + x^x$. However, in practice this ambiguity might not have a significant effect on performance.

3.2.2 Revised Ranking Formula

With the introduction of repetition tokens, this variant of Tangent-L generates three token types: text tokens, regular math tokens, and repetition tokens. Similar to the vanilla version, this variant also applies BM25$^+$ ranking to the query terms and the document terms during the search. The revised ranking formula with the repetition tokens is as follows:

Let q_t be the set of text tokens, q_m be the set of regular math tokens, and q_r be the set of repetition tokens generated for the query terms. Let d be a document represented by the set of all its indexed tokens. The score is then defined as

$$\begin{aligned} &\mathrm{BM25}^+_{\mathrm{w}}(q_t \cup q_m \cup q_r, d) = \\ &\qquad \alpha \cdot \big(\gamma \cdot \mathrm{BM25}^+(q_r, d) + (1-\gamma) \cdot \mathrm{BM25}^+(q_m, d)\big) + (1-\alpha) \cdot \mathrm{BM25}^+(q_t, d) \end{aligned} \tag{3.5}$$

where the component

$$\gamma \cdot \text{BM25}^+(q_r, d) + (1 - \gamma) \cdot \text{BM25}^+(q_m, d) \tag{3.6}$$

is the score for math features with the γ parameter being used to balance the weight of repetition tokens against that of regular math tokens. The weight of math features against that of keyword features is further balanced by the α parameter.

Unlike the vanilla version of Tangent-L, where the parameter α is unbounded (Equation 3.4), in this variant $0 \leq \alpha, \gamma \leq 1$ which makes the ranking function a convex combination. It is, however, easy to relate previously-studied parameter settings in the vanilla version to that for this variant: simply put γ to zero, followed by setting α to a proper scaling[1] of the α value from the previous setting. The parameters can be further tuned based on the target dataset.

[1]Let x be the value of α in Equation 3.4 of the vanilla version. Then the corresponding value of α to be set in Equation 3.5 is $\frac{x}{1+x}$.

3.3 Formula Normalization

3.3.1 Five Classes of Semantic Matches

Mathematical expressions can be rewritten in numerous ways without altering their meaning. For example, $A + B$ matches $B + A$ semantically because of the commutative law. To accommodate such variability and increase recall, this variant of Tangent-L is equipped with formula normalization, that is, the ability to generate similar math features for two formulas with the same semantics.

The following five classes of semantic matches are considered:

1. **Commutativity:** $A + B$ should match $B + A$
2. **Symmetry:** $A = B$ should match $B = A$
3. **Alternative Notation:** $A \times B$ should match $A\ B$, and $A \not> B$ should match $A \leq B$
4. **Operator Unification:** $A \prec B$ should match $A < B$
5. **Inequality Equivalence:** $A \geq B$ should match $B \leq A$

and simple adjustments are applied to Tangent L's regular math tokens to support these semantic matches.

The adjustment to handle the first two classes, *Commutativity* and *Symmetry*, are similar. In the vanilla version, Tangent-L generates a math token for each pair of adjacent symbols with their orders preserved (the feature type *symbol pairs* in Table 7). For example, two math tokens (A, $+$, $\rightarrow$) and ($+$, B, $\rightarrow$) are generated for the expression $A+B$, and two different math tokens (B, $+$, $\rightarrow$) and ($+$, A, $\rightarrow$) are generated for the expression $B + A$. In order for an exact match to take place for the two expressions, a simple adjustment to the math tokens is to ignore the order of a pair of adjacent symbols whenever commutative operators or symmetric relations are involved. With this approach, both expressions $A+B$ and $B + A$ generate the same pair of math tokens, ($+$, A, $\rightarrow$) and ($+$, B, $\rightarrow$), so that an exact match is made possible.

The next two classes, *Alternative Notation* and *Operator Unification*, can be easily accommodated by choosing a canonical symbol for each equivalence class of operators and consistently using only the canonical symbols in any math tokens generated as features.

The final class, *Inequality Equivalence*, can be handled by choosing a canonical symbol (for instance, choosing the symbol "$\leq$" in preference to "$\geq$") and then reversing the operands whenever necessary during math tokens generation.

3.3.2 Limitation

The proposed simple implementation for the three semantic classes *Commutativity*, *Symmetry* and *Inequality Equivalence* suffers from the fact that each math token encodes a local characteristic of the appearance of a formula—which handles only a pair of adjacent symbols at a time. As such, formula normalization by changing the order of adjacent symbols might result in a semantically distinct formula.

For example, with a longer expression such as $A + B \times 5$, the overly simplistic approach will generate the same set of math tokens as the expression $B + A \times 5$, failing to consider the priority of math operators where multiplications should precede addition. The same drawback can be observed with other commutative operations, symmetric relations, and inequality equivalence.

Nevertheless, overcoming this difficulty is no easy task since the input of Tangent-L—Presentation MathML—captures the layout of the symbols only without semantic meaning. To accommodate the pitfalls, this variant of Tangent-L provides a separate flag to control whether or not a semantic class is to be supported, so that only those deemed to be advantageous are applied when math tokens are generated.

Rank	*Retrieved Formula*	*Normalized Formula Similarity*
1	$6^{2(k+1)+1}+1$	1.000
2	$7^{2(k+1)+1}+1$	1.000
3	$a^{2(k+1)+1}+b^{2(k+1)+1}$	0.967
4	$3^{2(k+1)+1}-3$	0.855
5	$7^{2(k+1)+1}+1=7^2(8m)$	0.824

Table 9: Top-5 similar formulas for $a^{2(k+1)+1}+1$ using a formula retrieval model built with ARQMath-2 data (Section 2.4.1).

3.4 Holistic Formula Search

Formula matching within Tangent-L is based on comparing a set of math tokens from the query to those from each document (Section 3.1.2). If a document has multiple formulas, math tokens generated from all formulas within the document are then considered as a single unordered bag of terms without distinction of each separate formula.

This variant of Tangent-L proposes a solution that matches each formula as a whole within a document, instead of matching math tokens irrespective of formulas that might scatter across a document. Unlike the vanilla version of Tangent-L, this variant is essentially a two-stage retrieval model: during a search, each formula in the query is first replaced by the top-κ similar formulas existing in the dataset through the help of a formula retrieval model; then the new query is executed against the target document corpus where formulas are indexed holistically. This two-stage retrieval model is explained in greater detail in the following subsections.

3.4.1 Formula Retrieval with a Formula Corpus

At preparation time, a formula corpus is first pre-built with Tangent-L that indexes all visually distinct formulas in the target dataset, each as a separate document with a distinct *visual-id* serving as a key. In this formula corpus, each formula is indexed with the math features following the regular practice of Tangent-L, with the only difference being that each indexed document contains only a bag of math features from one single formula. A formula retrieval model is then defined as follows:

Let f_q be an arbitrary formula used as a query, F be the set of formulas in the formula corpus, and $f \in F$. Let q_m be the set of regular math tokens generated for the query terms

and let q_r be the set of repetition tokens, assuming its presence (adopting the Tangent-L variant from Section 3.2.1). The ranking function within the formula corpus is

$$\text{FormulaScore}(f_q, f) = (1 - \gamma) \cdot \text{BM25}^+(q_m, f) + \gamma \cdot \text{BM25}^+(q_r, f) \tag{3.7}$$

where the γ parameter is used to balance the weight of repetition tokens against that of regular math tokens with $0 \leq \gamma \leq 1$ (Equation 3.5). A corresponding formula similarity score, *Normalized Formula Similarity*, can then be defined as a *normalized* score of FormulaScore:

$$N(f, f_q) = \frac{\text{FormulaScore}(f_q, f)}{\max_{\varphi \in F} \text{FormulaScore}(f_q, \varphi)} \tag{3.8}$$

with $0 \leq N(f, f_q) \leq 1$ representing how well the query formula f_q is matched *in appearance* by f relative to other formulas within the formula corpus (Table 9).

During a keyword-and-formula search against the target document corpus, each query formula is first replaced by the top-κ similar formulas retrieved from the formula corpus through this formula retrieval model. The search is then performed against the target document corpus as described next.

3.4.2 Retrieval with Holistic Formulas

When indexing the target document corpus, instead of replacing each formula within the document with the set of math tokens generated for that formula, each formula is represented by a single *holistic formula token*. A holistic formula token is a token uniquely mapped to a visually distinct formula, implemented as a specific string embedded with the corresponding visual-id. Each formula is thus represented as a whole within a document.

During a keyword-and-formula search and after each query formula has been replaced by its top-κ retrieved formulas from the formula corpus, those top-κ retrieved formulas in queries are further replaced by their corresponding holistic formula tokens. The new query then consists of text tokens and holistic formula tokens which can be matched against the target document corpus. The final keyword-and-formula retrieval model is defined as follows:

Let q_t be the set of keyword tokens, q_f be the set of query formulas and d be a document represented by the set of all its indexed tokens. The ranking function is

$$\text{BM25}^+_{\text{w}}(q_t \cup q_f, d) = (1 - \alpha) \cdot \text{BM25}^+(q_t, d) + \alpha \cdot \text{BM25}^+(q_f, d) \tag{3.9}$$

where the α parameter is again used to balance the weight of formula terms against that of keyword tokens.

The component $\text{BM25}^+(q_f, d)$ that scores the formula terms takes a variant of the original BM25^+ as follows: let $f_q \in q_f$ be a query formula and let $S_\kappa(f_q)$ be the set of κ holistic formula tokens that replaces f_q in the original query. Let $N(f, f_q)$ be the Normalized Formula Similarity defined in Equation 3.8. Then

$$\text{BM25}^+(q_f, d) = \sum_{f_q \in q_f} \sum_{f \in (d \cap S_\kappa(f_q))} N(f, f_q) \cdot \left(\frac{(k+1)tf_f}{k\left(1.0 - b + b\frac{|d|}{\bar{d}}\right) + tf_f} + \delta \right) \log\left(\frac{|D|+1}{|D_f|}\right) \tag{3.10}$$

This score is a sum of weighted scores for all replacement formulas, with each individual score having a Term-Frequency component and an Inverse-Term-Frequency component just like the original BM25^+ (Equation 3.1) but further weighted by the Normalized Formula Similarity.

When compared to the vanilla version of Tangent-L, this retrieval model isolates the formula retrieval stage to achieve the goal of matching formulas holistically. The drawback, though, is that it is also computationally more expensive because of this extra retrieval stage.

Chapter 4

Addressing the MathCQA Task

Generally, math-aware search engines from the MathIR research community are designed to serve a *generic* information need: finding math documents relevant to a given query, where a query typically consists of a bag of keywords and formulas. In contrast, MathCQA poses a *specific* real-life challenge: finding potential answers from a CQA forum to a given math question, where the math question is expressed in mathematical natural language. Adaptations optimally tuned for the challenge are necessary for the math-aware search engine to fulfill its full potential.

As such, the following three-stage methodology is proposed for a math-aware search engine to adapt to a MathCQA challenge:

Stage 1: Query Conversion: transform a given math question into a well-formulated query consisting of a bag of keywords and formulas, as an input to the math-aware search engine.

Stage 2: Math-aware Retrieval: use the math-aware search engine to build an indexed corpus from the task collection and execute the formal query to find best matches.

Stage 3: Answer Ranking: from the retrieved matches, produce a ranked list of answers with respect to the original given math question.

This chapter discusses the adaptations of this methodology to the math-aware search engine Tangent-L (Chapter 3) for the MathCQA task from the ARQMath Lab series (Section 2.4.2).

The outline of this chapter is as follows: Sections 4.1, 4.2, and 4.3 present the methodology, Section 4.4 describes the experiments conducted for each adaption proposed in the methodology, and Section 4.5 discusses the submission runs and results for the team MathDowsers in ARQMath-1 and ARQMath-2.

4.1 Query Conversion: Creating Search Queries from Math Questions

In the MathCQA task, math questions are real-life questions selected from question posts of the latest MSE collection that are withheld from the task participants during training. Each given math question is formatted as a *topic* (Figure 3). Inside a topic, the title and body text are raw text, and together they describe the question in mathematical natural language. The tags indicate the question's academic areas.

In order to formulate a topic into a query consisting of a bag of keywords and formulas for input to the math-aware search engine, a rule-based approach is considered, with the following motivation:

1. filter away *unwanted* terms: unwanted terms are terms that lack the power to represent the topic and thus should be safe to remove from the query;
2. extract *useful* terms for searching: useful terms should capture key ideas of the topic and thus be helpful in locating relevant matches.

Below is a summary of potential rules that might be adopted to obtain query formulas and query keywords, respectively.

4.1.1 Basic Formula Extraction

The rules to extract formulas are simple: first, formulas within the topic's title and body text are selected into a formula pool. All formulas within the title are selected as query formulas. Formulas within the body text are selected only if they are not single variables (e.g., n or i) nor isolated numbers (e.g., 1 or 25), as shown in Figure 7.

As an input for Tangent-L, the extracted formulas are replaced by their Presentation MathML representations to serve as the query formulas for the topic.

Topic-ID: A.75

Title: Prove that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$

Body: I'm unsure how to show that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$. Looking at the solutions it starts with $e^u > \frac{u^{m+1}}{(m+1)!}$ but not sure how this is a logical step.

Tags: real-analysis, calculus, limits

Figure 7: Extracted formulas for a topic.

4.1.2 Keyword Extraction with "Mathy" Words

For a given topic, because both the topic title and the topic body text are raw text, first, the raw texts are processed with standard English NLP lower-casing and tokenization[1] to become a sequence of tokens. Stopwords among these tokens are unwanted terms and thus further removed by a standard English stopword list[2].

The remaining tokens, together with the topic tags, create a keyword pool. To extract useful keywords from the pool, a simple rule—without human intervention to actually understand the topic—is to keep any "mathy" words that are thought to represent a math concept or a math subject. One naive criterion for mathy words is to keep any hyphenated words *and their subwords* since these words are usually useful proper nouns, for instance, "Euler-Totient" (and "Euler", "Totient") or "Cesáro-Stolz" (and "Cesáro", "Stolz"). Another way is to pre-build a list of mathy words by considering the following two resources:

MSE Tags which refer to all available tags provided in the given MSE Collection. Many tags are multiwords, within which each subword might also represent a math concept, for instance, tags such as "linear-algebra" or "complex-analysis"[3]. As such, each subword of a multiword is also automatically considered.

NTCIR-12 MathIR Wikipedia Article Titles which refers to the title of a set of Wikipedia Articles used as the dataset in the NTCIR-12 MathIR Task (Section 2.2). This set of Wikipedia articles include documents explaining a *scientific*

[1]The Treebank tokenizer from the Python NLTK library (https://www.nltk.org/) is used.

[2]The stopword list is a combination of stopwords provided by the Python NLTK library and Snowball (https://snowballstem.org/), with punctuations and numerics.

[3]Each tag is either a single word or a hyphenated multiword.

Topic-ID: A.75

Title: Prove that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$

Body: I'm unsure how to show that for each integer m, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$. Looking at the solutions it starts with $e^u > \frac{u^{m+1}}{(m+1)!}$ but not sure how this is a logical step.

Tags: real-analysis, calculus, limits

Extra Keywords: real, analysis

From Both | From MSE Tags | From NTCIR-12 MathIR Wikipedia Article Titles

Figure 8: Extracted keywords for a topic using word lists from different sources.

concept, and thus their titles are also words or short expressions that include "mathy" words or scientific words. For implementation, the article's filename, such as "Algebra_(ring_theory).html" or "Algebraic_geometry_of_projective_spaces.html", is used instead of the actual title to get the words for consideration.

Around 1,500 words and 22,000 words are built from the two resources, respectively, after text cleaning and stopword removal. Example lists of words are attached in Appendix C. Tokens from the keyword pool are then compared against the lists of words, and all words that having a matching *stem*[4] are preserved (Figure 8), and the matches serve as the query keywords for the topic.

It is, however, worth noting that formulas are part of the raw text within the topic title and the topic body text. Putting aside operators, variables, and numerics, keywords such as `sin`, `cos`, `tan`, `mod` can also be extracted from the formula representations following the above rule-based approach.

[4]The Porter stemmer from the Python NLTK library is used.

4.2 Math-aware Retrieval: Searching Indexed Corpus for Best Matches

4.2.1 Different Forms of Retrievals

In this MathCQA challenge, a math-aware search engine serves to narrow the space of potential answers across the whole MSE collection concerning a specific math question. Given the CQA structure of the provided MSE collection (Section 2.4.1), various data—other than answer posts alone—might be composed to form the target corpus that the math-aware search engine searches. One can consider the following different approaches:

Answer Post Retrieval: The target corpus is formed by all answer posts from the forum, which is the most straightforward approach for the MathCQA task. Given a math question as a formal query, the math-aware search engine then retrieves from the corpus a pool of answer posts that best match the query, followed by a ranking to submit for the task.

Question-Answer Pair Retrieval: Considering the associated question post of each answer post as a valuable attribute, the target corpus is formed by answer posts that are attached to their associated question posts, that is, *question-answer pairs*. Given a math question as a formal query, the math-aware search engine then retrieves from the corpus a pool of question-answer pairs, each of which *together*[5] matches the query best. The task is then completed by ranking the answers of those retrieved pairs.

Question Post Retrieval: Instead of answer posts, the math-aware search engine might focus on retrieving question posts that match the query best, that is, searching for related questions. With the assumption that *related questions share their answers*, the target corpus is then formed by all question posts from the forum, and the task is completed by ranking the associated answers of the posts.

Thread Retrieval: Similar to a question post retrieval, but considering the associated answer posts of each question post as valuable attributes, the target corpus is formed by threads—that is, question posts together with all of their associated answers. The math-aware search engine then retrieves the best-matched threads, followed by a ranking on the answer posts among the retrieved threads to submit for the task.

[5] As our teachers admonished: "Always include the question as part of your answer!"

The above retrievals might be summarized as *answer retrieval* (answer post retrieval and question-answer pair retrieval) or *question retrieval* (question post retrieval and thread retrieval) respectively based on the searching target. While Tangent-L's internal ranking serves naturally as an answer ranking during an answer retrieval, extra step is necessary after a question retrieval to produce a ranked list of answers. A simple approach for the extra step is to order all associated answers by first favoring a larger retrieval score of its associated question post, breaking ties by a larger answer vote score, which is provided in the MSE collection (Section 2.4.1).

4.2.2 Parameter Tuning for Tangent-L

When Tangent-L serves as the retrieval system, the following parameters influence its effectiveness:

The α parameter, which is used to balance the weight of formulas to keywords within a search query (Equation 3.4, 3.5, 3.9).

The γ parameter, which is used to balance the weight of repeated symbols to other math features generated for a query formula (Equation 3.5, 3.7).

Flags for semantic classes: which is used to decide which type of semantic class is supported in formula normalization (Section 3.3), and which math features are created accordingly during indexing and searching.

and also the following parameter during a Holistic Formula Search (Section 3.4):

The κ parameter: which is used to decide the number of similar formulas being used to replace the original query formula (Equation 3.10).

Depending on the search queries and the form of corpus, parameter tuning might be beneficial for Tangent-L to adapt to the MathCQA challenge.

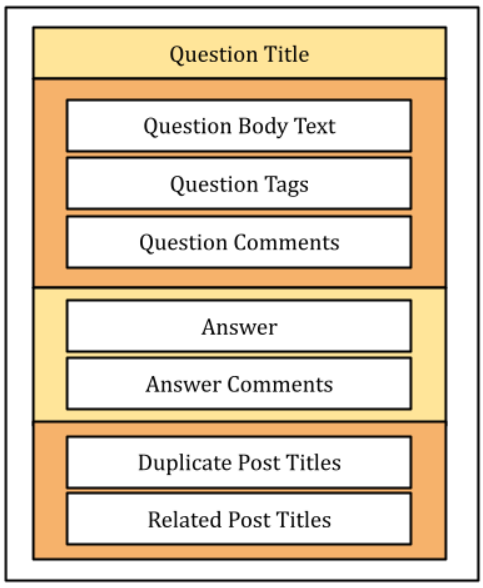


Figure 9: The structure of a corpus unit used in Question-Answer Pair Retrieval.

4.2.3 Creating Indexing Units

Regardless of which discussed retrieval approach in Section 4.2.1 is adopted, answer posts and question posts from the MSE collection are the core components to compose the target corpus units. To make use of the rest of the collection, *enriched* answer posts and *enriched* question posts are proposed for composing the corpus units, as follows:

Enriched Answer Post: the actual answer content attached with comments specific to this answer post.

Enriched Question Post: the actual question content—including the title, the body text, and tags—attached with comments specific to this question post, and the *titles* of all related and duplicated posts for this question post.

An example structure of a corpus unit using the enriched components is depicted in Figure 9. The corpus is then indexed by Tangent-L (Section 3.1.1) for searching. As an input for Tangent-L, all formulas existing in the corpus units are replaced by their Presentation MathML representations before indexing.

4.2.4 Data Cleansing for Formula Files

While building the corpus, the completeness and correctness of Presentation MathML representations in encoding input formulas are crucial to the effectiveness of Tangent-L, since they affect how well Tangent-L creates the math features during indexing and searching (Section 3.1.1). However, in the ARQMath corpus, the provided formula representation files have missing formulas due to conversion failures (Section 2.4.1). Furthermore, other errors are observed even for the up-to-date ARQMath-2 dataset. The following paragraphs describe the errors and the adopted remedies for Tangent-L.

Correcting HTML Conversion Errors

The provided Presentation MathML representation files contain conversion errors for formula instances including either less-than "<" or greater-than ">" operators. These originate from HTML-escaping. For example, when a LaTeX formula contains the operator "<", the provided formula presentation files give an incorrect encoding in Presentation MathML as shown in Table 10.

Expected Presentation MathML	*Erroneous Presentation MathML Provided*

`<mrow>`
`  <mrow>`
`  <mn>0.9999</mn>`
`  <mi mathvariant="normal">...</mi>`
`  <mo>&lt;</mo>`
`  <mn>1</mn>`
`  </mrow>`
`<mrow>`
	`<mrow>`
	`  <mrow>`
	`  <mn>0.9999</mn>`
	`  <mo></mo>`
	`  <mi mathvariant="normal">...</mi>`
	`  <mo></mo>`
	`  <mi mathvariant="normal">&amp;</mi>`
	`  <mo></mo>`
	`  <mi>l</mi>`
	`  <mo></mo>`
	`  <mi>t</mi>`
	`  </mrow>`
	`  <mo>;</mo>`
	`  <mn>1</mn>`
	`</mrow>`

Table 10: Erroneous Presentation MathML for the LaTeX formula "0.999... < 1" (formula id 382). The left hand side is the expected encoding, which is converted from the LaTeX formula first, followed by an HTML escaping from "<" to "`<`". The right hand side is erroneous, and would be generated by first converting from an already HTML-escaped LaTeX formula, which is wrong, followed by a second HTML escaping and thus creates a broken Presentation MathML that encodes "`&lt;`".

After applying a simple transformation script, during indexing and searching, Tangent-L uses corrected Presentation MathML ($\sim$ 3% of total formula instances).

Providing Missing Formula Identifiers

```
<p><span class="math-container">$\rm(2)\quad  1+J \subseteq U,\quad\ \ $</span>  i.e. <span class="math-
container">$\rm\: 1 + j\:$</span>  is a unit for every <span class="math-container">$\rm\: j \in J$</span></p>
<p><span class="math-container">$\rm(3)\quad  I\neq 1\ \Rightarrow\  I+J \neq 1,\qquad\ $</span>  i.e.  proper
ideals survive in <span class="math-container">$\rm\:R/J$</span></p>
<p><span class="math-container">$\rm(4)\quad M\:$</span> max <span class="math-container">$\rm\:\Rightarrow  M+J
\ne 1,\quad $</span>  i.e. max ideals survive in <span class="math-container">$\rm\:R/J$</span></p>
<p><strong>Proof</strong> <span class="math-container">$\: $</span> (sketch) <span class="math-container"
id="2276">\ </span>  With  <span class="math-container">$\rm\:i \in I,\ j \in J,\:$</span> and max ideal <span
class="math-container">$\rm\:M,$</span></p>
```

Figure 10: Partial text from an answer post (post id 2653) includes "`math-container`" blocks but without "`id`" attributes, even though the corresponding formulas are included in the formula representation files with formula-ids from 2285 to 2296.

In spite of the fact that formula instances present in the MSE collection files should be annotated (Section 2.4.1), some of them ($\sim$ 10% of total formula instances) are not correctly and completely captured and thus cannot be matched against the provided formula representation files. For example, many are missing their unique formula identifiers in the annotation, as shown in Figure 10.

To correct this, a modified version of the provided MSE collection files is used when building the target corpus for Tangent-L. Incomplete formula annotations such as those from Figure 10 are recognized as much as possible through regular expression matching for text within the `$` and `$$` blocks. These are then checked against the provided formula representation files to reverse-trace their Presentation MathML representations.

4.3 Answer Ranking: Finalizing the Ranked Answers

While Tangent-L's retrieval score decides a degree of similarity based on keywords and formulas, other signals provided from the dataset might be incorporated into the final answer ranking as well. The following subsections describe a few attempts.

4.3.1 Incorporating CQA Metadata

One of the provided CQA metadata fields is the *vote score*. The vote score of an answer post, which is computed from the number of received up-votes and down-votes for the post, reflects the community's belief in the answer's value for its associated question. Naturally, one might assume that an answer post with a higher vote score should be more valuable. A linear regression model can be built to make use of this assumption.

Linear Regression Model with CQA Metadata

Assuming that the vote score and potentially other CQA metadata might have a linear relationship with the answer relevance score, a linear regression model can be built to predict how variable a potential answer is with respect to any question.

