Ensembling Ten Math Information Retrieval Systems

MIRMU and MSM at ARQMath 2021

Vít Novotný[1], Michal Štefánik[1], Dávid Lupták[1], Martin Geletka[1], Petr Zelina[1] and Petr Sojka[1]

[1]Faculty of Informatics, Masaryk University, Botanická 68a, 602 00 Brno, Czech Republic

Abstract

We report on the systems that the Math Information Retrieval group at Masaryk University (MIRMU) and the team of Faculty of Informatics students (MSM) prepared for task 1 (find answers) of the ARQMATH lab at the CLEF conference. We have prototyped ten math-aware information retrieval (MIR) systems for the main question-answering task. We ensembled the results of the ten "weak" individual systems into committees and let them vote to provide answers to questions. We evaluated the proposed invidi-vidual systems and ensembles, considering their diversity, hyperparameters, and representations used, and classified their approaches. We have shown the diversity of all systems and evaluated four voting algorithms to collect and rank the answers. Ensembling techniques consistently outperformed the base systems and showed the power of voting of diverse systems. Our prototypes help to understand the challenging problems of question answering in the STEM domain and our novel reproducible evaluation framework sets a new direction in MIR research. Finally, we formulate ten commandments for future work in the area.

Keywords

Information retrieval, question answering, math representations, math-aware information retrieval, word embeddings, ensembling, voting, ranking, data fusion

> "I do not demand that you make me happy; my happiness does not lie in you." Anthony de Mello

1. Introduction

This report describes the submissions Math Information Retrieval (MIR) group at Masaryk University (MIRMU) [1] with the team of Master students (MSM) prepared for question answering task (task 1) of ARQMath 2021 lab [2, 3]. Encouraged with our previous results [4, 5, 6] and participation in NTCIR-10, NTCIR-11, NTCIR-12 Math information retrieval challenges, and recent ARQMath 2020 lab [7] results [8, 9], we continued our efforts to tackle the challenging math question answering task with new approaches as ensembling techniques. This year we concentrated on the program of ten specific research questions and challenges:

Q1: diversity of systems How different systems answer questions? What is the variance, and how to cope with it?

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

witiko@mail.muni.cz (V. Novotný); stefanik.m@mail.muni.cz (M. Štefánik); 422640@mail.muni.cz (D. Lupták); 456576@mail.muni.cz (M. Geletka); 469366@mail.muni.cz (P. Zelina); sojka@fi.muni.cz (P. Sojka)
0000-0002-3303-4130 (V. Novotný); 0000-0003-1766-5538 (M. Štefánik); 0000-0001-9600-7597 (D. Lupták); 0000-0002-5768-4007 (P. Sojka)

CEUR Workshop Proceedings (CEUR-WS.org)

Q2: diversity of topics How varies different expressions of information needs as questions answered by information systems? How systems handle out-of-distribution questions, and how their performance varies w.r.t. topics?

Q3: ensembling Could ensembling techniques give consistently better results than individual systems?

Q4: ensemble voting strategies To which extent could ensembles and query classification benefit from the diversity of individual systems?

Q5: representation To which extent the query and document representation and indexing of the meaning of formulae influence the system's performance? How to collate the representation of text and formulae together? How to grab and disambiguate the meaning of symbols in the formulae?

Q6: attention To which extent the new attention-based approaches could be deployed for math question answering? How does it compare to the standard fine-tuned information retrieval approaches, developed for sole text retrieval?

Q7: performance To what extent are appropriate and valuable the standard information retrieval techniques as query expansion, keyword similarity metrics, or probabilistic approaches as BM25?

Q8: canonicalization Does the canonical representation of formulae matter in the math question answering task?

Q9: inference How to integrate the deduction into a math question answering system?

Q10: explainability How to provide arguments of answer ranking based on the ensembling algorithm and scoring of individual systems?

To answer these questions, we collected ten individual math-aware systems for task 1. It was possible due to PV211 Introduction to Information Retrieval course student projects, and due to works of PV174 Seminar of Laboratory of Electronic and Multimedia Applications taught at the Faculty of Informatics, Masaryk University by the last author. Ten available systems, together with ground truth data from ARQMath 2020 evaluation lab gave us means to measure and evaluate their (hyper)parameters, different ensembling techniques, different representations, different data preprocessing techniques, different evaluation measures, different query expansion strategies, different reranking algorithms, and tackle prescribed questions with rigorous research methodology.

Our main objective was to gain insight into the research problems above and answer some of them. To this end, we submitted five result lists for each MIRMU (by PV174 seminar) and MSM (by PV211 course) teams.