Given an answer a, its associated question q, and an arbitrary question Q, the following CQA metadata are considered to build the model: (1) the *vote score* of a, (2) the *user reputation* of the author of a, and (3) the number of overlapping *tags* of q with Q. Together with Tangent-L's retrieval score of a with respect to Q, a linear function of the four variables can be fit with training data to learn its coefficients. The trained linear regression model can then be used to re-rank any list of answers retrieved by Tangent-L during an answer retrieval.

Mock Relevance by Vote Score and Question Relatedness

Because manual relevance assessment has been limited (Section 2.4.1) for the mentioned model, selected CQA metadata might be used to create a pool of mock relevance scores. The following mock relevance score is proposed, which is a real number with range [0, 3] and takes into account the vote score of an answer and the *question relatedness* of its associated question with respect to any question:

Question Relatedness: A form of question relatedness might be defined using the CQA metadata *duplicate posts*, *related posts*, and *tags* available for any two questions existing in the MSE collection. A proposed form of question relatedness for any two questions q_1, q_2 is a set of ordinal numbers as:

$$\text{Relatedness}_{\text{CQAMetaData}}(q_1, q_2) = \begin{cases} 2 & \text{if } q_1, q_2 \text{ are identical or duplicate posts} \\ 1 & \text{if } q_1, q_2 \text{ are related posts} \\ 0 & \text{if } q_1, q_2 \text{ have overlapping tags} \\ -1 & \text{otherwise} \end{cases} \tag{4.1}$$

In-thread Vote Score: For every thread with a question q and all associated answers A_q, let VoteScore(a, q) be the vote score of an answer $a \in A_q$ received with respect to q. Define vote scores for the thread as

$$\begin{aligned} \text{ThreadVotes}(A_q) &= \text{ThreadVotes}_{\text{pos}}(A_q) + \text{ThreadVotes}_{\text{neg}}(A_q) \\ \text{ThreadVotes}_{\text{pos}}(A_q) &= \sum_{\substack{a \in A_q \\ \text{VoteScore}(a,q) \geq 0}} \text{VoteScore}(a, q) \\ \text{ThreadVotes}_{\text{neg}}(A_q) &= \sum_{\substack{a \in A_q \\ \text{VoteScore}(a,q) < 0}} \big|\text{VoteScore}(a, q)\big| \end{aligned} \tag{4.2}$$

which are the sums of absolute vote scores of all or a subset of its associated answers. For every associated answer a, an *in-thread* vote score is then defined to be

$$\text{VoteScore}_{\text{InThread}}(a, q) = \begin{cases} \dfrac{\text{VoteScore}(a, q) + \text{ThreadVotes}_{\text{neg}}(A_q)}{\text{ThreadVotes}(A_q)} & \text{if VoteScore}(a, q) \geq 0 \\ \dfrac{\text{VoteScore}(a, q)}{\text{ThreadVotes}(A_q)} & \text{if VoteScore}(a, q) < 0 \end{cases} \tag{4.3}$$

which has a range of [-1, 1]. It reflects the importance of the answer a within the thread by comparing its vote score with vote scores received by other answers in the same thread.

The proposed mock relevance score for any answer a, its associated question q and an arbitrary question $\mathtt{Q}$ is then defined as:

$$\begin{aligned} &\text{MockRelevance}(a, q, \mathtt{Q}) \\ &\quad = \max\left(\text{Relatedness}_{\text{CQAMetaData}}(q, \mathtt{Q}) + \text{VoteScore}_{\text{InThread}}(a, q),\ 0\right) \end{aligned} \tag{4.4}$$

which is a real value having the same range [0, 3] as the ordinal manual relevance assessment.

4.3.2 Ranking by Proximity

Span:	length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences
Normalized-Span:	length of the shortest document segment that covers all query term occurrences in a document, including repeated occurrences, divided by the number of matched instances
Min-Span:	length of the shortest document segment that covers each matched query term at least once in a document
Normalized-Min-Span:	length of the shortest document segment that covers each matched query term at least once in a document, divided by the number of matched query terms
Min-Distance:	smallest distance value of all pairs of unique matched query terms
Ave-Distance:	average distance value of all pairs of unique matched query terms
Max-Distance:	largest value of all pairs of unique matched query terms

Table 11: Various proximity measures [57], each of which can also be normalized by document length.

Proximity is a measure of distance between matched query terms as detailed in Table 11. For the vanilla version of Tangent-L, Fraser has shown that [14] proximity might help improve ranking in the math-aware search engine Tangent-L, if each generated math token of the same formula is considered as the same position in proximity calculation. Answer ranking might thus be performed according to the proximity signal of the retrieved answers, breaking ties by a decreasing retrieval score provided by Tangent-L.

4.4 Experimental Runs for Best Configuration

This section discusses various experiments conducted for each alternative proposed in the three-stage methodology for the MathCQA task. Table 12 is a summary of findings for the best configuration observed from the experiments, which is explained in the subsequent subsections.

Query Conversion		
Search Queries:	$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$	(Section 4.4.2)
Mathaware Retrieval		
Form of Retrieval:	$\text{Corpus}_{\text{QAPair}}$, with Tangent-L's Internal Ranking	(Section 4.4.3)
Core Tangent-L		
Ranking Formula:	(α, γ) pair selected from: $0.2 \leq \alpha \leq 0.3$, $0.0 \leq \gamma \leq 0.2$	(Section 4.4.4)
Formula Normalization:	Disable FN_{IE}	
Tangent-L Variant		
Holistic Formula Search:	$\kappa = 400, \alpha = 0.47$ (Observing κ up to 500, with $\gamma = 0.1$)	(Section 4.4.6)
Answer Ranking		
Re-Ranking:	No re-ranking by Linear Regression Model	(Section 4.4.8)
	No re-ranking by Proximity	(Section 4.4.7)

Table 12: A summary of findings for the best configuration observed from experiments.

4.4.1 Setup for Evaluation

Throughout the experiments, the effectiveness for the MathCQA task is decided by the ARQMath-2 benchmark (Section 2.4.2), with the ARQMath-1 benchmark used for data study and parameter tuning.

For all retrieval tasks, top-1000 results are evaluated with a primary metric nDCG′ and secondary metrics P′@10 and MAP′, following the convention used in the ARQMath Lab series (Section 2.4.2). The metric bpref (Section 2.1.3) is also evaluated, which is a metric previously used for evaluating formula retrieval tasks. In addition, a variant of nDCG′, which evaluates the *potentially-best* possible nDCG′ (Equation 2.8) that might be achieved by the retrieval result if an optimal re-ranking is applied, is also computed with its definition as follows:

$$\text{nDCG}'^{\text{PB}}_p = \frac{\text{DCG}'^{\text{PB}}_p}{\text{IDCG}_p}\text{, where} \quad \text{DCG}'^{\text{PB}}_p = \sum_{i=1}^{REL_p^{Local}} \frac{rel_i}{\log_2(i+1)} \tag{4.5}$$

with REL_p^{Local} represents the list of relevant items *in the retrieval result ordered by their relevance* up to position p. This variant gives insight into the retrieval power of the testing system without being affected by the actual ranking methodology.

While differences exist between the provided datasets for ARQMath-1 and ARQMath-2 (Section 2.4.1), the ARQMath-2 version is adopted when creating the necessary input for Tangent-L, together with data cleansing as explained in Section 4.2.4.

The *core* version of Tangent-L used, unless during parameter tuning, is the version that incorporates repeated symbols (Section 3.2) and supports formula normalization by the semantic classes *Commutativity* and *Symmetry* (Section 3.3).

Exceptions to the above were the runs submitted in 2020 as part of ARQMath-1, when the benchmark, the latest version of dataset, and different system variants of Tangent-L were not available yet. These exceptions are addressed accordingly in Section 4.5.

Baselines	
$\text{Query}_{\text{manual}}$	A baseline approach, which is to manually select keywords and formulas from the given topics. A set of such queries for ARQMath-1 topics is available from the Lab organizers (Appendix B.1).
$\text{Query}_{\text{plain}}$	A baseline approach, in which all formulas and all keyword terms are extracted exactly as they appear.
Rule-based	
$\text{Query}_{\text{rmStop}_{\text{FF}}}$	Keyword terms are extracted with proper nouns, followed by stopword removal; and formulas in body text are filtered.
$\text{Query}_{\text{MSE}_{\text{FF}}}$	Keyword terms are extracted only if they fall into the list created from the MSE Tags source.
$\text{Query}_{\text{Wiki}_{\text{FF}}}$	Keyword terms are extracted only if they fall into the list created from the Wikipedia source.
$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$	Keyword terms are extracted only if they fall into either the list created from the Wikipedia source, or the list created from the MSE Tags source.

Table 13: Tested approaches for generating search queries (Section 4.1).

4.4.2 Comparing Generated Search Queries

This subsection examines the effectiveness of search queries generated by the rule-based approach suggested in Section 4.1, as depicted in Table 13.

For each described approach, the summary statistics of the keyword count and the formula count of the generated set of ARQMath-1 search queries can be found in Table 14. Observing from the term counts of $\text{Query}_{\text{plain}}$, it can be deduced that the length of each given topic varies widely: the keyword count ranges from four to 795, while the formula count ranges from one to 46 across different topics. Affected by this, all the proposed rule-based approaches also result in search queries with a relatively large range of term counts. Only in the approach $\text{Query}_{\text{manual}}$, where terms are manually selected, are the term counts across topics stable, with each topic having up to five keywords and up to two formulas. It is also observed that the means of *formulas-to-all-terms* vary among different approaches: with the smallest mean-ratio to be 0.048 from $\text{Query}_{\text{plain}}$ and the largest means to be 0.282 from $\text{Query}_{\text{MSE}_{\text{FF}}}$.

	Keyword Counts				*Formula Counts*			
Approach adopted	Mean	Std	Min	Max	Mean	Std	Min	Max
$\text{Query}_{\text{manual}}$	3.36	1.29	1	5	1.08	0.310	0	2
$\text{Query}_{\text{plain}}$	207	153	44	795	10.3	8.81	1	46
$\text{Query}_{\text{rmStop}_{\text{FF}}}$	48.2	35.0	9	202	8.60	7.10	1	35
$\text{Query}_{\text{MSE}_{\text{FF}}}$	22.4	14.4	3	64	8.60	7.10	1	35
$\text{Query}_{\text{Wiki}_{\text{FF}}}$	35.2	23.9	5	127	8.60	7.10	1	35
$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$	36.0	24.4	5	134	8.60	7.10	1	35

	Formulas-to-All-Terms Ratio			
Approach adopted	Mean	Std	Min	Max
$\text{Query}_{\text{manual}}$	0.265	0.103	0.000	0.667
$\text{Query}_{\text{plain}}$	0.048	0.022	0.005	0.108
$\text{Query}_{\text{rmStop}_{\text{FF}}}$	0.165	0.103	0.015	0.571
$\text{Query}_{\text{MSE}_{\text{FF}}}$	0.282	0.155	0.032	0.714
$\text{Query}_{\text{Wiki}_{\text{FF}}}$	0.209	0.124	0.021	0.645
$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$	0.204	0.122	0.021	0.645

Table 14: The summary statistics of the count of extracted keywords and formulas of different sets of ARQMath-1 search queries.

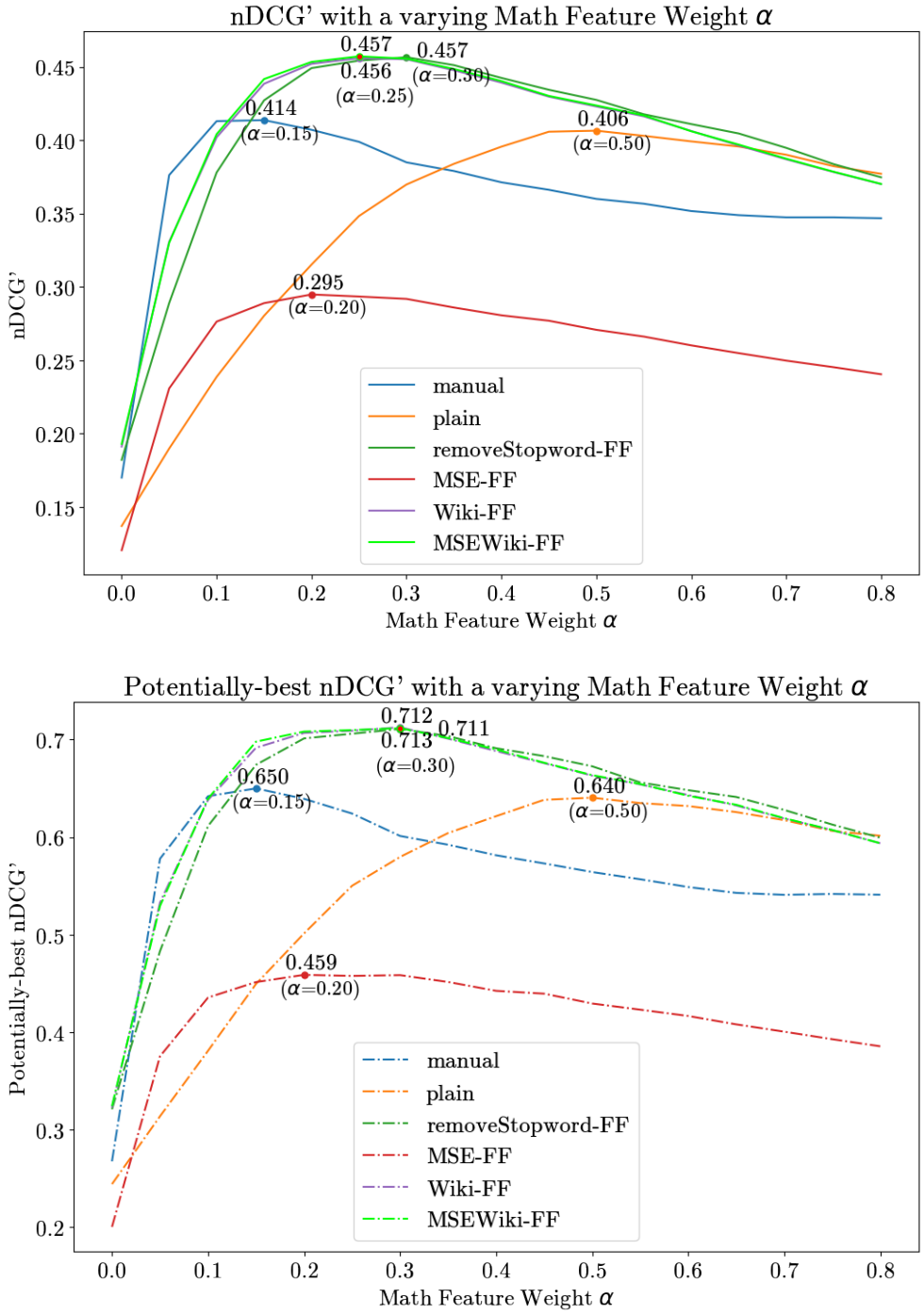


Figure 11: Evaluation of ARQMath-1 search queries, with Tangent-L having a fixed $\gamma = 0.1$ and varying α values: $0.00 \leq \alpha \leq 0.80$ and a step size of 0.05.

		ARQMath-2 (71 Topics)				
		nDCG′	MAP′†	P′@10†	bpref†	$\text{nDCG}^{\text{PB}\prime}$
$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$	$\alpha^* = 0.27$	**0.462**	**0.187**	0.241	**0.163**	**0.736**
$\text{Query}_{\text{rmStop}_{\text{FF}}}$	$\alpha^* = 0.29$	0.448	0.185	**0.245**	0.161	0.712
$\text{Query}_{\text{Wiki}_{\text{FF}}}$	$\alpha^* = 0.28$	0.442	0.180	0.235	0.156	0.708
$\text{Query}_{\text{plain}}$	$\alpha^* = 0.50$	0.418	0.173	0.240	0.152	0.666
$\text{Query}_{\text{MSE}_{\text{FF}}}$	$\alpha^* = 0.20$	0.302	0.122	0.170	0.107	0.482

† using H+M binarization

Table 15: Evaluation of different sets of ARQMath-2 search queries, each with a different optimal α value and a fixed $\gamma = 1$. The optimal α value is observed through testing on the ARQMath-1 benchmark.

Experimental runs for these sets of search queries are executed on the Question-Answer Pair corpus (Section 4.2.1). To examine the retrieval power of the approaches, $\text{nDCG}^{\text{PB}\prime}$ (Equation 4.5) is also evaluated in addition to nDCG′. Because the means of the ratio of formulas-to-all-terms across different approaches vary, it can be unfair to compare the effectiveness of these sets of search queries using a fixed α value, since α determines how much weight is given to the formula terms during searching. As such, experimental runs are executed with Tangent-L having a varying α value: $0.00 \leq \alpha \leq 0.80$ and a step size of 0.05, as shown in Figure 11.

It can be observed that the performance among different sets of search queries has been consistent regardless of whether nDCG′ or $\text{nDCG}^{\text{PB}\prime}$ is evaluated. This indicates that Tangent-L's internal ranking represents its retrieval power well.

It also appears that different sets of search queries have a unique range of optimal α values that generates the best result for them. In particular, sets of queries with a smaller mean-ratio of formulas-to-all-terms seem to have a range of larger optimal α values, such as $\text{Query}_{\text{plain}}$ with an optimal range of α value at around 0.50, and vice versa—though an exception can also be observed from $\text{Query}_{\text{manual}}$, which has a smaller mean-ratio and also a range of smaller optimal α values when compared with $\text{Query}_{\text{MSE}_{\text{FF}}}$.

If only the peak performance from each set of search queries is considered, the three approaches: $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$, $\text{Query}_{\text{rmStop}_{\text{FF}}}$, and $\text{Query}_{\text{Wiki}_{\text{FF}}}$ outperform other approaches on the ARQMath-1 benchmark, including the two baseline approaches. They have a very close performance, hinting that when the α value is carefully chosen (such as avoiding $\alpha < 0.15$ or $\alpha > 0.65$), a search result can be improved by simply filtering stopwords and unwanted formulas among all available tokens extracted from a topic, and by doing so,

	ARQMath-1 (77 Topics)				
	nDCG′	MAP′†	P′@10†	bpref†	nDCG$^{\text{PB}\prime}$
$\text{Query}_{\text{MSEWiki}_{\text{FF}}}$					
$\alpha = 0.25$	**0.457**	**0.207**	0.266	**0.191**	0.710
$\alpha = 0.26$	0.456	0.206	**0.268**	0.190	0.710
$\alpha = 0.27$	**0.457**	**0.207**	0.266	0.190	0.711
$\alpha = 0.28$	0.456	0.206	**0.268**	0.190	0.711
$\alpha = 0.29$	0.456	**0.207**	0.265	0.190	0.710
$\alpha = 0.30$	0.456	**0.207**	0.265	0.190	**0.712**
$\text{Query}_{\text{rmStop}_{\text{FF}}}$					
$\alpha = 0.25$	0.454	0.206	0.265	0.192	0.706
$\alpha = 0.26$	0.456	0.207	**0.268**	**0.193**	0.710
$\alpha = 0.27$	0.457	**0.208**	0.262	**0.193**	0.710
$\alpha = 0.28$	0.457	0.207	0.264	0.192	**0.712**
$\alpha = 0.29$	**0.458**	**0.208**	0.264	**0.193**	**0.712**
$\alpha = 0.30$	0.457	**0.208**	**0.268**	**0.193**	0.711
$\text{Query}_{\text{Wiki}_{\text{FF}}}$					
$\alpha = 0.25$	**0.456**	0.206	0.262	**0.190**	0.709
$\alpha = 0.26$	0.455	0.205	**0.265**	0.189	0.710
$\alpha = 0.27$	**0.456**	0.206	**0.265**	0.189	0.710
$\alpha = 0.28$	**0.456**	0.206	**0.265**	0.188	0.711
$\alpha = 0.29$	**0.456**	0.206	0.262	0.189	0.711
$\alpha = 0.30$	0.455	**0.207**	0.261	0.189	**0.713**

† using H+M binarization

Table 16: A closer examination of the three rule-based approaches on ARQMath-1 topics, with Tangent-L having a fixed $\gamma = 0.1$ and varying α values: $0.25 \leq \alpha \leq 0.30$ and a step size of 0.01. While most results are indistinguishable from each other, an optimal α is picked for each run (highlighted in red) based on a larger nDCG′, breaking ties by a larger nDCG$^{\text{PB}\prime}$, or otherwise selected randomly.

the result is already better than searches using manually selected keywords and formulas. It is, however, unclear whether keywords should be further filtered by external sources of lists of "mathy" words, since the results from $\text{Query}_{\text{rmStop}_{\text{FF}}}$ and the other two approaches are almost indistinguishable.

The three runs also have a similar range of optimal α values at around 0.25 to 0.30. A closer examination of this α range with a step size of 0.01 for these three runs is found in Table 16. Handpicking the optimal α values from both Figure 11 and Table 16, a final evaluation against the ARQMath-2 benchmark is shown in Table 15. With ARQMath-2 topics, $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ has an outstanding performance especially for nDCG$'$ and $\text{nDCG}^{\text{PB}\prime}$. It might thus be concluded that, extraction by lists of mathy words generated from the MSE Tags source and the Wikipedia source helps the retrieval performance overall.

Baseline	
$\text{Corpus}_{\text{Answer}}$	A corpus formed by answer posts.
Alternative	
$\text{Corpus}_{\text{QAPair}}$	A corpus formed by question-answer pairs, that is, each unit is an answer post with its associated question post.
$\text{Corpus}_{\text{Question}}$	A corpus formed by question posts.
$\text{Corpus}_{\text{Thread}}$	A corpus formed by threads, that is, each unit is a question post with all its associated answer posts.

Table 17: Corpora for different forms of math-aware retrieval. All answer posts and question posts for building the corpus units are *enriched* posts as explained in Section 4.2.3.

4.4.3 Comparing Corpora

This subsection examines which form of the corpus described in Section 4.2.1 and summarized in Table 17 is the best choice for the current system to adopt. $\text{Corpus}_{\text{Answer}}$ serves as a baseline since it is the naive approach to address the problem of ranking answers from the MSE collection (containing *only* answer posts).

Experimental runs using ARQMath-1 search queries from the presumably good approach $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ (Section 4.4.2) are executed on different corpora. For $\text{Corpus}_{\text{Answer}}$ and $\text{Corpus}_{\text{QAPair}}$, nDCG′ is evaluated with the list of ranked answers returned from Tangent-L directly, while for $\text{Corpus}_{\text{Question}}$ and $\text{Corpus}_{\text{Thread}}$, nDCG′ is evaluated with the list of ranked answers ordered first by the retrieval score of the retrieved questions and then the vote score of the answer, *selected up to 1,000*. In addition, $\text{nDCG}^{\text{PB}\prime}$ is also examined for all runs in which the order of answers are optimized. The result is shown in Figure 12, with Tangent-L having a varying α value: $0.0 \leq \alpha \leq 0.8$ and a step size of 0.1.

It can be observed that regardless of the choice of the corpus, all runs have an optimal α range at around 0.2 to 0.3, which is likely a consequence of using $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ as discussed previously. Considering their peak performance, the baseline $\text{Corpus}_{\text{Answer}}$ as well as $\text{Corpus}_{\text{Question}}$ do not perform as well as the other two corpora for both evaluation measures, suggesting that it is always better to include the content of both question post and answer post(s) when composing a corpus unit.

On the other hand, for nDCG′, $\text{Corpus}_{\text{QAPair}}$ outperforms $\text{Corpus}_{\text{Thread}}$ (0.456 vs 0.390), while for $\text{nDCG}^{\text{PB}\prime}$ the situation is the opposite: $\text{Corpus}_{\text{Thread}}$ outperforms $\text{Corpus}_{\text{QAPair}}$ (0.773 vs 0.712). The good performance of $\text{Corpus}_{\text{Thread}}$ for $\text{nDCG}^{\text{PB}\prime}$ might be explained by the fact that it has a larger pool of available answer candidates (all associated answers from

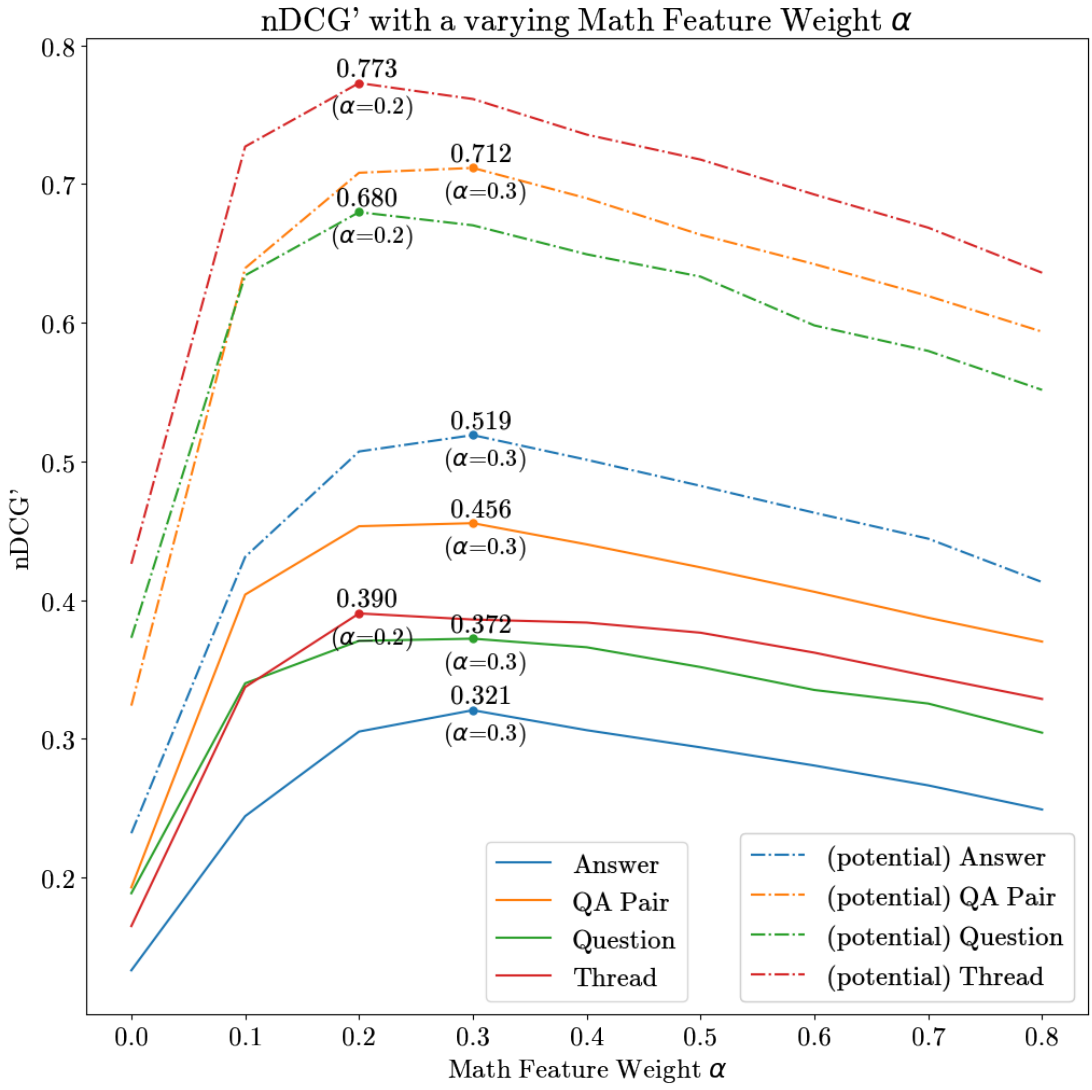


Figure 12: ARQMath-1 evaluation on different corpora, with Tangent-L having a fixed $\gamma = 0.1$ and varying α values: $0.0 \leq \alpha \leq 0.8$ and a step size of 0.1. $\text{nDCG}^{\text{PB}\prime}$ is reported together in the same graph (represented by the dashdotted lines).

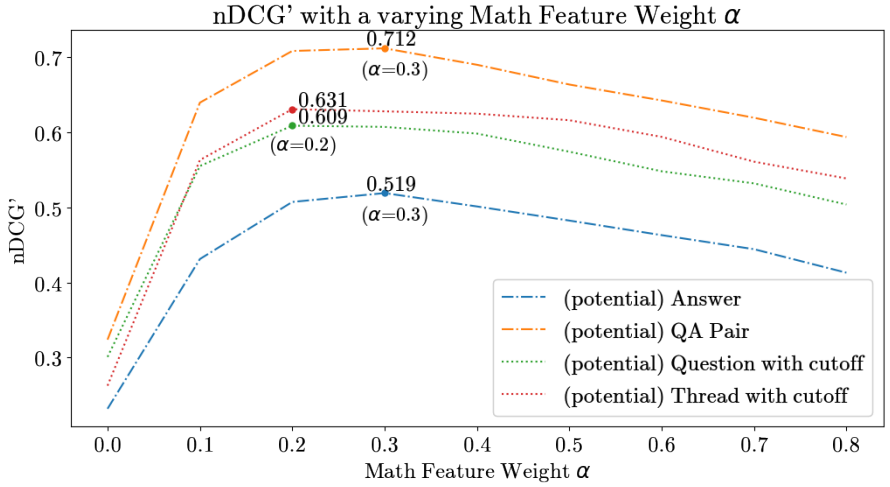


Figure 13: Similar to Figure 12, but evaluate $\text{Corpus}_{\text{Question}}$ and $\text{Corpus}_{\text{Thread}}$ with a cutoff on their pool of answer candidates before an optimal re-ranking.

the retrieved questions) for ranking. However, if such a pool of answer candidates is limited to 1,000 first before being evaluated for $\text{nDCG}^{\text{PB}\prime}$, its performance drops significantly and $\text{Corpus}_{\text{QAPair}}$ still has the best performance among all, as shown in Figure 13. Nonetheless, the result indicates that, while an answer retrieval among Question-Answer Pairs performs well enough with Tangent L's internal ranking, a question retrieval among threads has the potential to achieve a better result if a proper answer ranking can be applied. Table 18 shows the final evaluation result on the ARQMath-2 benchmark for different corpora, with Tangent-L having a fixed $\alpha = 0.27$ (the presumably optimal α value for $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ discussed previously).

	ARQMath-2 (71 Topics)				
	nDCG′	MAP′†	P′@10†	bpref†	$\text{nDCG}^{\text{PB}\prime}$
$\text{Corpus}_{\text{QAPair}}$	**0.462**	**0.187**	**0.241**	**0.163**	0.736
$\text{Corpus}_{\text{Thread}}$	0.427	0.119	0.151	0.092	**0.787**
$\text{Corpus}_{\text{Question}}$	0.400	0.125	0.180	0.106	0.704
$\text{Corpus}_{\text{Answer}}$	0.278	0.087	0.176	0.096	0.489

† using H+M binarization

Table 18: ARQMath-2 Evaluation on different corpora, with search queries from $\text{Query}_{\text{MSEWikiFF}}$ and Tangent-L set to $\alpha = 0.27, \gamma = 0.1$.

4.4.4 Core Tangent-L: Fine Tuning α, γ and Formula Normalization

The previous sections (4.4.2, 4.4.3) have shown that Tangent-L has a range of optimal α values at around 0.20 to 0.30 when working with $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$, but the experiments to check this α value have been conducted only with a fixed repeated symbol weight $\gamma = 0.1$ and a support for formula normalization by the semantic classes *Commutativity* and *Symmetry*. This section summarizes the effects of the γ value and formula normalization respectively.

Effect of the Repeated Symbols Weight γ

To explore the effect of the weight of repeated symbols in Equation 3.6, experimental runs on the ARQMath-1 benchmark with $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ and $\text{Corpus}_{\text{QAPair}}$ are repeated with a varying γ and α value: $0.0 \leq \gamma, \alpha \leq 1.0$ and a step size of 0.1, as shown in Figure 14.

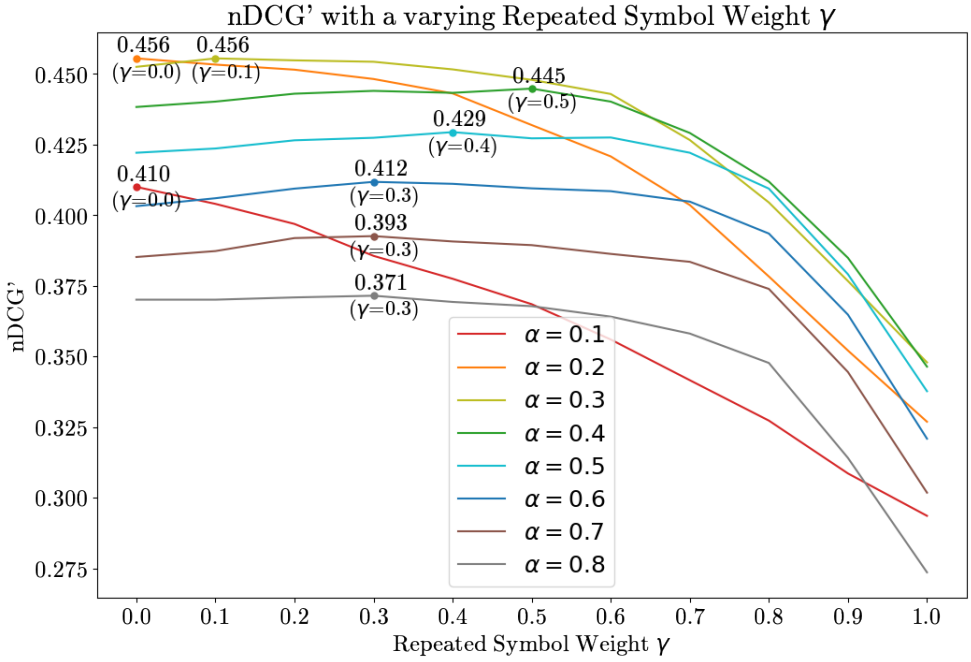


Figure 14: ARQMath-1 Evaluation with search queries from $\text{Query}_{\text{MSEWikiFF}}$ applied on $\text{Corpus}_{\text{QAPair}}$, and Tangent-L set as: $0.0 \leq \gamma, \alpha \leq 1.0$ with a step size of 0.1.

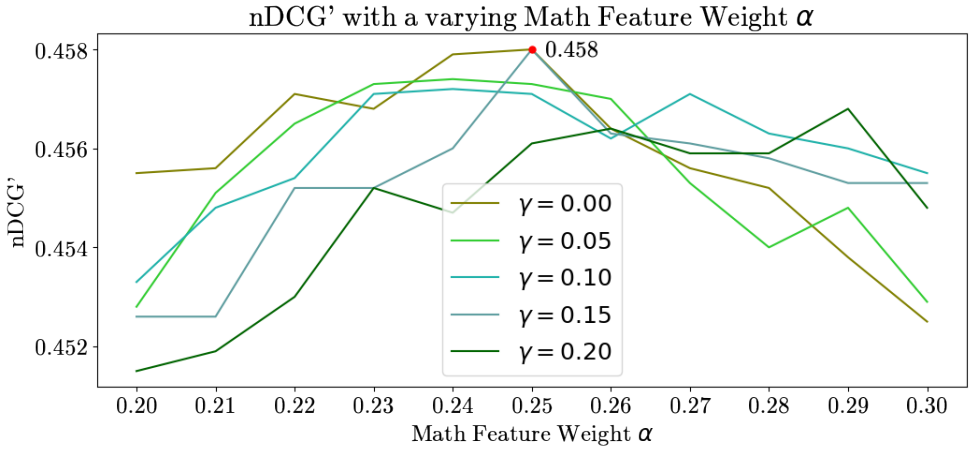


Figure 15: Similar to Figure 14 but with a closer examination of the γ values at different α with $0.00 \leq \gamma \leq 0.20 \leq \alpha \leq 0.30$, and a step size of 0.05 for γ and 0.01 for α respectively.

It can be observed that a unique range of optimal γ values exists for different α values. Nonetheless, after considering all α, γ pairs, the range of optimal α values remains roughly at around 0.2 to 0.3, which validates the previous testing. The corresponding range of optimal γ values is between 0.0 and 0.2. Notice also that an increasing γ gives slight improvement to some α values at particular ranges—which implies that the existence of repeated symbols helps improve the retrieval effectiveness to some degree. However, in general, α plays a larger role than γ in the evaluation measure: nDCG$'$ changes faster with a change of α than with a change of γ most of the time (unless with a small α value like 0.1 or a large γ value that is greater than 0.8 when an increasing γ drops the performance significantly).

A closer examination of finer γ and α values can be found in Figure 15 with $0.00 \leq \gamma \leq 0.20 \leq \alpha \leq 0.30$, and a step size of 0.05 for γ and 0.01 for α respectively. While the peak nDCG$'$ is at 0.458 with $\alpha = 0.25$ and $\gamma = 0.00$ or 0.15, such result is not significant since there are multiple pairs having similar values with a difference smaller than 10^{-3}. As such, it might be concluded that $0.0 \leq \gamma \leq 0.2 \leq \alpha \leq 0.3$ is an optimal range for the parameter pairs at the current experimental setting, and more evidence is necessary to look for their optimal values with a finer precision.

As a reference, the peak α, γ pairs in Figure 15 are evaluated and compared to the previously-evaluated pair $\alpha = 0.27, \gamma = 0.10$ against the ARQMath-2 benchmark in Table 19. The difference between the performance of the pairs is insignificant.

	ARQMath-2 (71 Topics)				
	nDCG′	MAP′†	P′@10†	bpref†	nDCG$^{\text{PB}\prime}$
$\alpha = 0.27, \gamma = 0.10$	**0.462**	**0.187**	0.241	0.163	**0.736**
$\alpha = 0.25, \gamma = 0.00$	0.461	**0.187**	**0.247**	**0.164**	**0.736**
$\alpha = 0.25, \gamma = 0.15$	0.461	0.186	0.242	0.162	**0.736**

† using H+M binarization

Table 19: ARQMath-2 Evaluation on different α, γ values of Tangent-L, with search queries from $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ executed on $\text{Corpus}_{\text{QAPair}}$. The α, γ values are selected among the observed range of optimal values from the ARQMath-1 benchmark.

Effect of Semantic Classes for Formula Normalization

Baseline	
No FN	Formula normalization is not supported.
Flags for Semantic Classes	
$\text{FN}_{\text{C+S}}$	The semantic class *Commutativity* and *Symmetry* are supported.
$\text{FN}_{\text{AN+OU}}$	The semantic classes *Alternative Notation* and *Operator Unification* are supported.
FN_{IE}	The semantic class *Inequality Equivalence* is supported.

Table 20: The available flags in Tangent-L to control whether or not to support a semantic class for formula normalization (Section 3.3).

To explore the effect of normalization, experimental runs on the ARQMath-1 benchmark are repeated with $\text{Query}_{\text{MSEWikiFF}}$ and $\text{Corpus}_{\text{QAPair}}$, each time with a different flag turned on to support formula normalization (Table 20). Given the previous result of the range of optimal α and γ values, runs are executed with a fixed $\gamma = 0.1$ on a smaller range of α values: $0.15 \leq \alpha \leq 0.35$ and a step size of 0.05. The result is shown at Figure 16.

It can be observed that the effect of formula normalization is not significant—except for FN_{IE}, which produces a noticeable drop in performance. The worse performance by FN_{IE} might be owing to the implementation limitation (3.3.2), however, such limitation has not resulted in the same drop in performance for $\text{FN}_{\text{C+S}}$. It remains to be investigated why reversing expressions with inequality operators produces a worse performance. A final evaluation against the ARQMath-2 benchmark is shown in Table 21.

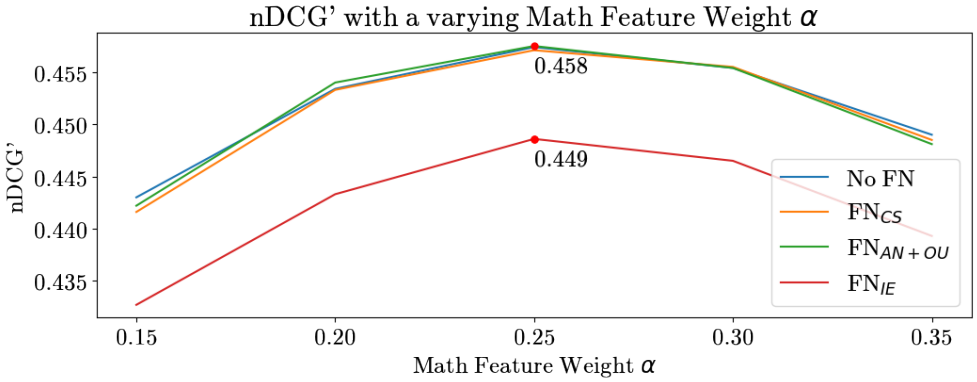


Figure 16: ARQMath-1 Evaluation with search queries from $\text{Query}_{\text{MSEWikiFF}}$ applied on $\text{Corpus}_{\text{QAPair}}$, and Tangent-L set as $\gamma = 0.1$, and $0.15 \leq \alpha \leq 0.35$ with a step size of 0.5.

4.4.5 Core Tangent-L: Fine Tuning α for Individual Queries

From Table 14, it can be observed that search queries produced by $\text{Query}_{\text{MSEWikiFF}}$ have different formula-to-all-terms ratio across different topics. A hypothesis is that, first, a fixed α value might hinder the performance for particular topics; and next, this might be because of this varying formula-to-all-terms ratio. To look at this into more detail, the performance of individual topics with a varying α value: $0.0 \leq \alpha \leq 0.8$ and a step size of 0.05 is plotted at Figure 17 (where the search queries are applied on $\text{Corpus}_{\text{QAPair}}$ with a fixed $\gamma = 0.1$).

It can be observed that the optimal α value (indicated by the red dot on each colored line) for each individual topic fluctuates from 0.05 to 0.8, while the mean of these α values (indicated by the red vertical line) is around 0.31— a close approximation to the optimal range of a *fixed* α value concluded in the previous section. It thus confirms the first part of the hypothesis: that a varying α value can help improve the performance of individual topics.

On the other hand, keyword counts and formula counts (or the counts of regular math tokens—repetition tokens are omitted due to their insignificance after being scaled by $\gamma = 0.1$), together with the optimal α value for each individual topic, are shown in Figure 18. However, no obvious relationship can be observed between the counts and the optimal α value, which rejects the second part of the hypothesis. This observation is also in line with that observed by Fraser in his thesis, in which he found that "there is not a clear relationship between math to keywords ratio and α" [14].

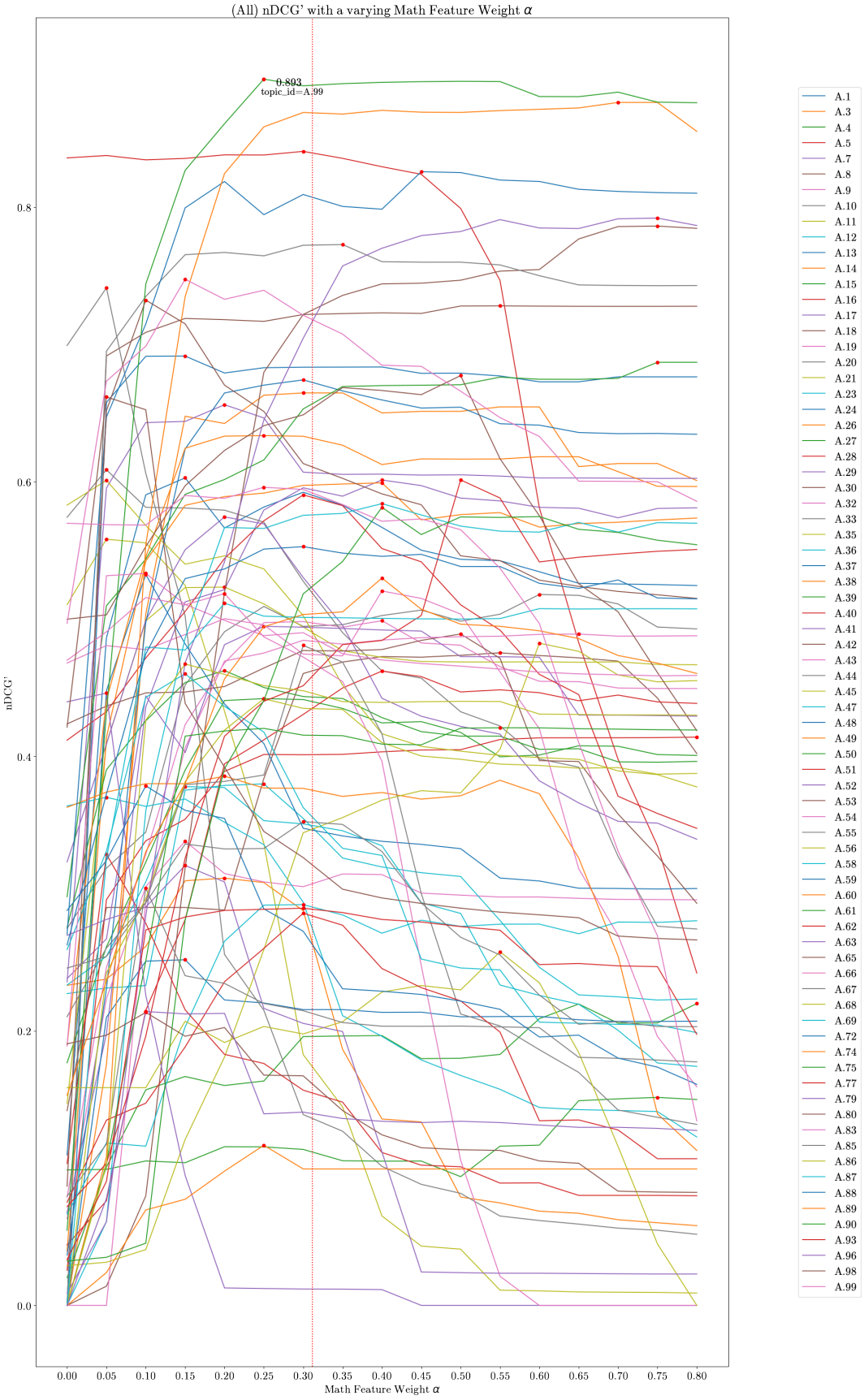


Figure 17: ARQMath-1 Evaluation on individual topics with a varying $\alpha : 0 \leq \alpha \leq 0.8$, with a step size of 0.05

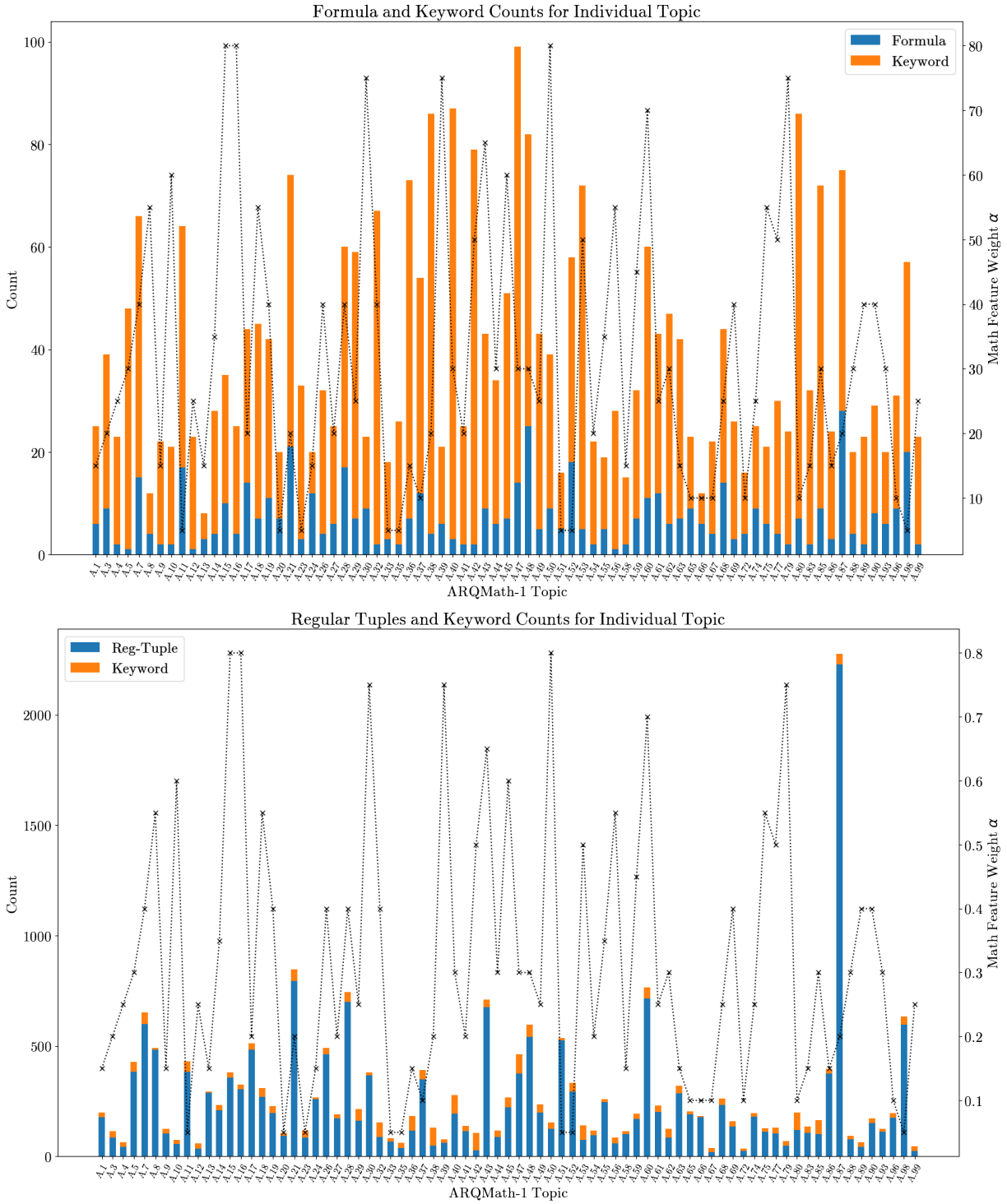


Figure 18: Keyword counts and formula (or regular math tokens) counts for the search queries from $\text{Query}_{\text{MSEWikiFF}}$ of individual topics, together with their individual optimal α value from Figure 17.

	ARQMath-2 (71 Topics)				
	nDCG′	MAP′†	P′@10†	bpref†	$\text{nDCG}^{\text{PB}\prime}$
No FN	**0.463**	**0.187**	0.242	**0.163**	**0.737**
$\text{FN}_{\text{AN+OU}}$	0.462	0.186	**0.247**	**0.163**	0.735
$\text{FN}_{\text{C+S}}$	0.462	**0.187**	0.241	**0.163**	0.736
FN_{IE}	0.447	0.180	0.237	0.157	0.717

† using H+M binarization

Table 21: ARQMath-2 Evaluation for different semantic classes supported in formula normalization, with search queries from $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ executed on $\text{Corpus}_{\text{QAPair}}$ and Tangent-L set to $\alpha = 0.27, \gamma = 0.1$.