We think that the key in solving the challenges lies in the accurate, evidence-based evaluation of available systems, and their parameters, preprocessing and representation of math-aware data and texts. We have compared our ten available systems using both unsupervised and supervised approaches for systems' hyperparameter tuning, representation learning. Finally,

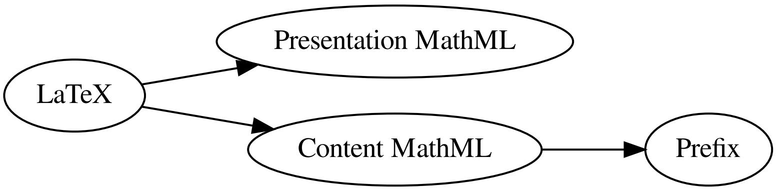


Figure 1: Dependencies between math representations ingested by our systems.

we have ensembled our submissions into four committees of MIRMU individual systems that have achieved a better result on 2020 data than any of the single ensembled algorithms alone.

In this paper, we report our experiments and achievements in detail. Section 2 describes the resources, data preprocessing, representations, and methods used. Section 3 on page 7 reports on ten individual systems and their settings. The gentle reader finds the description of our ensembling algorithms in Section 4 on page 14. We thoroughly discuss our results, insights we got, and future directions in sections 5 on page 19 and 6 on page 20.

> "Don't ask the world to change... you change first."
> Anthony de Mello

2. Datasets and Methods

This section will describe the math representations ingested by our information retrieval systems, the corpora used for training the models that power our systems, and the relevance judgments we used for parameter optimization, model selection, and performance estimation.

2.1. Math Representations

There are variety of formats for math formulae representation and ranking [10] at our fingertips: LaTeX, Presentation MathML (PMML), Content MathML (CMML), Symbol Layout Tree (SLT), Operator Tree (OPT), [11] M-Terms, the prefix notation, and the infix notation. For PMML and CMML canonicalized [12] versions might also be used. Figure 1 shows how the individual math representations are derived from LaTeX, which is prevalent author format.

2.1.1. LaTeX

As the most direct math representation, we used LaTeX, the standard and most frequent authoring format for math. Although LaTeX is easy to type and preferred by authors, it often encodes the presentation aspects of a math formula rather than its content. LaTeX is also a Turing-complete language and, therefore, impossible to parse in the general case statically. As a result, each formula is represented as a single token in the LaTeX representation.

LaTeX is helpful as a baseline math representation and as a basis for deriving more fine-grained math representations described in the following sections. Although having each formula represented as a single token may not seem helpful, two of our ten systems (SCM and COMPUBERT) model subwords, which allows them to extract symbols out of the formulae.

To give an example, the formula $x!! - y^2 = 0$ would be represented as a single token `$x!! - y^2 = 0$` in LaTeX.

2.1.2. Prefix Notation

To linearize the Operator Tree (OPT), [11] we converted math formulae from OPT into the prefix notation. The prefix notation corresponds to the list of visited nodes in the OPT in the depth-first-search order, i.e., the topological sorting of the OPT. Like LaTeX, the prefix notation is easy to type. Unlike LaTeX, the prefix notation is tokenized into math symbols and independent of a formula's presentation aspects.

To give an example, the formula $x!! - y^2 = 0$ would be represented as the following space-separated list of tokens in the prefix notation: `U!eq O!minus O!double-factorial V!x O!SUP V!y N!2 N!0`.

2.2. Document Collections

For training our models, we used the arXMLiv and Math Stack Exchange corpus. Our data preprocessing code is available online.[1]

2.2.1. ArXMLiv

The arXMLiv 08.2019 corpus [13] contains 1,374,539 articles from the arXiv.org open-access archive converted from LaTeX to HTML5 and MathML. We split the corpus into four subsets: no_problem (150,701 articles), warning_1 (500,000 articles), warning_2 (328,127 articles), and error (395,711 articles), according to the severity of errors encountered when converting LaTeX to HTML5. We only used the no_problem, warning_1, and warning_2 subsets (978,828 articles) of the corpus to train our models.

2.2.2. Math Stack Exchange

The Math Stack Exchange collection V1.2 (M-SE) provided by the organizers of the ARQMATH 2021 competition contains 2,466,080 posts from the Math Stack Exchange question answering website in HTML5 and LaTeX. Besides the answers (1,445,495), which are the retrieval unit in task 1 of ARQMATH, the posts also contain questions (1,020,585) related to the answers and can be used for learning what a good answer for a question is.

Posts in the M-SE collection contain 28,320,920 math formulae. In ARQMATH 2020, only 26,075,012 math formulae in PMML (92.07%) and 25,366,913 math formulae in CMML (89.57%) have been successfully converted and provided by the organizers as Formulas v1.0. To improve the conversion success rate, we performed our conversion from LaTeX to 26,705,527 math formulae in CMML (94.30%) and 27,232,230 math formulae in PMML (96.16%). In ARQMATH 2021, we collaborated with the organizers to provide 28,282,477 math formulae in both PMML and CMML (99.86%) as Formulas v2.0.