Another hypothesis is that the optimal α value for individual topics might also depends on whether the topic is formula-dependent, text-dependent, or both (Table 5). Subplottings of Figure 17 that isolate topics with their dependencies are shown in Appendix D. Again, no obvious relationship can be observed between the topic dependencies and the associated range of optimal α values. It might thus be concluded that it remains a challenge to fine-tune the α value based on individual topics.

4.4.6 Tangent-L Variant: Exploring Holistic Formula Search

This section explores the effectiveness of Holistic Formula Search as proposed in Section 3.4.

As a system variant of the core Tangent-L, Holistic Formula Search also has an α parameter, a γ parameter, and flags for formula normalization. However, it is different by being a two-stage retrieval system model, in which γ is involved in the first stage of formula retrieval model (Equation 3.7), and α and κ are involved in the second stage of keyword-and-formula retrieval (Equation 3.9, 3.10). Therefore, the effect of α, γ, and κ parameters are examined together as follows:

With again $\text{Query}_{\text{MSEWikiFF}}$ for search queries and $\text{Corpus}_{\text{QAPair}}$ for the retrieval corpus, experimental runs on the ARQMath-1 benchmark are executed with $\gamma = 0.1$ for formula retrieval and varying α and κ values for keyword-and-formula retrieval: $0.0 \leq \alpha \leq 0.8$ with a step size of 0.1, and $\kappa = 1, 5, 10, 20, 50, 100, 200, 300, 400, 500$. The result is shown in Figure 19.

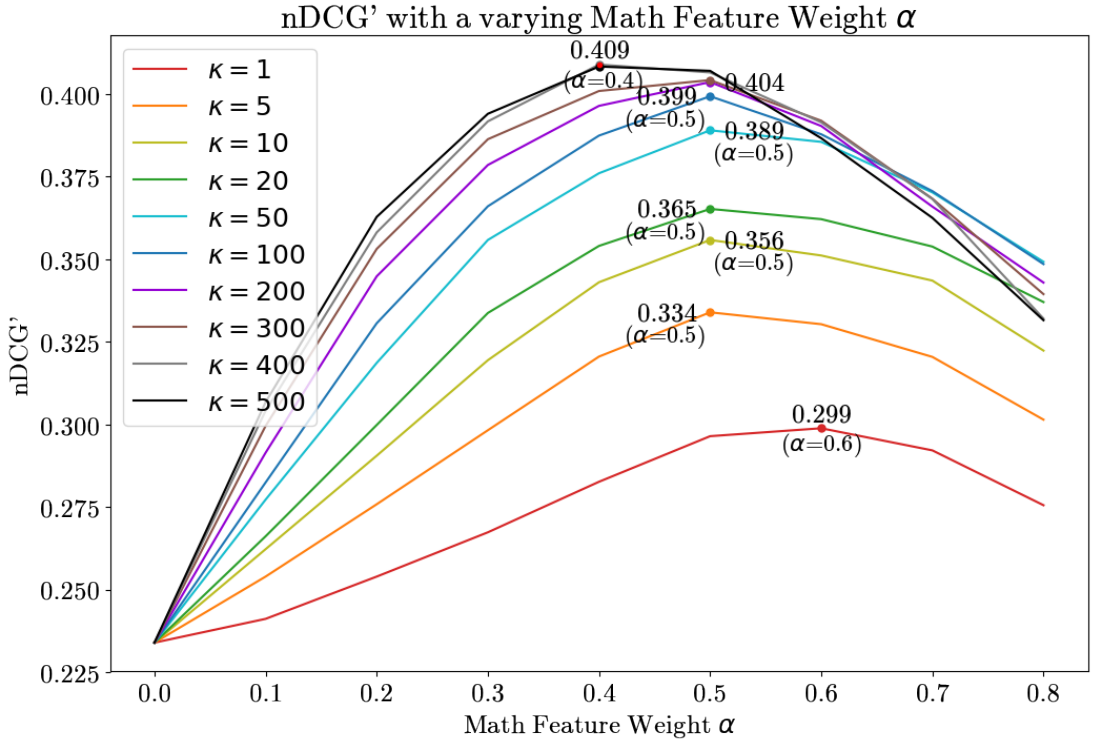


Figure 19: ARQMath-1 Evaluation using different κ values as the number of replacement formulas in a Holistic Formula Search. Tangent-L is set to have $\gamma = 0.1$ during formula retrieval, and when in document retrieval, search queries from $\text{Query}_{\text{MSEWikiFF}}$ are applied on $\text{Corpus}_{\text{QAPair}}$ with a varying α value: $0.0 \leq \alpha \leq 0.8$ and a step size of 0.1.

It can also be observed that each κ has a unique range of optimal values, and when κ increases, the corresponding optimal α value tends to shift to a smaller value. This makes sense since less weight should be given to query formulas to compensate for the potential "noisy" replacement formulas.

With a fixed query formula weight α, effectiveness generally increases with the increase of the number of replacement formulas κ (observed up to 500) for each query formula, that is, the more formulas the better. However, the increase in effectiveness slows down when κ grows large, and the effectiveness for $\kappa = 400$ and $\kappa = 500$ becomes similar. On the other hand, although the change of γ value is not tested, it might be deduced that the value of γ—excluding huge values—contributes less to the optimal performance when κ is large because the ranking power of the formula retrieval model becomes less important.

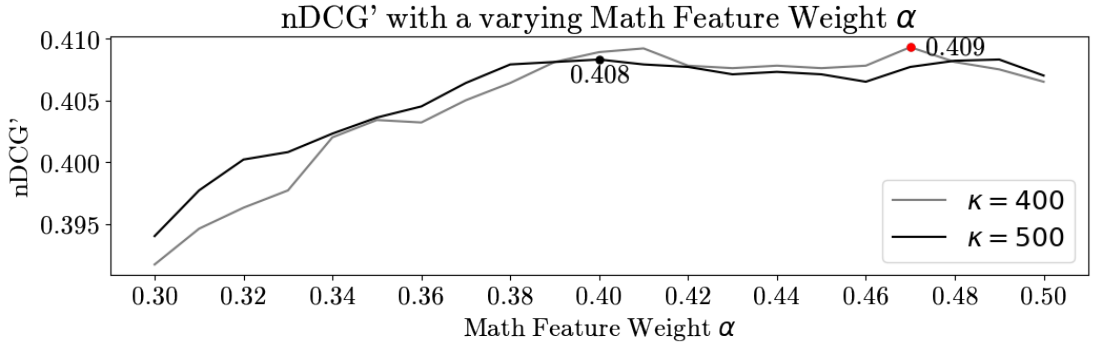


Figure 20: Similar to Figure 19 but with a closer examination of the result from $\kappa = 400, 500$ at $0.30 \leq \alpha \leq 0.50$ and a step size of 0.01.

	ARQMath-2 (71 Topics)				
	nDCG′	MAP′†	P′@10†	bpref†	nDCG$^{\text{PB}\prime}$
CoreSearch$_{\alpha=0.27,\gamma=0.10}$	**0.462**	**0.187**	**0.241**	**0.163**	**0.736**
HolisticFormulaSearch$_{\gamma=0.10,\alpha=0.47,\kappa=400}$	0.413	0.164	0.223	0.149	0.676

† using H+M binarization

Table 22: ARQMath-2 Evaluation for Holistic Formula Search compared to the search by the core version of Tangent-L, with search queries from Query$_{\text{MSEWiki}_{\text{FF}}}$ applied on Corpus$_{\text{QAPair}}$.

Considering each κ value with their optimal α value, the range of peak performance is at around $\kappa = 400$ or 500 with $\alpha = 0.4$. A closer examination of $\kappa = 400, 500$ in this range is shown in Figure 20, with a varying α value: $0.30 \leq \alpha \leq 0.50$ and a step size of 0.01. It can be observed that the peak is achieved by $\kappa = 400$ at $\alpha = 0.47$. As a final evaluation, this peak pair is evaluated and compared to the result from the previously discussed one-stage approach with $\alpha = 0.27, \gamma = 0.1$ against the ARQMath-2 benchmark. It might be concluded that Holistic Formula Search is less effective than regular search (at least until larger κ values are tested).

	Δ(H,M)	Δ(M,L)	Δ(L,NR)	Δ(M,NR)	Δ(H,NR)
Span	7%	8%	3%	10%	18%
Span-NormByDocLen	0%	1%	5%	5%	5%
Normalized-Span	-5%	-6%	-62%	-67%	-72%
Normalized-Span-NormByDocLen	-20%	-13%	-64%	-76%	-92%
Min-Span	9%	7%	6%	13%	21%
Min-Span-NormByDocLen	-1%	2%	8%	11%	10%
Normalized-Min-Span	2%	1%	-39%	-38%	-36%
Normalized-Min-Span-NormByDocLen	-11%	-3%	-40%	-43%	-53%
Min-Distance	1%	-2%	-89%	-90%	-89%
Min-Distance-NormByDocLen	-10%	-9%	-104%	-111%	-117%
Ave-Distance	4%	3%	-16%	-14%	-10%
Ave-Distance-NormByDocLen	-7%	-2%	-15%	-17%	-24%
Max-Distance	9%	7%	6%	13%	21%
Max-Distance-NormByDocLen	-1%	2%	9%	11%	10%

Table 23: Comparison of proximity measures on the ARQMath-1 benchmark for math answers of high (H), medium (M), low (L) relevance, and non-relevant (NR) math answers, where $\Delta(a, b) = \frac{\text{prox}(a)-\text{prox}(b)}{0.5(\text{prox}(a)+\text{prox}(b))}$.

4.4.7 Validating Proximity

The experimental design used by Tao and Zhai [57] can be used to explore the effect of each proximity signal outlined in Table 11 (Section 4.3.2). With respect to ARQMath-1 search queries created from $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$, the average proximity for documents from $\text{Corpus}_{\text{QAPair}}$ that have ARQMath-1 relevance assessment is shown in Table 23. It can be observed that strong signals from several measures distinguish relevance with the correct order (marked in gradient orange), particularly for normalized-span, which correctly orders all four levels of relevancy (a smaller normalized-span indicating a higher level of relevancy) without the need to be normalized by document length.

As part of the submission to ARQMath-2, a re-ranking is performed on the retrieved answers by Tangent-L in increasing order of normalized-span, breaking ties by a decreasing retrieval score returned from Tangent-L. However, the result explained in Section E.2 demonstrates that such re-ranking is unsatisfactory. A closer look at the computed normalized-span values shows that they have ranges of small magnitude (<0.001 to 0.05), hinting that the percentage differences between each class of documents might have been insignificant in their absolute values.

As a follow-up, the effect of proximity on document relevance is further studied as follows: first, divide a document into p portions of its content; then create d document

	ARQMath-1 (77 Topics)				*ARQMath-2 (71 Topics)*			
	nDCG′	MAP′†	P′@10†	bpref†	nDCG′	MAP′†	P′@10†	bpref†
docOnly	0.452	0.207	0.267	0.190	**0.462**	0.187	**0.241**	**0.163**
docWithFrags	**0.462**	**0.212**	**0.268**	**0.193**	0.458	**0.188**	0.239	0.159

† using H+M binarization

Table 24: ARQMath-1 and ARQMath-2 Evaluation when including document fragments.

Selected Document	*ARQMath-1*	*ARQMath-2*
Whole	∼75%	∼75%
Top Fragment	∼11%	∼13%
Middle Fragment	∼6%	∼6%
Bottom Fragment	∼8%	∼7%

Table 25: Percentage of selected document instances when searching a document corpus with document fragments.

fragments, where each document fragment contains portions of the original document with a sliding window size of w. The document fragments and the original document are then indexed together to create a new document corpus. During a search, each returned document fragment represents its original document, and the final list of rankings is produced by eliminating repeated documents that occur later in the ranking.

The motivation behind this approach is that if query terms in a relevant document are closer to each other (as observed in Table 23), then including the document fragments for searching might boost those relevant documents towards the top of the ranking. This study is carried out with $p = 4, d = 3, w = 2$ (that is, each document is represented by its top-half, "middle half", and bottom half, as well as the whole) and the result compared to using a regular corpus is shown in Table 24. It can be observed that while the studied approach has a better effectiveness on the ARQMath-1 benchmark, it does not help improve the performance on the ARQMath-2 benchmark. Table 24 shows that a large portion (∼75%) of the selected documents are the original documents, hinting again that proximity is somewhat limited.

Noticeably, holistic formula search (Section 4.4.6) is also an approach motivated by proximity, with the hypothesis that if math tokens in a query appear closer together in a document (or are forced to be reduced to a holistic formula token in a document), then that document is likely to be more relevant. Nonetheless, the approach has not produced

a promising result as well. Further study is required to incorporate the observed proximity signals for a better result.

4.4.8 Validating CQA Metadata

Vote Score

A key assumption behind the approach proposed in Section 4.3.1 is that an answer post with a higher vote score should be more valuable. However, the proposed linear model and the mock relevance score oversimplifies the relationship between the vote score and the actual relevance assessment.

As can be seen from Figure 21, most of the vote scores cluster at a small value that centers around one, and large vote scores are sparse. A similar distribution can be observed from the vote scores of the answer posts that have received ARQMath-1 relevance assessment. Observing also the average relevance received by ARQMath-1 assessed answer posts with a particular vote score, the average relevance for large vote scores (say, with a value larger than 40) fluctuates possibly due to the sparsity of those vote scores. It is, therefore, inappropriate to assume a *linear*, or even a *monotonic* relationship between the two even though there is a trend that the average relevance increases with the increase of vote scores when the vote scores are small. A similar conclusion can be deduced for the in-thread vote score component (Equation 4.3) of the proposed mock relevance score as well.

Question Relatedness

Another assumption behind the design of the mock relevance score is about *question relatedness*: that an answer post is more valuable if the associated question of the answer post is linked to the given math question through the attributes *duplicated post*, *related posts*, and their number of *overlapping tags*. To validate this assumption, the average relevance is computed for assessed answer posts in ARQMath-1 whose associated question posts have the mentioned relations with the given math question, as shown in Table 26. Notice that both duplicate posts and related posts of the given math questions are not available to the participants before the evaluation. The information of duplicate posts is available only from the task result of the baseline system *Linked MSE posts* (Section 2.4.2).

From the computation, it can be observed that the average relevance generally increases with the number of overlapping tags. Although the average relevance reaches zero when

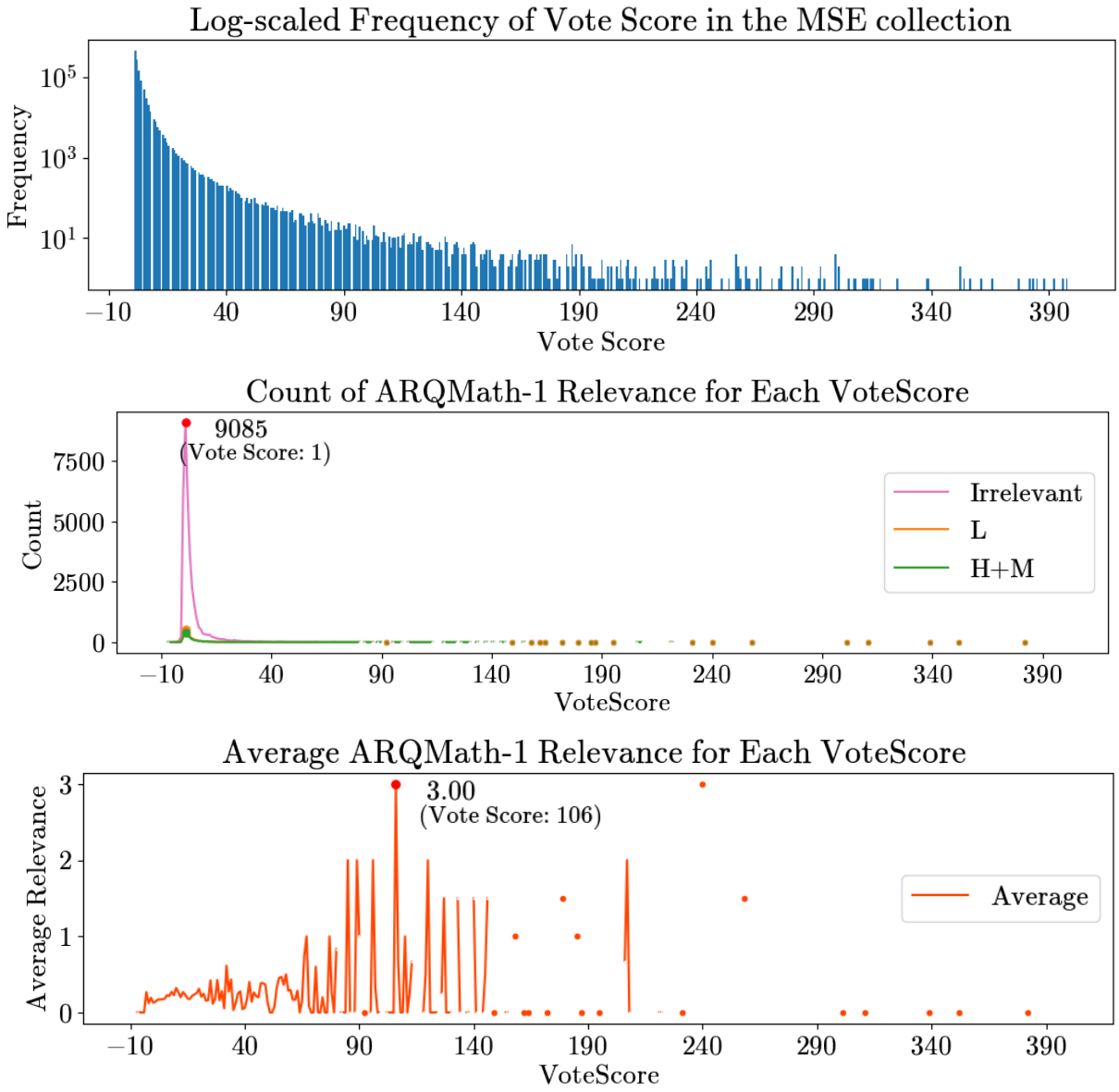


Figure 21: Overview for the CQA metadata *vote score*, displaying vote score from the minimum score -10 up to 400. Larger vote scores are ignored due to their sparseness. The ARQMath-1 relevance is denoted by H (a value of 3), M (a value of 2), L (a value of 1) and Irrelevant (a value of 0) respectively in the middle graph.

Relation of the Associated Question Post	*Average Relevance*	*Number of Answer Posts*
Any	0.176	36097
Number of Overlapping Tags: 0	0.091	22303
1	0.253	12718
2	0.315	3531
3	0.447	566
4	0	6
Duplicate Posts	1.56	723

Table 26: The average relevance for ARQMath-1 assessed answer posts, with respect to the relation between their associated question posts and the given math question.

the number of overlapping tags reaches its highest, the assessed number of answer posts is relatively small and thus might not be have a great influence. Comparing to the overall average relevance—whose associated question might have any relationship with the given math question—all relations, except for the previously mentioned case and the case when there is no overlapping tags, have a higher average relevance. As such, it might be concluded that these attributes provide some hint to the value of an answer post regarding the task.

It is, however, worth noting that while the average relevance for *duplicate posts* is the highest among all, its value at 1.56 is still fairly low if considering the fact that the associated question and the given math question are deemed by the forum users as duplicates, or in other words, identical. The effectiveness of the *question relatedness* score component (Equation 4.1) in the mock relevance score might thus be compromised.

Trained Linear Regression Model

As part of the submission to ARQMath-1—when the above analysis was not possible— a linear regression model proposed in Section 4.3.1 is trained using training data built by first picking around 1,300 question posts from the MSE collection, and then for the top 10,000 answers that Tangent-L retrieved using the corresponding search queries, associating those answers with the proposed mock relevance score and the proposed CQA metadata according to their search queries. The trained linear regression model is then used to re-rank retrieved answers by Tangent-L for the ARQMath-1 topics. The ARQMath-1 result explained in Section E.1 shows that such re-ranking is indeed detrimental to the performance.

4.5 MathDowsers' Submission Runs and Results

4.5.1 Submissions Overview

Following the presented methodology in Section 4.1, 4.2, and 4.3, runs were submitted to ARQMath-1 and ARQMath-2. An overview of all results, including the submission runs of both years and the experimental runs, can be found in Table 27 with notations of the configuration defined in Tables 13, 17, 20, and 28.

It can be observed that the presented methodology has been successful, with MathDowsers' submitted runs achieving the best participant runs for both years in terms of the primary measure nDCG′ (0.345 by *alpha05-noR* for the ARQMath-1 benchmark and 0.434 by *primary* for the ARQMath-2 benchmark), even though the runs did not adopt the best possible configuration outlined in Section 4.4. With the best possible configuration, experimental runs achieve an even better nDCG′ (for example, 0.458 and 0.463 by *No FN* for the two benchmarks, respectively). The success of the presented methodology also proves the effectiveness of Tangent-L as a math-aware search engine.

Additionally, the performance of the runs on the ARQMath-1 benchmark—despite adopting the same methodology—has a significant improvement over the years, with the nDCG′ being increased from an initial 0.345 (by *alpha05-noR* during ARQMath-1) to 0.433 (by *primary* during ARQMath-2), followed by a further increase to 0.458 (by *No FN* during experiments). Such improvement is reflected on the ARQMath-2 benchmark as well, demonstrating an overall enhancement of the developing system.

Full details of each year's runs have been published in CLEF Working Notes [62, 30]. Some of the key discussions from those papers can be found in Appendix E.1 and E.2.

		ARQMath-1 (77 Topics)				*ARQMath-2 (71 Topics)*			
		nDCG′	MAP′†	P′@10†	bpref†	nDCG′	MAP′†	P′@10†	bpref†
MathDowsers (ARQMath-1) (Primary Settings: $\text{Dataset}_{\text{original}}$, $\text{uniqueQuery}_{\text{MSEWiki}_{\text{FF}}}$, $\text{Corpus}_{\text{QAPair}}$, vanillaTL-$\alpha_{0.50}$, $\text{reRank}_{\text{LRM}}$)									
alpha05-noR (no $\text{reRank}_{\text{LRM}}$)	$*\mathbb{B}$	0.345	0.139	0.162	0.126	-	-	-	-
alpha02 ($\alpha = 0.20$)	$*$	0.301	0.069	0.075	0.044	-	-	-	-
alpha05-trans ($\text{Query}_{\text{manual}}$)	$*\mathbb{M}$	0.298	0.074	0.079	0.050	-	-	-	-
alpha05	¶	0.278	0.063	0.073	0.041	-	-	-	-
alpha10 ($\alpha = 1.00$)	$*$	0.267	0.063	0.079	0.042	-	-	-	-
MathDowsers (ARQMath-2), tuned on ARQMath-1 (Primary Settings: $\text{Dataset}_{\text{clean}}$, $\text{uniqueQuery}_{\text{MSEWiki}_{\text{FF}}}$, $\text{Corpus}_{\text{QAPair}}$, coreTL-$\alpha_{0.27}\gamma_{0.10}$, $\text{FN}_{\text{C+S}}$)									
primary	¶$\mathbb{B}$	0.433	0.191	0.249	0.178	0.434	0.169	0.211	0.145
proximityReRank ($\text{reRank}_{\text{prox}}$)	$*$	0.373	0.117	0.131	0.095	0.335	0.081	0.049	0.052
Experimental Runs, tuned on ARQMath-1 (Primary Settings: $\text{Dataset}_{\text{clean}}$, $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$, $\text{Corpus}_{\text{QAPair}}$, coreTL-$\alpha_{0.27}\gamma_{0.10}$, $\text{FN}_{\text{C+S}}$)									
No FN		0.458	0.207	0.261	0.191	**0.463**	0.187	0.242	0.163
$\text{primary}_{\text{exp}}$		0.457	0.207	0.267	0.190	0.462	0.187	0.241	0.163
$\text{FN}_{\text{AN+OU}}$		0.457	0.206	0.262	0.191	0.462	0.186	**0.247**	0.163
coreTL-$\alpha_{0.25}\gamma_{0.00}$		0.458	0.208	0.269	0.190	0.461	0.187	**0.247**	**0.164**
coreTL-$\alpha_{0.25}\gamma_{0.15}$		0.458	0.207	0.267	0.190	0.461	0.186	0.242	0.162
docWithFrags		**0.462**	**0.212**	0.268	**0.193**	0.458	**0.188**	0.239	0.159
$\text{Query}_{\text{rmStop}_{\text{FF}}}$, coreTL-$\alpha_{0.29}\gamma_{0.10}$		0.458	0.208	0.264	**0.193**	0.448	0.185	0.245	0.161
FN_{IE}		0.448	0.199	0.260	0.184	0.447	0.180	0.237	0.157
$\text{Query}_{\text{Wiki}_{\text{FF}}}$, coreTL-$\alpha_{0.28}\gamma_{0.10}$		0.456	0.206	0.265	0.188	0.442	0.180	0.235	0.156
$\text{Corpus}_{\text{Thread}}$		0.386	0.153	0.212	0.138	0.427	0.119	0.151	0.092
$\text{Query}_{\text{plain}}$, coreTL-$\alpha_{0.50}\gamma_{0.10}$		0.406	0.188	0.248	0.178	0.418	0.173	0.240	0.152
holisticTL-$\gamma_{0.10}\alpha_{0.47}\kappa_{400}$		0.409	0.193	**0.281**	0.183	0.413	0.164	0.223	0.149
$\text{Corpus}_{\text{Question}}$		0.371	0.152	0.234	0.144	0.400	0.125	0.180	0.106
$\text{Query}_{\text{MSE}_{\text{FF}}}$, coreTL-$\alpha_{0.20}\gamma_{0.10}$		0.295	0.135	0.181	0.127	0.302	0.122	0.170	0.107
$\text{Corpus}_{\text{Answer}}$		0.316	0.114	0.201	0.121	0.278	0.087	0.176	0.096

¶ submitted primary run $*$ submitted alternate run $\mathbb{B}$ best partipant run $\mathbb{M}$ manual run

† using H+M binarization

Table 27: An overview result for the MathCQA Task in ARQMath-1 and ARQMath-2, including MathDowsers' submission runs and experimental runs. The result of the runs are ordered by their nDCG′ in ARQMath-2 (or otherwise ARQMath-1).

Data Cleansing	
$\text{Dataset}_{\text{original}}$	The original dataset provided during ARQMath-1.
$\text{Dataset}_{\text{clean}}$	An improved dataset that is provided during ARQMath-2 (Section 2.4.1), with data cleansing adopted (Section 4.2.4).
Tangent-L Variants	
vanillaTL-α_x	The vanilla version of Tangent-L (Section 3.1), with the math feature weight α set to a value of x.
coreTL-$\alpha_x\gamma_y$	The system variant of Tangent-L that incorporates repeated symbols (Section 3.2), with the math feature weight α set to a value of x, and the repeated symbol weight γ set to a value of y.
holisticTL-$\gamma_y\alpha_x\kappa_z$	The system variant of Tangent-L that conducts Holistic Formula Search (Section 3.4), with the repeated symbol weight γ set to a value of y, the math feature weight α set to a value of x, and the number of replacement formulas κ for a query formula set to a value of z.
Query Conversion	
$\text{uniqueQuery}_{\text{MSEWiki}_{\text{FF}}}$	Compared to $\text{Query}_{\text{MSEWiki}_{\text{FF}}}$ in Table 13, the extracted terms are de-duplicated without being boosted for TangentL's internal ranking according to their frequencies. This algorithm is a result of oversight in implementation.
Answer Ranking	
$\text{reRank}_{\text{LRM}}$	Re-ranking with a trained linear regression model using CQA metadata (Section 4.3.1).
$\text{reRank}_{\text{Prox}}$	Re-ranking by proximity signal (Section 4.3.2).

Table 28: Additional notations to describe settings in Table 27. Other notations are defined in Tables 13, 17, and 20.

		ARQMath-2 (71 Topics)			
		nDCG′	MAP′ †	P′@10†	bpref†
Baselines					
Linked MSE posts	¶	(0.203)	(0.120)	**(0.282)**	(0.131)
TF-IDF+Tangent-S	¶	0.201	0.045	0.086	0.048
TF-IDF	*	0.185	0.046	0.063	0.046
Tangent-S	*	0.111	0.027	0.052	0.039
Top Experimental Run					
$\text{primary}_{\text{exp}}$		**0.462**	**0.187**	0.241	**0.163**
Top Participant Runs					
$\text{MathDowsers}_{\text{primary}}$	¶𝔹	0.434	0.163	0.211	0.145
$\text{DPRL}_{\text{QASim}}$	*	0.388	0.147	0.193	0.135
$\text{TU\_DBS}_{\text{TU\_DBS\_P}}$	¶	0.377	0.158	0.227	0.158

¶ submitted primary run ∗ submitted alternate run
𝔹 best partipant run † using H+M binarization

Table 29: Comparison of the top experimental run to the baseline runs and the top three participant runs on the ARQMath-2 benchmark. Parentheses indicates that the submissionis made with privately-held data which is not available to participants.

4.5.2 Strengths and Weaknesses

This section describes the strengths and weaknesses of the presented methodology compared to the baseline systems and the top participant systems in the task. The benchmark of focus is the ARQMath-2 benchmark, since from Appendix A it can be observed that more teams participated in ARQMath-2, and most participant systems have improved ever since ARQMath-1.

Overall Performance

Table 29 compares the performance of the top experimental run $primary_{exp}$ with the baseline runs and the top three participant runs—the submitted *primary* run, the *QASim* run from DPRL that trains a system with SentenceBERT (Section 2.5.5), and the *TU_DBS_P* run from TU_DBS that trains a system with ALBERT (Section 2.5.6).

Focusing on the top three participant runs, the submitted *primary* run, which is the best participant run, also achieves the best MAP′ in addition to nDCG′. However, it has a lower P′@10 (0.211 vs 0.227) and bpref (0.145 vs 0.158) than another participant run *TU_DBS_P*. Nonetheless, the top experimental run $primary_{exp}$, which corrects an oversight

	Count	$primary_{exp}$ nDCG′	MAP′	P′@10	bpref
Overall	71	0.462	0.187	0.241	0.163
Dependency					
Text	10	0.423	0.158	0.260	0.142
Formula	21	**0.516**	**0.235**	**0.319**	**0.204**
Both	40	0.443	0.169	0.195	0.146
Topic Type					
Computation	25	0.455	0.189	0.200	0.165
Concept	19	0.429	0.160	0.232	0.137
Proof	27	**0.492**	**0.204**	**0.285**	**0.178**
Difficulty					
Easy	32	**0.509**	**0.216**	**0.300**	**0.199**
Medium	20	0.383	0.116	0.150	0.098
Hard	19	0.466	0.213	0.237	0.169

Table 30: Effectiveness breakdown by topic categories of the top experimental run $primary_{exp}$ on the ARQMath-2 benchmark. The better performance measure within each topic category is highlighted in bold.

in the implementation of the *primary* run, improves all evaluation measures and achieves a higher P′@10 and bpref than *TU_DBS_P*.

Remarkably, the top experimental run $primary_{exp}$ also has the best nDCG′, MAP′, and bpref among all runs, including the baseline *Linked MSE posts*, which uses privately-held data that is not available to participants (Section 2.4.2). It might thus be concluded that the presented methodology with the best possible configuration has been strong in evaluations measures not limited to nDCG′, while it is still relatively weak in P′@10 overall when compared to the baseline *Linked MSE posts* (0.241 vs 0.282).

Topic Category Breakdown

Given the released topic labels by Lab organizers (Table 5), Table 30 shows the effectiveness breakdown of $primary_{exp}$ by topic category. It can be observed that the run has a strong performance for Formula-dependent topics, Proof-like topics, and topics of Easy-difficulty in all evaluation measures.

Comparing to other systems in nDCG′ as shown in Table 31, a similar category performance to $primary_{exp}$ can be observed not only from the submitted *primary* run but also

		Experimental	*Top Participant Runs*			*Baselines*			
	Count	*primary*$_{exp}$	MathDowsers *primary*	DPRL *QASim*	TU_DBS *TU_DBS_P*	*Linked MSE posts*	*TF-IDF +Tangent-S*	*TF-IDF*	*Tangent-S*
		.....			nDCG′		.....		
Overall	71	0.462	0.434	0.388	0.377	(0.203)	0.201	0.185	0.111
Dependency									
Text	10	0.423	0.385	0.347	0.360	**(0.246)**	**0.253**	**0.296**	0.042
Formula	21	**0.516**	**0.480**	**0.443**	**0.382**	(0.184)	0.194	0.128	**0.170**
Both	40	0.443	0.422	0.369	0.378	(0.207)	0.191	0.187	0.097
Topic Type									
Computation	25	0.455	0.441	0.399	0.348	(0.215)	**0.211**	0.185	**0.125**
Concept	19	0.429	0.390	0.301	0.294	**(0.217)**	0.189	0.178	0.083
Proof	27	**0.492**	**0.459**	**0.438**	**0.462**	(0.189)	0.199	**0.190**	0.117
Difficulty									
Easy	32	**0.509**	**0.472**	**0.426**	**0.426**	(0.197)	**0.236**	**0.205**	**0.131**
Medium	20	0.383	0.346	0.350	0.353	(0.188)	0.179	0.184	0.096
Hard	19	0.466	0.463	0.362	0.318	**(0.243)**	0.164	0.152	0.092

Table 31: Category performance in nDCG′ of the top experimental run, the baseline runs, and the top three participant runs on the ARQMath-2 benchmark. The better performance measure within each topic category is highlighted in bold. Parentheses indicates that the submission is made with privately-held data which is not available to participants.

from the other two top participant runs. On the other hand, a different category performance can be observed from the baselines. Most baselines have a stronger performance on Text-dependent topics rather than Formula-dependent topics, while having a relatively average performance in topics of different types. The baselines only have a similar category performance to the participant runs in terms of the difficulty level (with the only exception being *Linked MSE posts*). It might thus be concluded that, while most systems are good at topics of Easy-difficulty, the presented methodology—together with other participant systems—are particularly strong in Formula-dependent topics and Proof-like topics.

nDCG′ vs P′@10

It is worthwhile to study the performance of P′@10, since it stands out from other effective measure by the fact that it measures only the top ten judged results, instead of the whole 1,000 results. A better nDCG′ score but a poorer P′@10 score might indicate that a system has retrieved some good results but fail to rank them at early positions.

	Overall	*Dependency*			*Topic Type*			*Difficulty*		
		Text	Formula	Both	Comp.	Concept	Proof	Easy	Medium	Hard
Topic Count	77	10	21	40	25	19	27	32	20	19
					nDCG′					
Baselines										
Linked MSE posts	(0.203)	(0.246)	(0.184)	(0.207)	(0.215)	(0.217)	(0.189)	(0.197)	(0.188)	(0.243)
TF-IDF+Tangent-S	0.201	0.253	0.194	0.191	0.211	0.189	0.199	0.236	0.179	0.164
TF-IDF	0.185	0.296	0.128	0.187	0.185	0.178	0.190	0.205	0.184	0.152
Tangent-S	0.111	0.042	0.170	0.097	0.125	0.083	0.117	0.131	0.096	0.092
Top Experimental Run										
$\text{primary}_{\text{exp}}$	**0.462**	**0.423**	**0.516**	**0.443**	**0.455**	**0.429**	**0.492**	**0.509**	**0.383**	**0.466**
Top Participant Runs										
$\text{MathDowsers}_{\text{primary}}$	0.434	0.385	0.480	0.422	0.441	0.390	0.459	0.472	0.346	0.463
$\text{DPRL}_{\text{QASim}}$	0.388	0.347	0.443	0.369	0.399	0.301	0.438	0.426	0.350	0.362
$\text{TU\_DBS}_{\text{TU\_DBS\_P}}$	0.377	0.360	0.382	0.378	0.348	0.294	0.462	0.426	0.353	0.318
					P′@10					
Baselines										
Linked MSE posts	**(0.282)**	**(0.364)**	(0.250)	**(0.285)**	**(0.230)**	**(0.339)**	(0.301)	**(0.304)**	**(0.253)**	**(0.250)**
TF-IDF+Tangent-S	0.086	0.120	0.086	0.077	0.088	0.100	0.074	0.122	0.055	0.058
TF-IDF	0.063	0.150	0.043	0.053	0.048	0.089	0.059	0.091	0.050	0.032
Tangent-S	0.052	0.000	0.086	0.048	0.064	0.058	0.037	0.062	0.045	0.042
Top Experimental Run										
$\text{primary}_{\text{exp}}$	0.241	0.260	**0.319**	0.195	0.200	0.232	0.285	0.300	0.150	0.237
Top Participant Runs										
$\text{MathDowsers}_{\text{primary}}$	0.211	0.190	0.276	0.183	0.188	0.168	0.263	0.256	0.125	0.226
$\text{DPRL}_{\text{QASim}}$	0.193	0.130	0.300	0.152	0.204	0.116	0.237	0.247	0.115	0.184
$\text{TU\_DBS}_{\text{TU\_DBS\_P}}$	0.227	0.190	0.295	0.200	0.208	0.126	**0.315**	0.300	0.150	0.184

Table 32: A comparison in nDCG′ and P′@10 on the ARQMath-2 benchmark of different topic sub-categories. The better performance measure within each topic sub-category is highlighted in bold for the top experimental run, the baseline runs, and the top three participant runs.

As a supplement to Table 31, Table 32 compares the effectiveness in nDCG′ and P′@10 between the runs in different topic sub-categories. It can be observed that the strong performance of $primary_{exp}$ in nDCG′ observed from Table 29 is indeed well-rounded in every topic sub-categories. However, it is noticeable that while $primary_{exp}$ has a stronger overall P′@10 than the other two participant runs, it is not well-rounded in all topic sub-categories. In particular, *TU_DBS_P* achieves a better P′@10 for Proof-like topics (0.315 vs 0.285), and the score is the best among all runs, including even the baseline *Linked MSE posts*. It might thus be concluded that there is room for re-ranking the retrieved result to improve P′@10.

Meanwhile, it is noticable that $primary_{exp}$ also has a higher P′@10 than the baseline *Linked MSE posts* specifically for Formula-dependent topics (0.319 vs 0.250) but not for other topics. This further validates the claim that the presented methodology with the best possible configuration is particularly strong in Formula-dependent topics, demonstrating again the math-aware ability of Tangent-L.

Chapter 5

Addressing In-context Formula Retrieval

In-context formula retrieval can be viewed as a cross-disciplinary task between a regular formula retrieval task and the MathCQA Task. Similar to a regular formula retrieval task, the retrieval targets are relevant formulas with respect to some topic formulas. On top of that, the relevance of a retrieved formula is defined with respect to its *expected utility* to the topic formula, when the context—that is, the associated question post—of both the retrieved formula and the topic formula is considered (Section 2.4.2). As such, the retrieval goal is also grounded in math questions, just like the MathCQA Task.

In spite of the complex nature of this task, simple approaches are proposed based on the developed MathCQA system (Chapter 4) as side experiments. The following sections (Sections 5.1, 5.2) present the approaches, and Section 5.3 discusses the submitted runs and results by the team MathDowsers for ARQMath-2 and compares those results to the baseline and other participants' submissions.

5.1 Formula-centric: Selecting Visually Matching Formulas

One straightforward approach is to handle the task almost like more traditional formula retrieval. This approach is *formula-centric*, since its effectiveness relies heavily on Tangent-L's internal formula matching capability to find the matching formulas. The details of the approach are as follows:

First, the formula corpus of visually distinct formulas created in Section 3.4.1 is searched by Tangent-L with respect to the given topic formula, resulting in a ranking R of visually distinct formulas.

This ranking R is then used to create a ranked list of formula instances. Each element of R is first expanded with its set of formula occurrences—formulas that have the same visual-id but appear in different posts, hereafter referred to as a *visual group*. Since only question-posts and answer-posts are of concern in the task, any formula instances from comment-posts are ignored.

To produce a formula ranking more grounded in the math questions, the formula instances are selected and ranked with respect to how much their associated post is relevant to the associated question of the topic formula. To accomplish this, the result of the *primary* run (Section E.1) from the MathCQA task with up to 10,000 ranked answers is adopted to decide the ranking. In detail,

1. Formulas within the same visual group are ranked in the same order as the ranking of their associated posts in the MathCQA task for the corresponding topic. If the associated posts of formulas are question-posts that are not associated with any answer from the MathCQA task, the formulas are assigned the lowest ranking. Finally, the lexical order of formula-ids is used to break ties.

2. For each of the top-20 visually distinct formulas in R, the top five formulas from its visual group (or all formulas in the visual group if there are fewer than five) are selected; for the remainder, only the top formula instance is selected (if any have associated question or answer posts).

3. Sequentially considering the formulas in R in order, selected formula instances from each visual group are appended to the final list of matching formulas until a target of 1,000 formula instances are selected in total.

5.2 Document-centric: Screening Formulas from Matched Documents

Another straightforward approach is to return formulas with respect to a topic formula by screening formulas from posts that are relevant to the associated post of the topic formula. This approach is *document-centric*, since its effectiveness relies more on ranked documents resulting from the MathCQA task. Given the result of the *primary* run (Section E.1) from the MathCQA task with up to 10,000 ranked answers, the final matching formula instances are selected from the answers as follows:

1. For each matched answer-post for the corresponding topic in the MathCQA task, the question-answer pair (Section 4.2.3) document is considered. If the document contains only one formula, that formula is selected. Otherwise, each formula from the document is mapped to its visual group, and its *Normalized Similarity Score* (Equation 3.8) with respect to the topic formula is computed via a formula retrieval with $\gamma = 0.1$ through the formula corpus of visually distinct formulas (Section 3.4.1, but see below). Formulas having a score less than a threshold of 0.8 are screened out, and the rest are preserved and ranked accordingly.

2. Following the original answer-ranking, preserved formulas from each document are appended to the final list until 1,000 formulas are selected in total.

During implementation, it is highly inefficient to compute the Normalized Similarity Score for every formula that appears in a document, since the computation requires a formula retrieval of over 8.5 million FormulaScores (Equation 3.7) for each topic formula. Therefore, for each topic, formulas in documents that are not within the top 10,000 retrieved formulas to the topic formula are assigned a score of 0 and therefore screened out.

		ARQMath-1 (45 Topics)				*ARQMath-2* (58 Topics)			
		nDCG′	MAP′†	P′@10†	bpref†	nDCG′	MAP′†	P′@10†	bpref†
Baselines									
Tangent-S		**0.691**	**0.446**	**0.453**	**0.412**	0.492	0.272	0.419	0.290
MathDowsers									
formulaBase	¶	0.562	0.370	0.447	0.374	0.552	0.333	0.450	0.348
docBase	∗	0.404	0.251	0.386	0.275	0.433	0.257	0.359	0.291
Best Participant Run (ARQMath-2)									
$\text{Approach0}_{\text{P300}}$	∗𝕄	0.507	0.342	0.441	0.343	**0.555**	**0.361**	**0.488**	**0.362**
Best Participant Run (ARQMath-1)									
$\text{DPRL}_{\text{Tangent-CFTED}}$	*	0.563	0.388	0.436	0.372	-	-	-	-

¶ submitted primary run ∗ submitted alternate run
𝕄 manual run † using H+M binarization

Table 33: Comparison of the submitted runs to the baseline run and best participant runs.

5.3 MathDowsers' Submission Runs and Results

As a first attempt at the in-context formula retrieval challenge, a primary run and an alternative run were submitted to ARQMath-2 as follows:

formulaBase: The primary run, in which formulas are among retrieved formulas from Tangent-L as described in Section 5.1;

docBase: An alternative run in which formulas are selected from matched documents from the MathCQA task as described in Section 5.2.

The results of both runs are shown in Table 33, together with the baseline run and the best participant runs for the ARQMath-1 and ARQMath-2 benchmarks. In terms of nDCG′, the primary run *formulaBase* achieves a very close performance to the best participant run *Tangent-CFTED* produced from the DPRL team during ARQMath-1 [28] (0.562 vs 0.563). On the ARQMath-2 benchmark with an unseen set of math topics, the primary run *formulaBase* performs approximately as well with an nDCG′ score of 0.552. While the best participant run is achieved by the *P300* run from the Approach0 team [65] (an nDCG′ score of 0.555), that run is a manual run (with the use of manual mapping rules to expand math tokens in a query to additional text keywords for search) and thus, in fact, the primary run *formulaBase* is the best among all automatic runs as shown in

Appendix A.4. Its performance in nDCG′ is almost indistinguishable from that of *P300* as well, with a difference of less than one-point (0.552 vs 0.555).

On the other hand, the alternative run *docBase* does not perform well. On the ARQMath-1 benchmark, this run shows nearly a 16-point loss when compared to the primary run (0.404 vs 0.562). As a submitted run in ARQMath-2, it also shows a nearly 12-point loss (0.433 vs 0.552) in nDCG′ and achieves lower scores in all other evaluation measures.

Nonetheless, the success of the primary run *formulaBase* shows that even if the task is being handled almost like a regular formula retrieval task, Tangent-L's internal formula matching capability already serves as a strong foundation to produce a decent result. This again gives proof to the effectiveness of Tangent-L as a math-aware search engine.

Chapter 6

User Interface for Data Exploration

In addition to effectiveness measures, a visual demonstration can be of great help in understanding the performance of a system. This chapter describes two implementations that assist in data exploration.

6.1 The MathDowsers' Browser

The MathDowsers' Browser[1] is built with an aim to provide a convenient interface for users to examine the MathCQA task of the ARQMath Lab series (Figure 22). Users might:

View ARQMath Questions: select by topic categories and view the details of an ARQMath question (Appendix G.1).

View Ranked Answers: after selecting a question, view the ranked list of retrieved answers by the MathDowsers' runs (Appendix G.2); or input a custom answer ranking and view the corresponding list of answers (Appendix G.3).

Check Relevance Judgements: view also the human relevance judgements of the answers that are used to evaluate the runs (Appendix G.4).

The website provides an interface for reading the performance (as a list of ranked answers with their details) of a CQA system. It might be most useful for people to inspect the precision measure (such as P′@10) of a system manually.

[1] https://cs.uwaterloo.ca/~yk2ng/MathDowsers-ARQMath

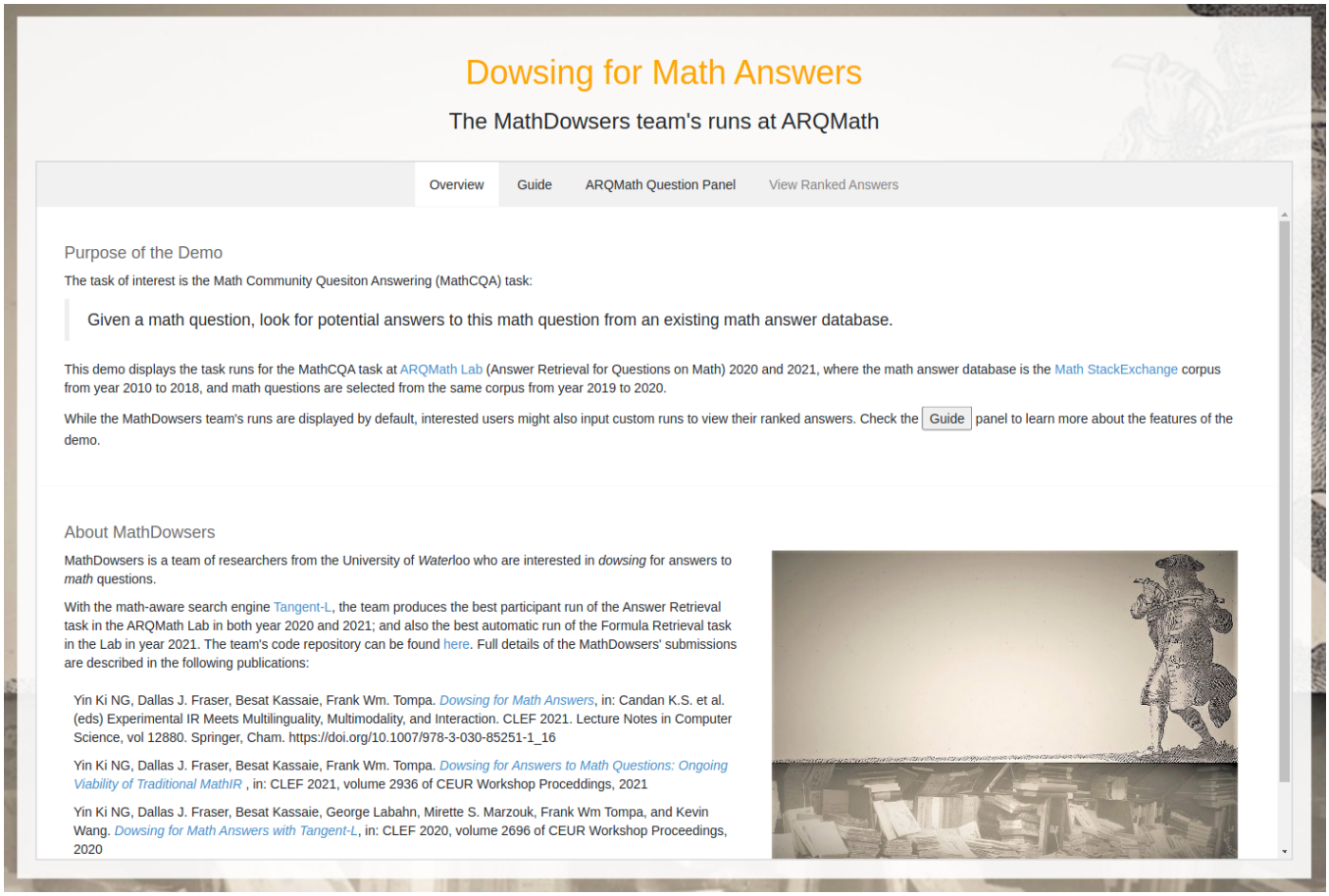


Figure 22: The MathDowsers' Browser.

The website is a static site that is written in jQuery and HTML. The displayed data from this website, including the document corpus, MathDowsers' submissions, and the human relevance judgements are stored in the hosting machine locally. Most user actions are accomplished by triggering AJAX GET requests to store these data at the client side, followed by jQuery scripts that write the data to the webpages on-the-fly.

A limited data version of the website is also hosted at the MathDowsers' code repository[2].

[2] https://kiking0501.github.io/MathDowsers-ARQMath

6.2 Highlighting of Matching Terms

To better inspect and understand Tangent-L's search results, and to provide a rough overview for users to quickly look for query terms that they search for, a highlighting feature is implemented in Tangent-L's *BrushSearch*[3] Web User Interface[4].

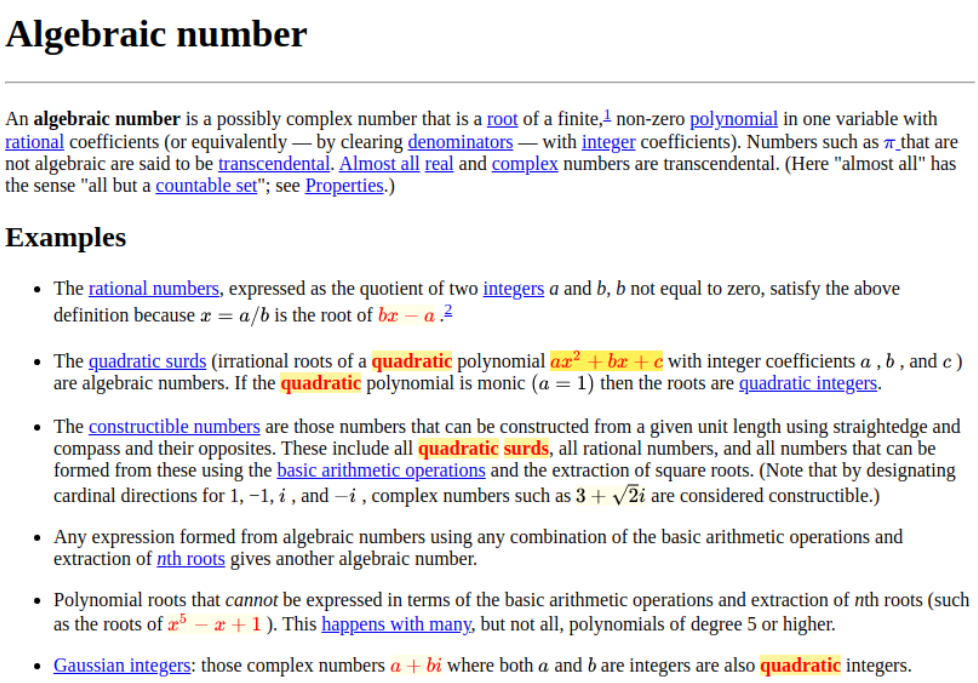
Upon a user search, the results are displayed with highlighting according to the user's input query terms (Figure 23). The highlighting is implemented *approximately*, that is, it does not reflect the true reasoning behind Tangent-L's retrieval logic. Rather, it is an estimation of the retrieval logic that is computed during post-processing. The implementation is outlined as follows:

For Keyword Terms: For each keyword term in the query, highlight the term whenever it appears in the document through pattern matching from regular expressions;

For Formula Terms: For a query formula, at first its math tuples are extracted. Then for each document formula, also extract its math tuples and then compare them against the math tuples from the query. A matching percentage is computed by *inferring* from the math tuples the *number of symbols that are matched*, divided by the total number of symbols from the query formula. As an example, a document formula with $ax^2 + bx + c + d$ will match a query formula $ax^2 + bx + c$ with a 100% matching percentage, while a document formula a will match the same query formula with a 12.5% matching percentage. The matching percentage of the formulas are then reflected in shades of the highlight color, as shown in Figure 24.

The above descriptions reflects the logic that is implemented at the time of writing. To provide a better matching percentage for formulas, the Normalized Formula Similarity (Equation 3.8) can be adopted, in which case, a formula-corpus is pre-indexed and during a user search, document formulas will be highlighted according to its obtained Normalized Formula Similarity from the formula-corpus with respect to the query formula.

[3] https://cs.uwaterloo.ca/brushsearch
[4] http://mathbrush.cs.uwaterloo.ca/



Algebraic number

An **algebraic number** is a possibly complex number that is a root of a finite,[1] non-zero polynomial in one variable with rational coefficients (or equivalently — by clearing denominators — with integer coefficients). Numbers such as π that are not algebraic are said to be transcendental. Almost all real and complex numbers are transcendental. (Here "almost all" has the sense "all but a countable set"; see Properties.)

Examples

- The rational numbers, expressed as the quotient of two integers a and b, b not equal to zero, satisfy the above definition because $x = a/b$ is the root of $bx - a$.[2]
- The quadratic surds (irrational roots of a **quadratic** polynomial $ax^2 + bx + c$ with integer coefficients a , b , and c) are algebraic numbers. If the **quadratic** polynomial is monic ($a = 1$) then the roots are quadratic integers.
- The constructible numbers are those numbers that can be constructed from a given unit length using straightedge and compass and their opposites. These include all **quadratic surds**, all rational numbers, and all numbers that can be formed from these using the basic arithmetic operations and the extraction of square roots. (Note that by designating cardinal directions for 1, −1, i , and $-i$, complex numbers such as $3 + \sqrt{2}i$ are considered constructible.)
- Any expression formed from algebraic numbers using any combination of the basic arithmetic operations and extraction of nth roots gives another algebraic number.
- Polynomial roots that *cannot* be expressed in terms of the basic arithmetic operations and extraction of nth roots (such as the roots of $x^5 - x + 1$). This happens with many, but not all, polynomials of degree 5 or higher.
- Gaussian integers: those complex numbers $a + bi$ where both a and b are integers are also **quadratic** integers.

Figure 23: An highlighted document with respect to the query terms "`quadratic surds`" and "$ax^2 + bx + c$".

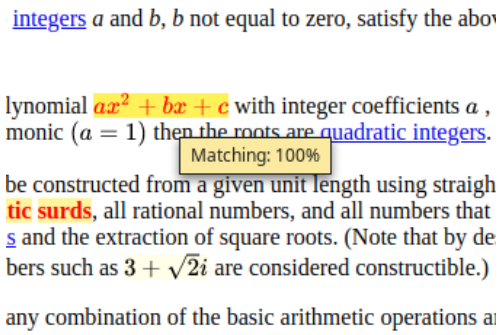


Figure 24: Showing the matching percentage of a formula with respect to a query formula.

Chapter 7

Conclusion and Future Work

This research presents a continuation of work on the math-aware search engine Tangent-L, focusing on the MathCQA application promoted in the ARQMath Lab series using Math StackExchange data.

A three-stage framework—query conversion, math-aware retrieval, and answer re-ranking—for the MathCQA application is proposed and set up for Tangent-L. Various experiments are carried out in each stage to get the best-combined configuration.

With Tangent-L as the cornerstone, the MathCQA task submissions to the ARQMath Lab series achieved the best participant runs with respect to the measures nDCG′ and MAP′. Additionally, Tangent-L performs well in getting relevant answers for formula-dependent questions for all measures, including P′@10.

Simple applications of Tangent-L to the in-context formula-retrieval task are also tested in the second year of the ARQMath Lab. Without much parameter tuning, one of the submissions is found to have comparable effectiveness to the best participant run for the task.

Remarkably, the submissions with Tangent-L have stronger results than many participant systems that have made use of modern NLP models such as word embeddings and BERT-related models. This observation suggests that a traditional MathIR system remains a viable option for the CQA challenges and formula-matching under a real-world scenario.

In the future, two directions might be pursued: either to further improve Tangent-L's internal math-aware search capability or to build a stronger CQA system as an application of Tangent-L. Some ideas are discussed below:

Further study of on math tokens:

Experimental study of repetition tokens has shown a weak effect in improving the search result. In fact, the number of repetitions tokens usually outnumber the number of regular math tokens, which explains the observed small value of its optimal weight in parameter tuning. Perhaps the formulation of repetition tokens might be reviewed to create a smaller number of tokens that capture repeated symbols more effectively.

The number of regular math tokens generated by a single formula is not trivial, although not as much as that of repetition tokens. Perhaps some math tokens can be deemed "stopword" tokens because of their frequent appearance in most formulas. Inspired by the observation from query conversion, maybe a proper removal of stopword math tokens can bring a significant change to the search result as well. It is important to realize that the proximity measure is sensitive to the abundance of math tokens, too, so identifying unnecessary math tokens might bring more insights into the study of ranking by proximity. Similarly, tuning α—the weight of math features—for each individual query might become an easier task with an improved set of math tokens.

A hybrid or integrated CQA system of Tangent-L with other models:

Submissions involving Tangent-L have a stronger performance in formula-dependent questions than other ARQMath participants' systems, showing that it is vital to catch the visual appearance of formulas (with the optional help of keywords matching). Meanwhile, it lacks semantic or even contextualized formula representations to improve its performance in text-dependent questions further. Perhaps results from Tangent-L can be ensembled with results from other retrieval models such as formula embedding models or BERT models to create a complete MathCQA system that performs better for text-dependent questions.

It is also noticeable that the SLT representation is relatively rarely used among ARQMath participants (except for Tangent-CFT2 [27] which uses both the SLT and OPT representation to create formula embeddings), and it is rare to build BERT-models with the SLT representation as well. The LaTeX encoding has been the most common choice among ARQMath participants, while Peng et al. has proposed MathBERT [45] which is a BERT model that takes as input the OPT representation. Such decisions of representations are intuitively understandable since the SLT representation is purely capturing visual information just like the LaTeX, while SLT has been

harder to process. Yet with Tangent-L's good formula-matching performance by using solely information extracted from the SLT representation, perhaps it is worthwhile to spend some effort to incorporate more of the SLT representation, as an example, a BERT model based on tree properties extracted from the SLT.

Improvement of other stages of the three-stage MathCQA framework:
As well as improving the core retrieval model, more techniques might be applied to the overall MathCQA system for further improving the task result, for example, query expansion. Current experiments on query conversion has hinted that limiting the number of keywords selected might easily deteriorate the performance. Therefore, expanding the query with related keywords and formulas might be the direction to improve the query conversion phrase further. For example, given the demonstrated strong signal from question tags towards relevancy, related keywords and formulas from the provided question tags might be added into search queries to improve retrieval over documents that have overlapping tags.

Furthermore, given the ARQMath-1, and now ARQMath-2, benchmark includes relevance judgments, learning-to-rank models are more available for exploration now. Further study might be conducted on the relation between relevance judgments and the proximity signals as well as the rich CQA metadata to build potential features for model learning.

Use of pre-trained CQA data:
To build a transformer model or the BERT model—the latest NLP trends—from scratch that effectively learns the formula language is difficult. Some pre-trained transformers or BERT models related to the CQA settings are available (for example, a pre-trained model for Quora Duplicate Questions Detection[1]) and may be useful as attempted by the DPRL team in ARQMath-2 [27]. While these pre-trained models might not have directly addressed the main CQA task, a new framework that builds on top of these pre-trained models might help save effort in CQA-specific modeling and addressing more of the core "formula-learning" process of the models.

[1] Cross-Encoder for Quora Duplicate Questions Detection

References

[1] Akiko Aizawa and Michael Kohlhase. Mathematical Information Retrieval. In *Information Retrieval Series*, volume 43, pages 169–185. Springer Nature, 2021.

[2] Akiko Aizawa, Michael Kohlhase, and Iadh Ounis. NTCIR-10 Math Pilot Task Overview. In *NTCIR-10*. National Institute of Informatics (NII), 2013.

[3] Akiko Aizawa, Michael Kohlhase, Iadh Ounis, and Moritz Schubotz. NTCIR-11 Math-2 Task Overview. In *NTCIR-11*. National Institute of Informatics (NII), 2014.

[4] Robin Avenoso. Spatial vs. Graph-Based Formula Retrieval. Master's thesis, Rochester Institute of Technology, 2021.

[5] Robin Avenoso, Behrooz Mansouri, and Richard Zanibbi. XY-PHOC Symbol Location Embeddings for Math Formula Retrieval and Autocompletion. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 25–35, 2021.

[6] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text. In *EMNLP-IJCNLP 2019*, pages 3613–3618. Association for Computational Linguistics, 2019.

[7] Andrzej Bialecki, Robert Muir, and Grant Ingersoll. Apache Lucene 4. In *SIGIR 2012*, pages 17–24. University of Otago, Dunedin, New Zealand, 2012.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching Word Vectors with Subword Information. *CoRR*, abs/1607.04606, 2016.

[9] Chris Buckley and Ellen M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR 2004*, pages 25–32. ACM, 2004.

[10] Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR 2009*, pages 758–759. ACM, 2009.

[11] Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay. BERT-Based Embedding Model for Formula Retrieval. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 36–46, 2021.

[12] Kenny Davila and Richard Zanibbi. Layout and Semantics: Combining Representations for Mathematical Formula Search. In *SIGIR 2017*, pages 1165–1168. ACM, 2017.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.

[14] Dallas J. Fraser. Math Information Retrieval using a Text Search Engine. Master's thesis, Cheriton School of Computer Science, University of Waterloo, 2018.

[15] Dallas J. Fraser, Andrew Kane, and Frank Wm. Tompa. Choosing Math Features for BM25 Ranking with Tangent-L. In *DocEng 2018*, pages 17:1–17:10, 2018.

[16] Ferruccio Guidi and Claudio Sacerdoti Coen. A Survey on Retrieval of Mathematical Knowledge. *Mathematics in Computer Science*, 10(4):409–427, 2016.

[17] Mark Hopkins, Ronan Le Bras, Cristian Petrescu-Prahova, Gabriel Stanovsky, Hannaneh Hajishirzi, and Rik Koncel-Kedziorski. SemEval-2019 Task 10: Math Question Answering. In *SemEval-2019*, pages 893–899, June 2019.

[18] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 60(5):493–502, 2004.

[19] Omar Khattab and Matei Zaharia. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In *SIGIR 2020*, pages 39–48. ACM, 2020.

[20] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *CoRR*, abs/1909.11942, 2019.

[21] Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. Pretrained Transformers for Text Ranking: BERT and Beyond. *CoRR*, abs/2010.06467, 2020.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019.

[23] Hans Peter Luhn. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development*, 1(4):309–317, 1957.

[24] Yuanhua Lv and ChengXiang Zhai. Lower-bounding term frequency normalization. In *CIKM'11*, pages 7–16. ACM, 2011.

[25] Behrooz Mansouri, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. Finding Old Answers to New Math Questions: The ARQMath Lab at CLEF 2020. In *ECIR 2020*, volume 12036 of *Lecture Notes in Computer Science*, pages 564–571. Springer, 2020.

[26] Behrooz Mansouri, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. Advancing Math-Aware Search: The ARQMath-2 Lab at CLEF 2021. In *ECIR 2021*, volume 12657 of *Lecture Notes in Computer Science*, pages 631–638. Springer, 2021.

[27] Behrooz Mansouri, Douglas Oard, and Richard Zanibbi. DPRL Systems in the CLEF 2021 ARQMath Lab: Sentence-BERT for Answer Retrieval, Learning-to-Rank for Formula Retrieval. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 47–62, 2021.

[28] Behrooz Mansouri, Douglas W Oard, and Richard Zanibbi. DPRL Systems in the CLEF 2020 ARQMath Lab. volume 2696 of *CEUR Workshop Proceedings*, 2020.

[29] Behrooz Mansouri, Shaurya Rohatgi, Douglas W. Oard, Jian Wu, C. Lee Giles, and Richard Zanibbi. Tangent-CFT: An Embedding Model for Mathematical Formulas. In *ICTIR 2019*, pages 11–18. ACM, 2019.

[30] Behrooz Mansouri, Richard Zanibbi, Douglas W. Oard, and Anurag Agarwal. Overview of ARQMath-2 (2021): Second CLEF Lab on Answer Retrieval for Questions on Math (Working Notes Version). In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 1–24, 2021.

[31] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. In *ICLR 2013*, 2013.

[32] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In *NeurIPS 2013*, pages 3111–3119, 2013.

[33] Bruce R. Miller and Abdou Youssef. Technical Aspects of the Digital Library of Mathematical Functions. volume 38, pages 121–136, 2003.

[34] Robert R Miner, David Carlisle, and Patrick D F Ion. Mathematical Markup Language (MathML) Version 3.0 2nd Edition. W3C recommendation, W3C, April 2014.

[35] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. SemEval-2017 Task 3: Community Question Answering. In *SemEval-2017*, pages 27–48, Dec 2018.

[36] Preslav Nakov, Lluís Màrquez, Walid Magdy, Alessandro Moschitti, Jim Glass, and Bilal Randeree.

[37] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, James Glass, and Bilal Randeree. SemEval-2016 Task 3: Community Question Answering. In *SemEval-2016*, pages 525–545, 2016.

[38] Yin Ki Ng, Dallas Fraser, Besat Kassaie, and Frank Tompa. Dowsing for Answers to Math Questions: Ongoing Viability of Traditional MathIR. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 63–81, 2021.

[39] Yin Ki Ng, Dallas J. Fraser, Besat Kassaie, and Frank Wm. Tompa. Dowsing for Math Answers. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 12th International Conference of the CLEF Association, CLEF 2021*, volume 12880 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2021.

[40] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. In *Coco@NIPS 2016*, volume 1773 of *CEUR Workshop Proceedings*, 2016.

[41] Vít Novotný, Michal Štefánik, Dávid Lupták, Martin Geletka, Petr Zelina, and Petr Sojka. Ensembling Math Information Retrieval Systems: MIRMU and MSM at ARQMath 2021. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 82–106, 2021.

[42] María-Dolores Olvera-Lobo and Juncal Gutiérrez-Artacho. Question Answering Track Evaluation in TREC, CLEF and NTCIR. In *WorldCIST 2015*, volume 353 of *Advances in Intelligent Systems and Computing*, pages 13–22. Springer, 2015.

[43] Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier Information Retrieval Platform. In *ECIR 2005*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer, 2005.

[44] Barun Patra. A survey of Community Question Answering. *CoRR*, abs/1705.04009, 2017.

[45] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding. *CoRR*, abs/2105.00377, 2021.

[46] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *EMNLP 2014*, pages 1532–1543. ACL, 2014.

[47] Deanna C. Pineau. Math-Aware Search Engines: Physics Applications and Overview. *CoRR*, abs/1609.03457, 2016.

[48] Yves Rasolofo and Jacques Savoy. Term Proximity Scoring for Keyword-Based Retrieval Systems. In *ECIR 2003*, volume 2633, pages 207–218. Springer, 2003.

[49] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *EMNLP-IJCNLP 2019*, pages 3980–3990. Association for Computational Linguistics, 2019.

[50] Anja Reusch, Maik Thiele, and Wolfgang Lehner. TU_DBS in the ARQMath Lab 2021, CLEF. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 107–124.

[51] Stephen Robertson. The Probability Ranking Principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[52] Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In *TREC 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST), 1994.

[53] Stephen E. Robertson and Hugo Zaragoza. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

[54] Shaurya Rohatgi, Jian Wu, and C Lee Giles. Ranked List Fusion and Re-ranking with Pre-trained Transformers for ARQMath Lab. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 125–132.

[55] Tetsuya Sakai and Noriko Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008.

[56] Ivan Srba and Mária Bieliková. A Comprehensive Survey and Classification of Approaches for Community Question Answering. In *ACM Transactions on the Web*, volume 10, pages 18:1–18:63, 2016.

[57] Tao Tao and ChengXiang Zhai. An exploration of proximity measures in information retrieval. In *SIGIR 2017*, pages 295–302, 2007.

[58] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS 2017*, pages 5998–6008, 2017.

[59] Peilin Yang, Hui Fang, and Jimmy Lin. Anserini: Enabling the Use of Lucene for Information Retrieval Research. In *SIGIR 2017*, pages 1253–1256. ACM, 2017.

[60] Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topic, and Kenny Davila. NTCIR-12 MathIR Task Overview. In *NTCIR-12*. National Institute of Informatics (NII), 2016.

[61] Richard Zanibbi, Kenny Davila, Andrew Kane, and Frank Wm. Tompa. Multi-Stage Math Formula Search: Using Appearance-Based Similarity Metrics at Scale. In *SIGIR 2016*, pages 145–154. ACM, 2016.

[62] Richard Zanibbi, Douglas W. Oard, Anurag Agarwal, and Behrooz Mansouri. Overview of ARQMath 2020 (Updated Working Notes Version): CLEF Lab on Answer Retrieval for Questions on Math. In *CLEF 2020*, volume 2696 of *CEUR Workshop Proceedings*, 2020.

[63] Wei Zhong, Shaurya Rohatgi, Jian Wu, C. Lee Giles, and Richard Zanibbi. Accelerating Substructure Similarity Search for Formula Retrieval. In *ECIR 2020*, volume 12035 of *Lecture Notes in Computer Science*, pages 714–727. Springer, 2020.

[64] Wei Zhong and Richard Zanibbi. Structural Similarity Search for Formulas Using Leaf-Root Paths in Operator Subtrees. In *ECIR 2019*, volume 11437 of *Lecture Notes in Computer Science*, pages 116–129. Springer, 2019.

[65] Wei Zhong, Xinyu Zhang, Ji Xin, Jimmy Lin, and Richard Zanibbi. Approach Zero and Anserini at the CLEF-2021 ARQMath Track: Applying Substructure Search and BM25 on Operator Tree Path Tokens. In *CLEF 2021*, volume 2936 of *CEUR Workshop Proceedings*, pages 133–156.

APPENDICES

Appendix A

The ARQMath Lab Official Results

The official results announced in the worknote papers by the Lab organizers.

A.1 The MathCQA Task in ARQMath-1

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 Type		Evaluation Measures		
Run	Data	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

A.2 The MathCQA Task in ARQMath-2

Table 4
ARQMath-2 Task 1 (CQA) results. **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.

		Run Type		ARQMath-1 77 Topics			ARQMath-2 71 Topics		
Run	Data	P	M	nDCG′	MAP′	P′@10	nDCG′	MAP′	P′@10
Baselines									
Linked MSE posts	n/a	(✓)		(0.279)	(0.194)	(**0.386**)	0.203	0.120	**(0.282)**
TF-IDF + Tangent-S	Both	(✓)		0.248	0.047	0.073	0.201	0.045	0.086
TF-IDF	Both			0.204	0.049	0.074	0.185	0.046	0.063
Tangent-S	Math			0.158	0.033	0.051	0.111	0.027	0.052
MathDowsers									
primary	Both	✓		**0.433**	0.191	0.249	**0.434**	**0.169**	0.211
proximityReRank	Both			0.373	0.117	0.131	0.335	0.081	0.049
DPRL									
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
Math Stack Exchange	Both			0.409	0.232	0.322	0.323	0.083	0.078
TU_DBS									
TU_DBS_P	Both	✓		0.380	0.198	0.316	0.377	0.158	**0.227**
TU_DBS_A2	Both			0.356	0.173	0.291	0.367	0.147	0.217
TU_DBS_A3	Both			0.359	0.173	0.299	0.357	0.141	0.194
TU_DBS_A1	Both			0.362	0.178	0.304	0.353	0.132	0.180
TU_DBS_A4	Both			0.045	0.016	0.071	0.028	0.004	0.009
Approach0									
B60	Both		✓	0.364	0.173	0.256	0.351	0.137	0.189
B60RM3	Both		✓	0.360	0.168	0.252	0.349	0.137	0.192
B55	Both	✓	✓	0.364	0.173	0.251	0.344	0.135	0.180
A55	Both		✓	0.364	0.171	0.256	0.343	0.134	0.194
P50	Both		✓	0.361	0.171	0.255	0.327	0.122	0.155
MIRMU									
WIBC	Both			0.381	0.135	0.161	0.332	0.087	0.106
RBC	Both	✓		0.392	0.153	0.220	0.322	0.088	0.132
IBC	Both			0.338	0.114	0.153	0.286	0.073	0.117
CompuBERT	Both			0.304	0.114	0.207	0.262	0.083	0.135
SCM	Both			0.324	0.119	0.156	0.250	0.059	0.072
MSM									
MG	Both	✓		0.310	0.114	0.170	0.278	0.077	0.127
PZ	Both			0.336	0.126	0.181	0.275	0.085	0.124
MP	Both			0.203	0.059	0.094	0.154	0.036	0.047
MH	Both			0.184	0.057	0.108	0.131	0.028	0.037
LM	Both			0.178	0.058	0.107	0.128	0.029	0.048
PSU									
PSU	Both	✓		0.317	0.116	0.165	0.242	0.065	0.110
GoogolFuel									
2020S41R71	Both	✓		0.292	0.086	0.153	0.203	0.050	0.092
2020S41R81	Both			0.290	0.085	0.153	0.203	0.050	0.089
2020S41R91	Both			0.289	0.084	0.157	0.203	0.050	0.089
2020S51R71	Both			0.288	0.082	0.140	0.202	0.049	0.089
2020S41	Both			0.281	0.076	0.135	0.201	0.048	0.080
BetterThanG									
Combiner1vs1	Both	✓	✓	0.233	0.046	0.073	0.157	0.031	0.051
Combiner2vs1	Both		✓	0.229	0.044	0.069	0.153	0.030	0.054
CombinerNorm	Both		✓	0.215	0.045	0.073	0.141	0.026	0.042
LuceneBM25	Text			0.179	0.052	0.079	0.119	0.025	0.032
Tangent-S	Math			0.158	0.033	0.051	0.110	0.026	0.061

A.3 In-Context Formula Retrieval in ARQMath-1

A.3.1 Official Result in ARQMath-1

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

A.3.2 Official Result in ARQMath-1 and Re-evaluation during ARQMath-2

		ARQMath-1 Official			*Unofficial Re-evaluation*		
		nDCG′	MAP′†	P′@10†	nDCG′	MAP′†	P′@10†
Baselines							
Tangent-S	¶	**(0.506)**	**(0.288)**	**(0.478)**	**(0.691)**	**(0.446)**	**(0.453)**
DPRL							
TangentCFTED	¶	**0.420**	**0.258**	**0.502**	**0.563**	**0.388**	**0.436**
TangentCFT		0.392	0.219	0.396	0.527	0.334	0.349
TangentCFT+		0.135	0.047	0.207	0.225	0.106	0.211
MIRMU							
SCM		0.119	0.056	0.058	0.132	0.063	0.076
Formula2Vec	¶	0.108	0.047	0.076	0.126	0.055	0.076
Ensemble		0.100	0.033	0.051	0.118	0.041	0.053
Formula2Vec		0.077	0.028	0.044	0.095	0.034	0.042
SCM	¶	0.059	0.018	0.049	0.068	0.022	0.049
NLP_NITS							
formulaembedding	¶	0.026	0.005	0.042	0.058	0.017	0.040

¶ primary run † using H+M binarization

A.4 In-Context Formula Retrieval in ARQMath-2

Table 5
ARQMath-2 Task 2 (Formula Retrieval) results, computed over 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.

				ARQMath-1 45 Topics			ARQMath-2 58 Topics		
		Run Type							
Run	Data	P	M	nDCG′	MAP′	P′@10	nDCG′	MAP′	P′@10
Baseline									
Tangent-S	Math	(✓)		(0.692)	(0.446)	(0.453)	(0.492)	(0.272)	(0.419)
Approach0									
P300	Math		✓	0.507	0.342	0.441	**0.555**	**0.361**	**0.488**
B	Math		✓	0.493	0.340	0.425	0.519	0.336	0.461
B30	Math		✓	0.527	0.358	0.446	0.516	0.295	0.393
C30	Math		✓	0.527	0.358	0.446	0.516	0.295	0.393
P30	Math	✓	✓	0.527	0.358	0.446	0.505	0.284	0.371
MathDowsers									
formulaBase	Both	✓		0.562	0.370	0.447	0.552	0.333	0.450
docBase	Both			0.404	0.251	0.386	0.433	0.257	0.359
XY-PHOC-DPRL									
XY-PHOC	Math	✓		0.611	0.423	0.478	0.548	0.323	0.433
DPRL									
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
TangentCFT2-TED	Math			0.648	0.480	0.502	0.410	0.253	0.464
TangentCFT-2	Math			0.607	0.437	0.480	0.338	0.188	0.297
TU_DBS									
TU_DBS_A3	Math			0.426	0.298	0.386	-	-	-
TU_DBS_A1	Math			0.396	0.271	0.391	-	-	-
TU_DBS_A2	Math			0.157	0.085	0.122	0.154	0.071	0.217
TU_DBS_P	Both	✓		0.152	0.080	0.122	0.153	0.069	0.216
NLP_NITS									
FormulaEmbedding_P	Math	✓		0.233	0.140	0.271	0.161	0.059	0.197
FormulaEmbedding_A	Math			-	-	-	0.114	0.039	0.152
Baseline	Math			-	-	-	0.091	0.032	0.151

Appendix B

The ARQMath Lab Resources

The ARQMath Lab resources shared by the Lab organizers in the ARQMath Forum[1] for reference.

[1] https://groups.google.com/g/arqmath-lab/

B.1 Manually-selected Keywords and Formulas for ARQMath-1 Topics

Topic	List of formulas and keywords
A.1	[range, of, rational, function, $f(x) = \frac{x^2+x+c}{x^2+2x+c}$]
A.2	[differential, equations, $f'(x) = f(x+1)$]
A.3	[bisection, algorithm, $\sqrt{5}$]
A.4	[combinatoric, sum, $\sum_{k=0}^{n} \binom{n}{k} k$]
A.5	[conditional, probability, formula, $P((2)\|(1)) = \frac{P((2)\cap(1))}{P((1))} = \frac{P((2))P((1))}{P((1))} = P((2))$]
A.6	[number, mod, with, big, exponent, $5^{133} \mod 8$.]
A.7	[remainder, using, modulus, $11^{10} - 1 = x \pmod{100}$]
A.8	[finding, value, of, limit, $\lim_{n\to\infty} \sqrt[n]{\frac{(27)^n (n!)^3}{(3n)!}}$]
A.9	[simplifying, series, $\sum_{n=0}^{N} nx^n$]
A.10	[integral, converges, $\int_0^\infty \frac{\sin x}{x^a}$]
A.11	[cross, product, $u \times v = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a & b & c \\ d & e & f \end{vmatrix}$]
A.12	[roots, of, a, complex, number, $(1 + i\sqrt{3})^{1/2}$]
A.13	[simplify, expression, $\int_a^b f(x)dx + \int_{f(a)}^{f(b)} f^{-1}(x)dx$]
A.14	[first-order, differential, equation, $y = xy' + \frac{1}{2}(y')^2$]
A.15	[derive, the, sum, $\sum_{i=1}^{n} ix^{i-1}$]
A.16	[compute, integral, $\int_0^1 \frac{\ln(1+x)\ln(1-x)}{1+x} dx$]
A.17	[calculate, function, $\int_{x=0}^{\infty} \frac{\sin(x)}{x}$, $\frac{e^{iz}}{z}$]
A.18	[Cesáro-Stolz, theorem, $\lim_{n\to\infty} \frac{[(n+1)(n+2)\cdots(n+n)]^{1/n}}{n}$]
A.19	[greatest, common, factor, $p^4 - 1$]
A.20	[euler-Totient, Function, $\phi(n) = 40$]
A.21	[finding, the, last, two, digits, $9^{9^{\cdot^{\cdot^{\cdot^9}}}}$]
A.22	[find, number, of, $d'sd, d+1, d+2... = N$]
A.23	[find, the, lcm, $2^7 \cdot 3^8 \cdot 5^2 \cdot 7^{11}$, $2^3 \cdot 3^4 \cdot 5$]
A.24	[evaluate, $\sqrt{2i-1}$]
A.25	[polynomial, $P(x^2+1) = (P(x))^2 + 1$, $P(x^2+1) = (P(x))^2 + 1$]
A.26	[ndefinite, integral, using, Taylor, series, $\int_0^\infty \frac{\sin x}{x} dx$.]
A.27	[solving, for, the, value, $e^{3i\pi/2}$]
A.28	[right, triangle, $\sin(18) = \frac{a+\sqrt{b}}{c}$]
A.29	[dividing, Complex, Numbers, by, Infinity, $\frac{5i}{\infty} = 0$]
A.30	[binomial, theorem, $a^3 + b^3 + c^3 - 3abc$]
A.32	[are, definitions, axioms, ,Empty(x) $\iff \nexists y(y \in x)$]
A.33	[physical, meaning, of, third, derivative, $\frac{\partial^3 f}{\partial x^3}, \frac{\partial^3 f}{\partial t^3}$]
A.34	[Knuth, up-arrow, notation,

$$a \uparrow b = a^b$$
$$a \uparrow^n b = \underbrace{a \uparrow^{n-1} (a \uparrow^{n-1} (\ldots (a \uparrow^{n-1} a) \ldots))}_{b \text{ copies of } a} \text{]}$$

A.35 [function, not, have, an, antiderivative, $\int e^{x^2} dx$, $\int e^{2x} dx$]

A.36 [proof, by, contradiction, $\neg P \rightarrow A_1 \rightarrow \ \ldots \ \rightarrow A_n \rightarrow P$]

A.37 [real-valued, functions, $f \circ g = g \circ f$]

A.38 [the, axiom, of, choice, $q, r : a = bq + r$]

A.39 [which, value, is, bigger, $\log 2019 \log 2018$]

A.40 [linear, equation, meaning, $a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_nx_n =$]

A.41 [find, number, of, onto, functions, $\sum_{r=1}^{n} (-1)^{(n-r)} \binom{n}{r} (r)^m$]

A.42 [emergence, of, complex, numbers, $x^2 + 1 = 0$, $\sqrt{-1}$]

A.43 [prove, with, Fourier, series, $\sum_{n \geq 1} \frac{1}{n^2+1} = \frac{\pi \coth \pi - 1}{2}$]

A.44 [infinite, order, $A, B \in M_{2\times 2}(\mathbb{Q})$]

A.45 [prove, independent, in, R, $\sin(x), \sin(2x), \sin(3x), ..., \sin(nx)$]

A.46 [lebesgue, integrable, function, $\int x^k f(x) dx = 0$]

A.47 [Wilson's, Theorem, $rq \equiv 1 \bmod p$]

A.48 [inequality, proof, $(x + y)^k \geq x^k + y^k$]

A.49 [combinatoric, interpretation, identity, $\binom{2n}{n} = \sum_{k=0}^{n} \binom{n}{k}^2$]

A.50 [divergent, series, $\sum \frac{1}{n^{2+\cos n}}$]

A.51 [Sum, of, series, binomial, coefficients, $\sum_{r=0}^{n} \binom{n+r}{r} \frac{1}{2^r} = 2^n$]

A.52 [contradiction, proof, prime, number, $n_i \nmid n = n_1 n_2 ... n_k + 1$]

A.53 [inverse, of, a, square, matrix, $AB = 1 \Rightarrow BA = 1$]

A.54 [diagonal, argument, for, uncountable, powerset, $P(N) = (S|SN)$]

A.55 [mistake, in, calculation, $\frac{1}{\sqrt{-1}} = \sqrt{-1}$]

A.56 [logical, formula, involving, prime, numbers, $\exists p \ (p \text{ is prime } \rightarrow \forall x \ (x \text{ is prime}))$]

A.57 [continuous, one-to-one, function, $f^{-1} : f(X) \mapsto X$]

A.58 [trigonometric, functions, proof, $3 \arcsin \frac{1}{4} + \arccos \frac{11}{16} = \frac{\pi}{2}$]

A.59 [Euler's, totient, proof, $\sum_{d|n} \phi(d) = n$]

A.60 [limiting, value, of, a, sequence, $a_n = \left(1 - \frac{1}{\sqrt{2}}\right) ... \left(1 - \frac{1}{\sqrt{n+1}}\right)$]

A.61 [equation, proof, $n = 3i + 5j, \ n \geq 8$]

A.62 [rational, integer, numbers, set, cardinality, $|\mathbb{Q}| = |\mathbb{Z}|$]

A.63 [gcd, and, lcm, relationship, $\operatorname{lcm}(n_1, n_2) = \frac{n_1 n_2}{\gcd(n_1, n_2)}$]

A.64 [Intermediate, Value, Theorem, $f([a, b]) \subset [a, b]$]

A.65 [show, inequality, proof, $e^{-2\lambda t} \lambda^2 \leq \frac{1}{e^2 t^2}$]

A.66 [justify, equation, $(x^T A h)^T = h^T A^T x$]

A.67 [combination, of, matrixes, $\det \begin{bmatrix} A & B \\ O & C \end{bmatrix} = \det(A) \det(C)$]

A.68 [prove, divisible, $a^n + 1$]

A.69 [induction, with, two, variable, parameters, $\binom{s}{s} + \binom{s+1}{s} + \ldots + \binom{n}{s} = \binom{n+1}{s+1}$]

A.70 [Euler, formula, and, geometric, serie, $\sum_{j=0}^{N-1} \cos \frac{(2j+1)\pi}{2N} = 0$]

A.71 [proof, with, induction, $1^2 + 2^2 + ... + n^2 = \frac{n(n+1)(2n+1)}{6}$]

A.72 [set, equality, $\mathcal{X} = \mathcal{Y}, \ \mathcal{X} \in \mathcal{Y}$]

A.73 [binomial, theorem, proof, $\binom{n}{0}^2 + \binom{n}{1}^2 + ... + \binom{n}{n}^2 = \binom{2n}{n}$]

A.74 [image, of, function, interval, $f(x) = x + \dfrac{1}{x},\ f : (0, \infty) \to \mathbb{R}$]

A.75 [limit, proof, $\lim_{u\to\infty} \frac{u^m}{e^u} = 0$]

A.76 [covering, Z, by, arithmetic, progressions, $\bigcup_{i\in\mathbb{N}}(a_i + b_i\mathbb{Z}) = \mathbb{Z}$]

A.77 [distributive, law, $(-1)(-1) = 1$]

A.79 [inequality, with, complex, exponential, $\left|\dfrac{e^{-ixu} - 1}{u}\right| \leq |x|$]

A.80 [set, finite, subsets, is, countable, $\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \{4\}, \ldots$]

A.81 [infinite, set, contains, countable, subset, $\{1, \cdots, n\}$]

A.82 [definite, integrals, evaluate, to, 0, $A = \displaystyle\int_0^{2\pi} f(x)\mathrm{d}x$]

A.83 [sequence, sums, inverse, natural, numbers, $1, 1 + \frac{1}{2}, 1 + \frac{1}{2} + \frac{1}{3}, 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4}, ...$]

A.84 [principal, ideals, $Z[x]$]

A.85 [bug, steps, to, reach, $Np_n = \frac{1}{2}p_{n-1}$]

A.86 [proof, $\sum_{k=0}^{n} k \cdot \begin{pmatrix} n \\ k \end{pmatrix} = O\left(2^{n \log_3 n}\right)$]

A.87 [proof, $\forall n \in \mathbb{N} : (\sum_{i=1}^{n} a_i)(\sum_{i=1}^{n} \frac{1}{a_i}) \geq n^2$]

A.88 [reducible, polynomial, $x^4 + 10x^2 + 1$]

A.89 [parametrization, of, pythagorean-like, equation, $\mathrm{A}^2 + B^2 = C^2 + D^2$]

A.90 [definition, of, an, Inverse, matrix, $A^{-1}A = \mathbb{I}_n \quad \wedge \quad AA^{-1} = \mathbb{I}_n \quad (1)$]

A.91 [function, reaches, value, $2times\mathbb{R} \to \mathbb{R}$]

A.92 [principal, maximal, ideal, $\mathbb{F}_q[X, Y]$]

A.93 [characteristic, polynomial, $det(xI - AB) = det(xI - BA)$]

A.94 [natural, numbers, sets, finite, intersection, $2^{\aleph_0}$]

A.95 [inequality, about, lengths, of, bounds, $\lim_{x\to 0} \frac{\sin x}{x} = 1$]

A.96 [convergent, sum, $\sum_i \frac{a_i}{a_i + a_{i+1} + a_{i+2} + \cdots}$]

A.97 [R, dimension, in, vector, space, $\mathbb{R}$]

A.98 [irrational, rotation, and, dense, set, $R^n(e^{2\pi ix}) = e^{2\pi i(x+n\alpha)}$]

A.99 [set, of, continuities, of, function, $f : \mathbb{R} \to \mathbb{R}$]

A.100 [closed, set, $E = (0, 1]$]

Appendix C

Word Lists for Search Queries

Example word lists built for keyword extraction during Search Query Conversion (Section 4.1).

C.1 Top-50 Most Common Words from the MSE Tags

The top-50 most common words built from MSE tags. The count of each word is the count of its tag appearing in the MSE collection, or the sum of counts of tags that contain this word. For example, the word "theory" is the sum of counts of tags "group-theory", "number-theory", "probability-theory", etc.

Count	*Word*	*Stemmed Word*	*Count*	*Word*	*Stemmed Word*
277847	theory	theori	38275	general	gener
212782	analysis	analysi	36058	functional	function
198991	algebra	algebra	33771	numbers	number
129851	calculus	calculu	33363	precalculus	precalculu
116095	probability	probabl	31312	limits	limit
99186	geometry	geometri	30157	ordinary	ordinari
98427	linear	linear	29735	measure	measur
96792	real	real	27506	set	set
65173	number	number	27140	mathematics	mathemat
58843	differential	differenti	26158	statistics	statist
58478	series	seri	26033	groups	group
58446	abstract	abstract	25623	integrals	integr
58185	topology	topolog	24870	logic	logic
54582	integration	integr	23214	multivariable	multivari
52519	complex	complex	23184	discrete	discret
47535	equations	equat	22695	polynomials	polynomi
47494	spaces	space	22420	verification	verif
46469	elementary	elementari	21739	inequality	inequ
45891	sequences	sequenc	21735	optimization	optim
45010	functions	function	21347	derivatives	deriv
40864	algebraic	algebra	20906	trigonometry	trigonometri
40166	group	group	20616	vector	vector
39624	matrices	matric	19677	convergence	converg
39522	proof	proof	17753	graph	graph
39335	combinatorics	combinator	17585	distributions	distribut

C.2 Top-50 Most Common Words from NTCIR MathIR Wikipedia Articles Titles

The top-50 most common words built from around 32,000 of NTCIR MathIR Wikipedia Article titles. The count of each word is its count appearing as a whole or as part of the article titles.

Count	*Word*	*Stemmed Word*	*Count*	*Word*	*Stemmed Word*
986	theorem	theorem	189	law	law
627	theory	theori	189	system	system
569	function	function	188	template	templat
501	synthase	synthas	188	list	list
500	number	number	180	analysis	analysi
487	dehydrogenase	dehydrogenas	176	alpha	alpha
431	equation	equat	172	index	index
425	model	model	165	phosphate	phosphat
309	method	method	158	inequality	inequ
300	space	space	156	coa	coa
296	group	group	153	geometry	geometri
284	algebra	algebra	149	kinase	kinas
274	reductase	reductas	147	monooxygenase	monooxygenas
268	mathematics	mathemat	146	file	file
266	distribution	distribut	145	linear	linear
260	algorithm	algorithm	142	point	point
259	problem	problem	141	test	test
242	matrix	matrix	141	conjecture	conjectur
233	graph	graph	138	vector	vector
219	methyltransferase	methyltransferas	137	energy	energi
215	beta	beta	136	time	time
214	set	set	130	operator	oper
201	quantum	quantum	129	surface	surfac
197	field	field	128	papyrus	papyru
193	formula	formula	127	ring	ring

Appendix D

Optimal α values for Individual Topics of Different Dependencies

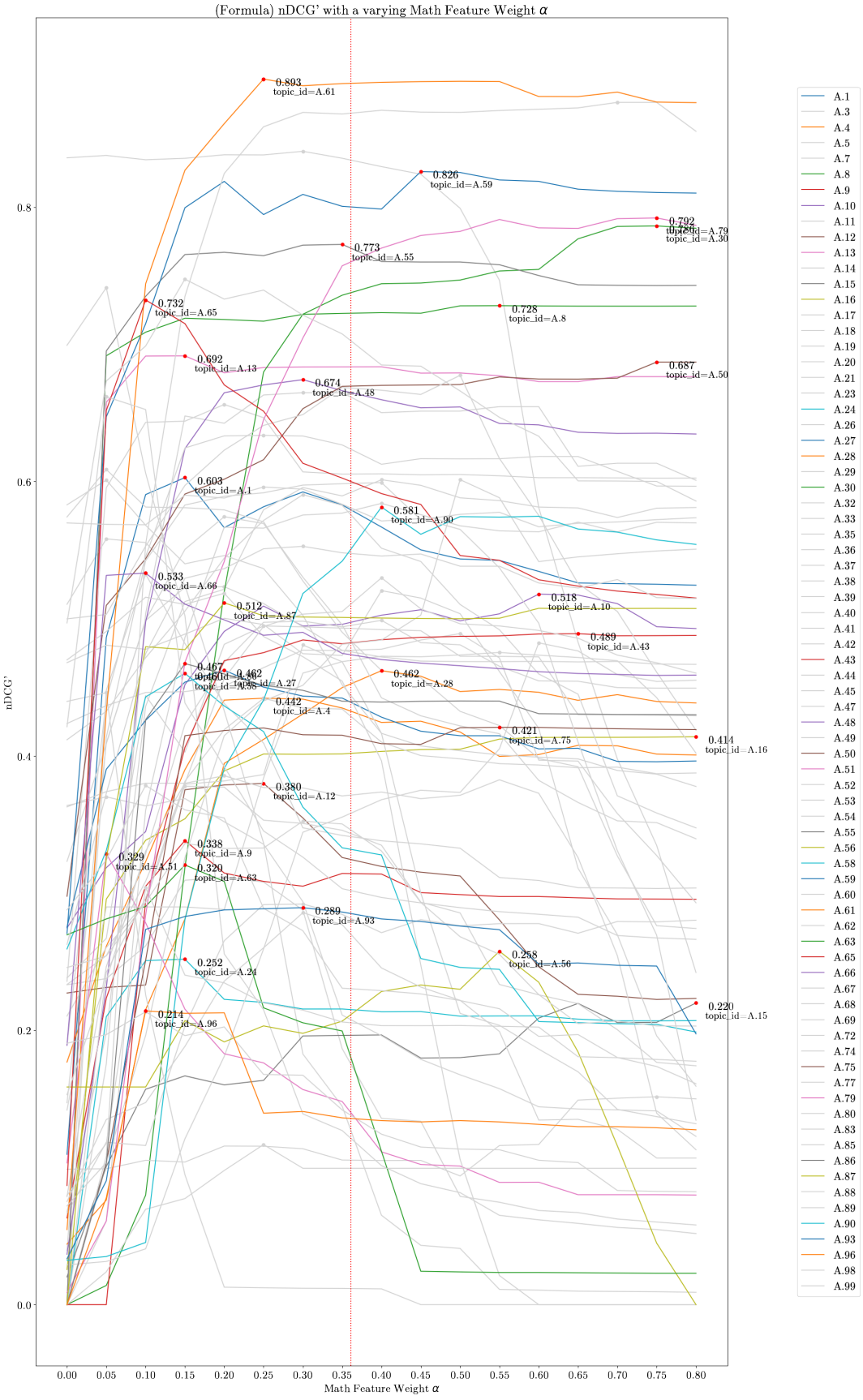


Figure 25: ARQMath-1 Evaluation on individual formula-dependent topics with a varying $\alpha : 0 \leq \alpha \leq 0.8$, with a step size of 0.05.

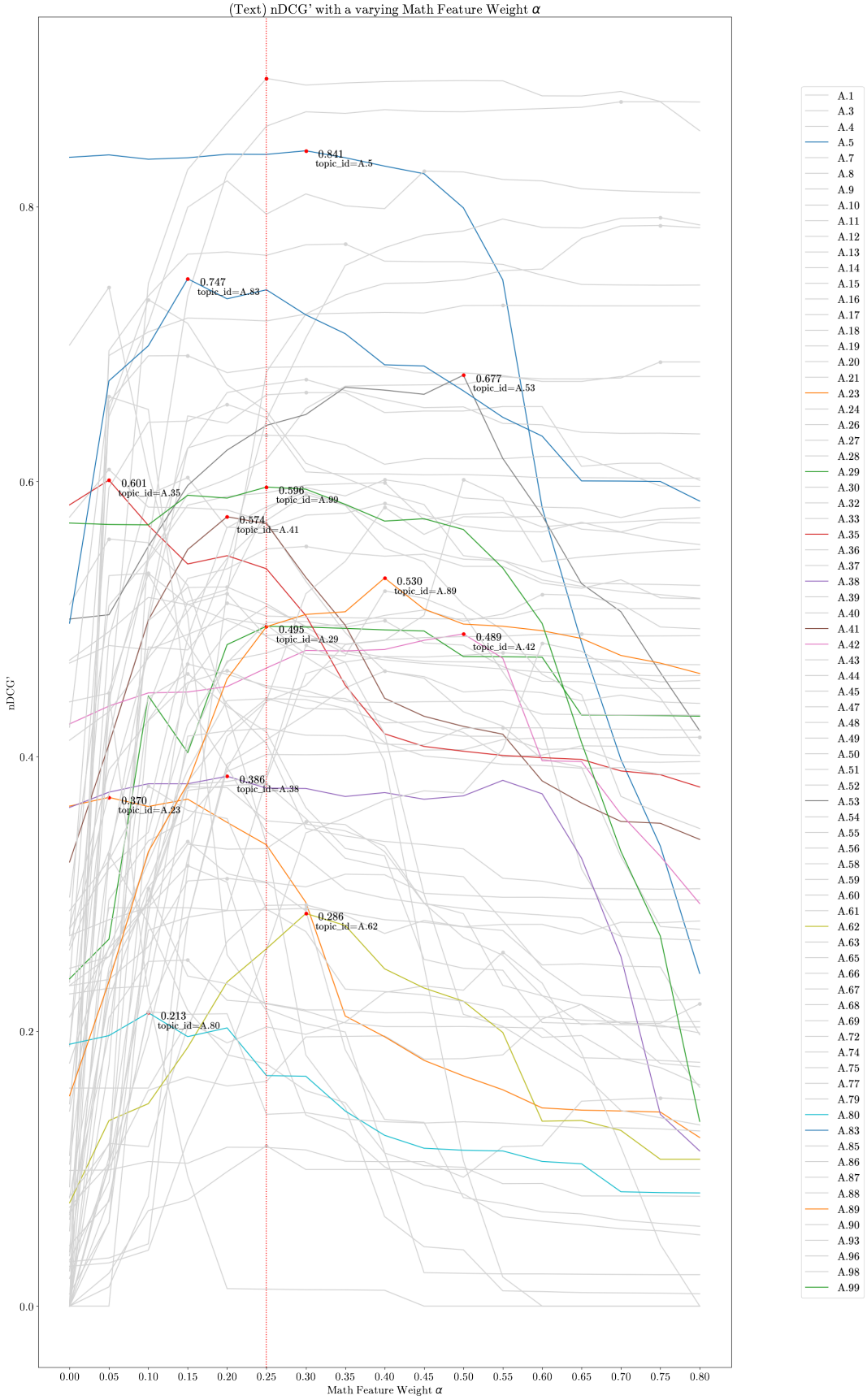


Figure 26: ARQMath-1 Evaluation on individual text-dependent topics with a varying $\alpha : 0 \leq \alpha \leq 0.8$, with a step size of 0.05.

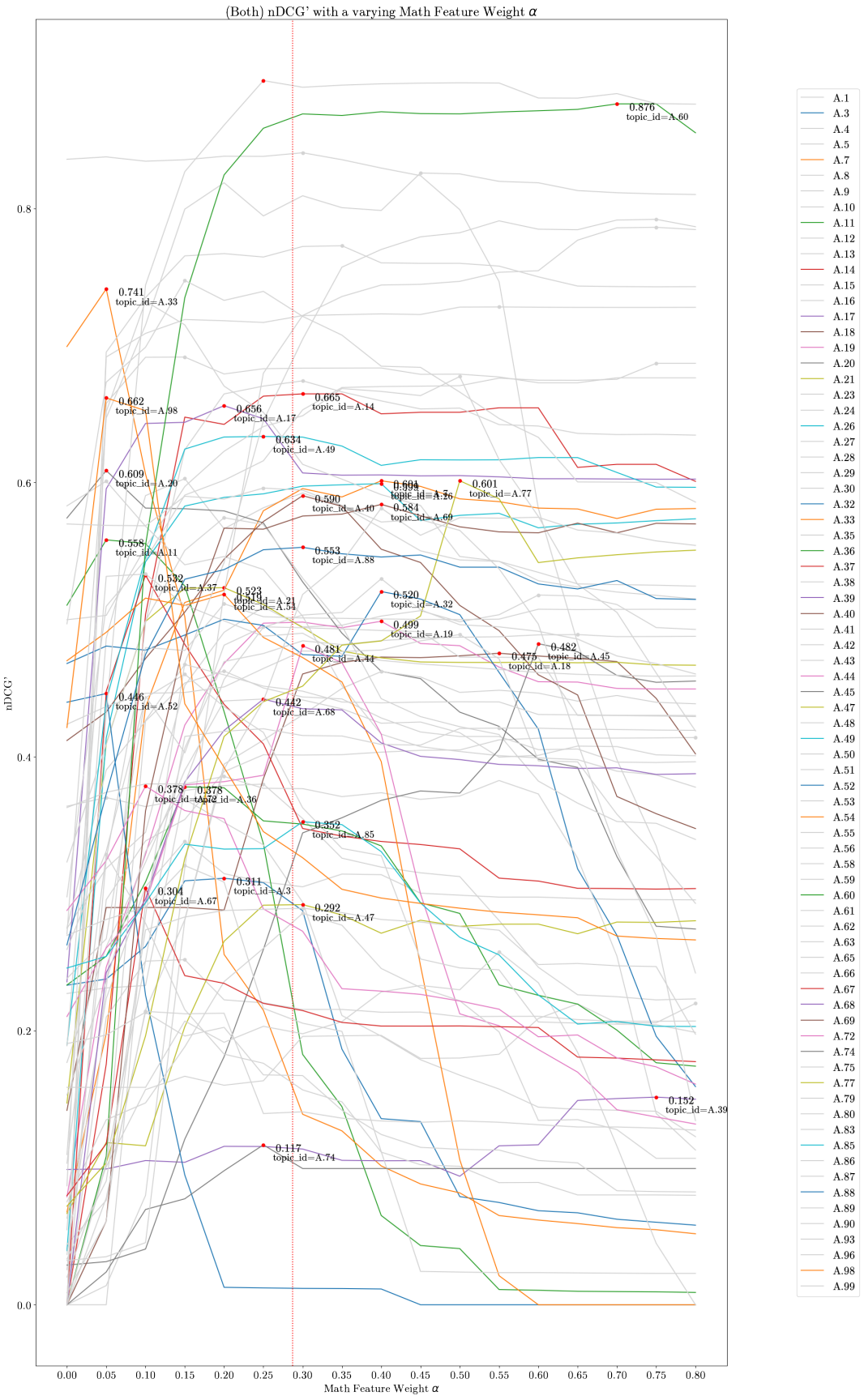


Figure 27: ARQMath-1 Evaluation on individual both-dependent topics with a varying $\alpha : 0 \leq \alpha \leq 0.8$, with a step size of 0.05.

Appendix E

Conclusions from MathDowsers' Working Notes in the MathCQA Task

The following sections discuss the performance of the submitted runs for the MathCQA task in ARQMath-1 and ARQMath-2. Full details can be found in the published CLEF Working Notes [62] and [30].

E.1 ARQMath-1 Submission Runs

As a first attempt at the MathCQA challenge, the submitted runs for ARQMath-1 were designed to address the following research objectives:

RQ1 What is an effective way to convert each mathematical question (expressed in mathematical natural language) into a formal query consisting of keywords and formulas?

RQ2 Should keywords or math formulas be assigned heavier weights in a query?

RQ3 What is the effect of a re-ranking algorithm that makes use of metadata?

A primary run and four alternate runs were submitted in total. The primary run was designed as a combination of hypotheses for presumably "best" configuration at that time for all research objectives. The alternate runs were designed to test these hypotheses: the

setup for each of them is the same as the primary run, except for a single aspect that is associated with a testing hypothesis. The runs are as follows:

alpha05: The primary run, with $\text{uniqueQuery}_{\text{MSEWiki}_{\text{FF}}}$ adopted to create search queries; the target corpus to be $\text{Corpus}_{\text{QAPair}}$; the math feature weight α during query time to be 0.5 for the vanilla Tangent-L, that is, math terms were given half the weight of keywords in a query; followed by a re-ranking using $\text{reRank}_{\text{LRM}}$.

alpha05-trans: An alternative run where $\text{Query}_{\text{manual}}$ was adopted instead to create search queries. This is to compare the effectiveness of the query conversion algorithm from Section 4.1 against human understanding of the questions. It is also the only manual run submitted among all runs.

alpha02: An alternative run with α set to be 0.2 instead, that is, math terms were only given one-fifth of the weight of keywords in a query.

alpha10: An alternative run with α set to be 1.0 instead, that is, math terms were given equal weight with keywords in a query.

alpha05-noR: An alternative run with no re-ranking, that is, answers were ranked by Tangent-L's internal ranking only.

Regarding the primary measure nDCG′ in Table 27, the first observation is that the primary run (*alpha05*) performs better than only one of the alternate runs, which hints that its setting has not been optimal as assumed. The second observation is that the alternate run using manually extracted formulas and keywords performs better (0.298 by *alpha05-trans*) than the primary run (0.278 by *alpha05-trans*) when α is set to 0.5. Thirdly, lowering the weight placed on math terms improves the performance (0.301 when $\alpha = 0.2$), and using a higher weight hurts the performance (0.267 when $\alpha = 1.0$). Finally, the alternative run without re-ranking achieves the best performance (0.345 by *alpha05-noR*) and it is also the best participant run in ARQMath-1.

These observations motivate later experiments for the effect of query conversion and the effect of α (Section 4.4.2), and also for the effect of re-ranking with CQA metadata (Section 4.4.8 and Table 35). The result of those experiments provide answers to the original research objectives: that the proposed query conversion approach together with an optimal α is an effective algorithm (**RQ1**); and the optimal α is small, which means keywords should be assigned heavier weights in a query (**RQ2**); and that the re-ranking design using the CQA metadata has been detrimental to the performance (**RQ3**).

vanillaTL	*reRank*$_{LRM}$	*No reRank*$_{LRM}$
$\alpha = 1.0$	(alpha10) 0.267	0.327
$\alpha = 0.5$	(alpha05) 0.278	(alpha05-noR) 0.345
$\alpha = 0.2$	(alpha02) 0.301	0.368
$\alpha = 0.1$	0.312	0.388

Table 35: A comparison of the performance of ARQMath-1 submission runs (indicated inside parenthesises) and experimental runs in nDCG′. The runs have the same primary setting but with a varying α and with or without a re-ranking using $\text{reRank}_{\text{LRM}}$. It can be observed that the result without a re-ranking is consistently better regardless of the α value.

E.2 ARQMath-2 Submission Runs

Following the same methodology, the ARQMath-2 submissions are continuations of the ARQMath-1 submissions. Different from the ARQMath-1 submissions—where alternative runs serve as a validation for the varying configurations—the ARQMath-2 submissions already use most of the best possible configurations validated by parameter tuning on the ARQMath-1 benchmark (Section 4.4). As such, the submissions serve to verify the authenticity of such parameter tuning, since the result from experimental runs can be doubtful when the runs are actually evaluated with unjudged answers removed, while submitted runs can have all of their top answers being fully judged (Section 2.4.2).

A primary run and one alternative run were submitted. The details of the runs are as follows:

primary: The primary run with most of the best possible configurations tuned on the ARQMath-1 benchmark. Compared to the previous primary run submitted in ARQMath-1 (*alpha05*), this primary run was created with $\text{Dataset}_{\text{clean}}$ and the Tangent-L variant in use was coreTL-$\alpha_{0.27}\gamma_{0.10}$ with $\text{FN}_{\text{C+S}}$ enabled. The answer ranking used Tangent-L's internal ranking without any re-ranking.

proximityReRank: An alternative run that has the same setting as the primary run, followed by a re-ranking with $\text{reRank}_{\text{prox}}$. The proximity measure used was *Normalized-Span* as described in Section 4.3.2 and analysed in Section 4.4.7.

From Table 27, it can be observed that the primary run (*primary*) is the best participant run, with indistinguishable difference in nDCG′ (0.433 for ARQMath-1, 0.434 for

ARQMath-2) on both benchmarks. A similar improvement reflected on the ARQMath-2 benchmark indicates that parameter tuning using the ARQMath-1 benchmark has been effective. Remarkably, during ARQMath-2—as well as ARQMath-1—when creating search queries for the submitted runs, duplicate terms were extracted, but their weights were not boosted accordingly ($\text{uniqueQuery}_{\text{MSEWiki}_{\text{FF}}}$) because of an oversight in the implementation. After the correction ($\text{Query}_{\text{MSEWiki}_{\text{FF}}}$), the experimental run with almost the same setting ($primary_{exp}$[1]) as the *primary* run achieves a further 3-point gain in nDCG′ (0.458 vs 0.433 for ARQMath-1, 0.463 vs 0.434 for ARQMath-2). Such success validates the power of the presented methodology and the adopted configurations.

On the other hand, the alternative run *proximityReRank* does not perform well. As an experimental run applied on the ARQMath-1 benchmark without full judgement, it already shows a 6-point loss of nDCG′ when compared to the *primary* run (0.373 vs 0.433). As a submitted run with its top answers fully judged for the ARQMath-2 benchmark, its performance loss is further enlarged to nearly 10 points (0.335 vs 0.434), indicating an unsatisfactory re-ranking.

It remains a challenge to further improve the result. Besides the alternative run *proximityReRank*, another attempt is the experimental run $holisticTL\text{-}\gamma_{0.10}\alpha_{0.47}\kappa_{400}$[2], where Tangent-L adopts a different retrieval model to search formulas holistically as described in Section 3.4. The result of this experimental run, however, shows that it is not competitive yet when compared to $primary_{exp}$, with a 5-point loss in nDCG′ on both benchmarks (0.409 vs 0.458 for ARQMath-1, 0.413 vs 0.463 for ARQMath-2).

[1] reported as the *duplicateTerms* run in [38].
[2] reported as the *holisticSearch* run in [38].

Appendix F

Machine Specifications and Efficiency

F.1 Machines used for the ARQMath-1 system

Indexing.

Indexing is done on a Ubuntu 16.04.6 LTS server, with two Intel Xeon E5-2699 V4 Processors (22 cores 44 threads, 2.20 GHz for each), 1024GB RAM and 8TB disk space (on an USB3 external hard disk). The size of the document corpus is 24.3GB.

Tangent-L requires 5.0GB of storage on the hard drive and approximately 6 hours to index all documents with parallel processing.

Searching and Re-ranking.

Training and testing the model for re-ranking is done on a Linux Mint 19.1 machine, with an Intel Core i5-8250U Processor (4 cores 8 threads, up to 3.40 GHz), 24GB RAM and 512GB disk space.[1]

Model training using the Python scikit-learn library[2] takes less than 30 seconds, and re-ranking for all 98 topics requires around 3 seconds per run.

Searching is executed on this same Mint machine, and retrieval time statistics for Tangent-L are reported in Table 36.

[1] A NVIDIA GeForce MX150 graphics card with 2GB on-card RAM is available on the machine, but it was not used for the experiments.

[2] https://scikit-learn.org

	Avg. (sec)	*Top-2 Min (sec)*	*Top-2 Max (sec)*
alpha05 †	13.3	0.669 (A.67) / 0.775 (A.94)	63.4 (A.76) / 48.7 (A.28)
alpha02	13.3	0.661 (A.67) / 0.850 (A.94)	59.1 (A.76) / 49.5 (A.28)
alpha10	13.1	0.616 (A.67) / 0.784 (A.83)	54.0 (A.76) / 48.7 (A.28)
alpha05-translated	5.3	0.247 (A.99) / 0.291 (A.94)	32.8 (A.11) / 25.0 (A.67)
alpha05-noReRank	13.3	0.669 (A.67) / 0.775 (A.94)	63.4 (A.76) / 48.7 (A.28)

† The run *alpha05* does not, in fact, take any *additional* retrieval time, since it merely re-ranks the retrieved items from the *alpha05-noReRank* run.

Table 36: Retrieval times per topic, in seconds, using single-threaded processing.

F.2 Machines used for the ARQMath-2 system

The machines used for the experiments have the following specifications:

Machine A	A Ubuntu 20.04.1 LTS Server with an AMD EPYC™ 7502P Processor (32 Cores 64 Threads, 2.50GHz), 512GB RAM and 3.5TB disk space.
Machine B	A Linux Mint 19.1 Server with an Intel Core i5-8250U Processor (4 Cores 8 Threads, up to 3.40GHz), 24GB RAM and 512GB disk space.

All indexing was performed on Machine A, yielding the following performance characteristics:

Corpus	See Section	Data Size (GB)	Index Size (GB)	Indexing Speed (sec)
Document Corpus	4.2.3	23	4.1	4394
Formula Corpus	3.4.1	34	4.7	4834
Document Corpus (for holistic search)	4.4.6	23	0.6	167

Note that data and index sizes show the values reported by the `du` command on Linux, which measures disk space usage based on blocks; thus the many small documents in the formula corpus require much more disk space than might be expected. (In fact, the total size of the data in the formula corpus is only 9.2 GB.)

Runs for ARQMath-2 were executed on Machine B with the following average, minimum, and maximum query times per topic as follows:

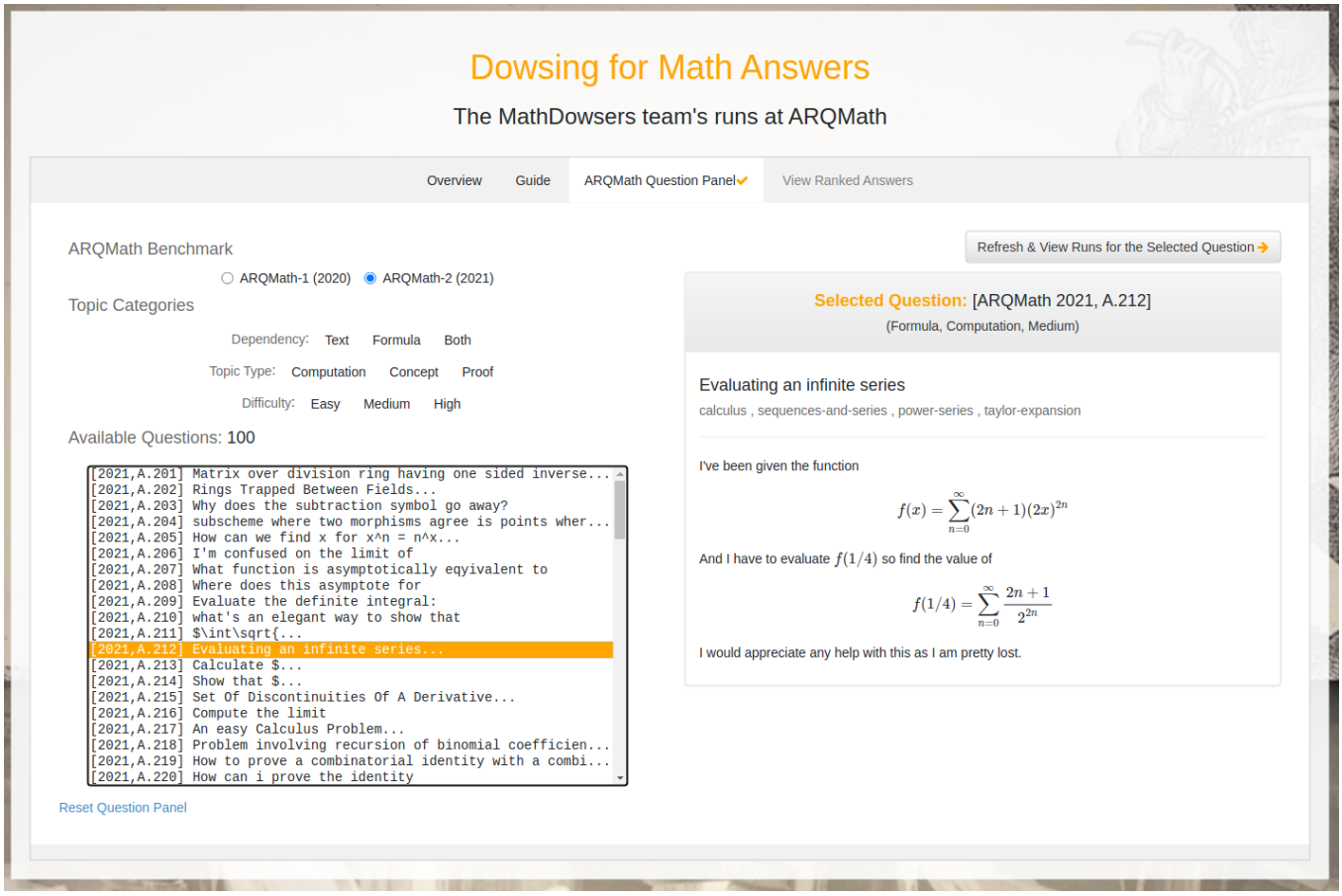
Run \ Query Time	Avg. (sec)	Min. (sec)	Max. (sec)
Task 1			
primary	1.90	0.34 (A.264)	6.39 (A.221)
holisticSearch	7.77	2.37 (A.264)	24.5 (A.221)
duplicate	1.92	0.30 (A.264)	6.04 (A.272)
Task 2			
(pre-computing Answer-Ranking)	1.94	0.48 (A.264)	6.03 (A.221)
formulaBase	1.16	0.22 (B.244)	3.79 (B.270)
docBase	56.5	16.5 (B.209)	122 (B.221)

The proximityReRank run uses Machine A to re-rank the output from the primary run, thus requiring first the time shown for the primary run on Machine B and then an additional 8 hours to re-rank all topics on Machine A.

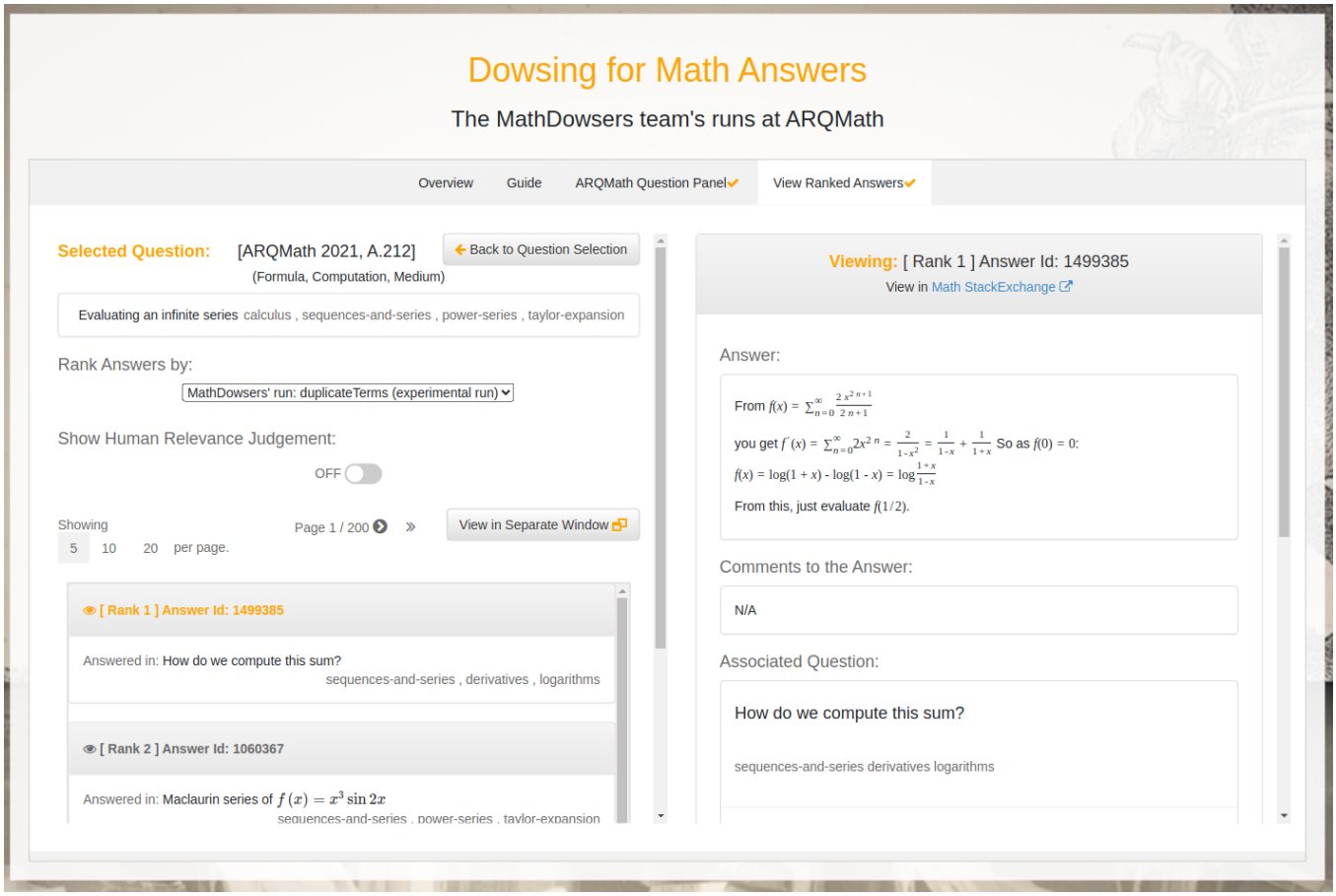
Appendix G

User Interface of the MathDowsers' Browser

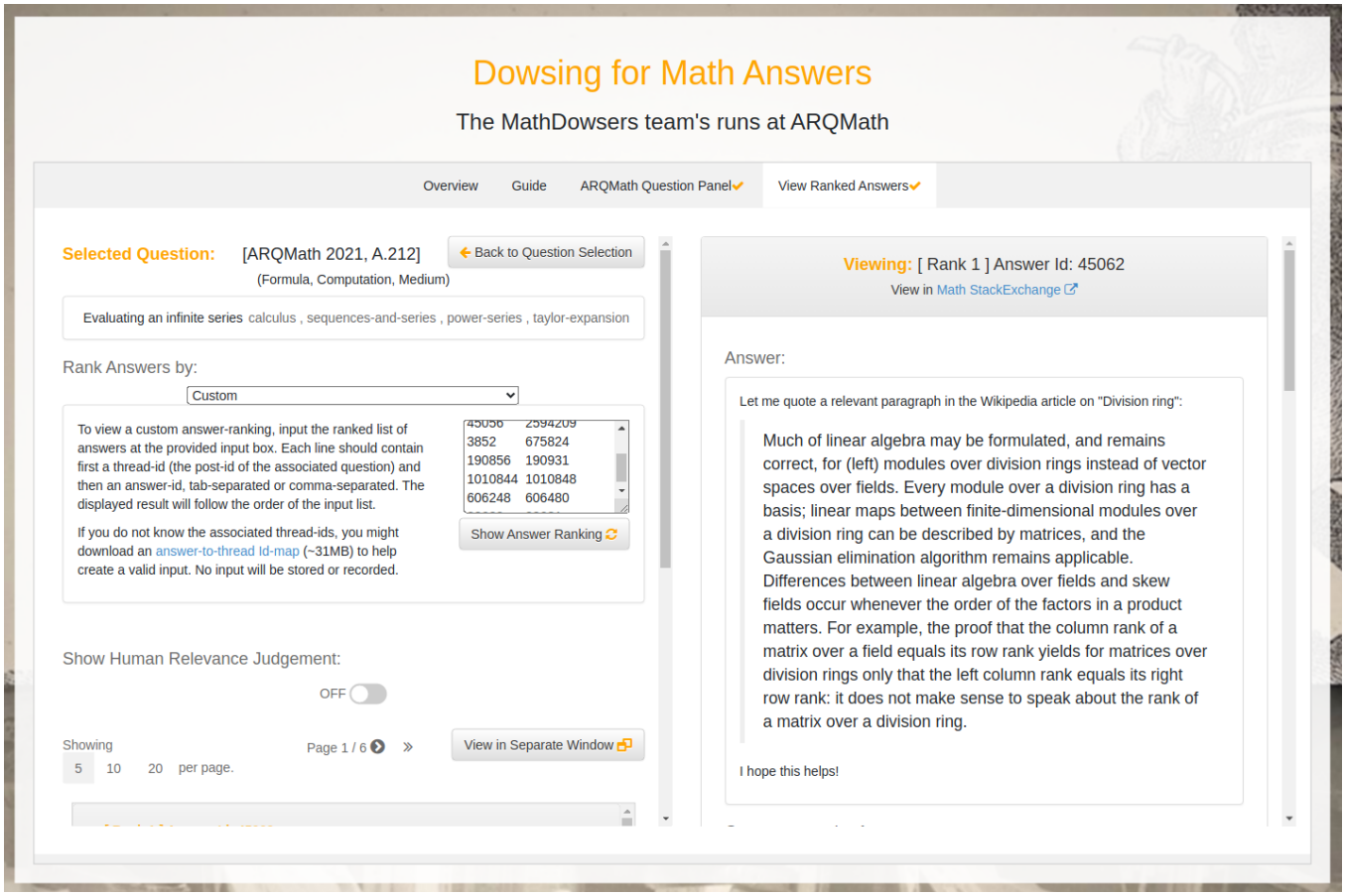
G.1 The ARQMath Question Panel



G.2 The Answers Panel.



G.3 Interface for inputting a custom answer ranking



G.4 Displaying human relevance judgments