The M-SE collection is structured and contains not only the body texts but also the titles, tags, comments, up- and down-votes, view counts, and authorship information, among other things.

[1] https://github.com/MIR-MU/ARQMath-data-preprocessing

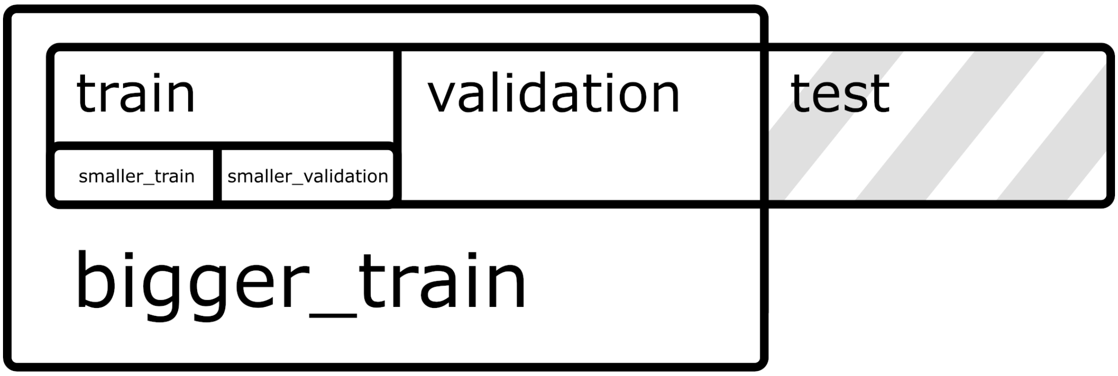


Figure 2: We split the 39,124 judgements over 77 topics from task 1 of ARQMATH 2020 into subsets for training, hyperparameter optimization, model selection, and performance estimation in our systems.

Although the collection provides a wealth of information, it is difficult to navigate and can cause choice overload for newcomers. To make the development of new systems easier, we simplified the corpus into two OOP classes `ArqmathQuestionBase` and `ArqmathAnswerBase`, in our `pv211-utils` Python library.[2] We also created a Jupyter Notebook at the Google Colaboratory service[3] as a template for quick development of new math information retrieval systems by our students using our `pv211-utils` library.

2.3. Queries and Relevance Judgements

Official ARQMATH 2020 human-annotated task 1 and 2 relevance judgments produced by eight annotators with the fair agreement ($\kappa = 0.34$) [14] are available. We used the relevance judgments for training, hyperparameter optimization, model selection, and performance estimation in our systems. For 77 topics from task 1, the documents were evaluated with a range from 0 (not relevant) to 3 (highly relevant) as the gain. The relevance judgements are highly imbalanced in favor of non-relevant answers: Out of all 39,124 judgements, there exist as many as 35,051 judgements (89.59%) with gain 0, 2,269 judgements (5.8%) with gain 1, 1,071 judgements (2.74%) with gain 2, and only 733 (1.85%) judgements with gain 3.

Out of the 39,124 judgements over 77 topics, we produced three primary subsets (see Figure 2):

Train (55) The train subset contains a stratified sample of 55 topics (71.43%) and their associated 27,830 judgements (71.13%). We produced the stratified sample of 55 topics by taking three simple random samples of 19 computation topics, 7 concept topics, and 29 proof topics. To divide the topics into computation, concept, and proof categories, we used the detailed annotations provided by organizers [15] for topics of both years. We used the train subset for training supervised models in our systems.

Validation (11) The validation subset contains a stratified sample of 11 topics (14.29%) and their associated 5,652 judgements (14.45%). We produced the stratified sample of 11 topics as

[2] https://gitlab.fi.muni.cz/xstefan3/pv211-utils/-/blob/master/pv211_utils/arqmath/entities.py
[3] https://drive.google.com/file/d/1HA2Bm77nBiz-UTckaYousZLe8RhgaDP4

in the train subset. We used the validation subset for either hyperparameter optimization or model selection in our systems.

Test (11) The test subset contains a stratified sample of 11 topics (14.29%) and their associated 5,642 judgements (14.42%). We produced the stratified sample of 11 topics as in the train and validation subsets. We used the test subset for performance estimation of our systems before their submission to ARQMath 2021.

Out of the three primary splits, we produced three secondary subsets (see Figure 2 on the previous page):

Bigger train (66) By taking a union of the train and validation subsets, we produced the bigger train subset of 66 topics (85.71%) and their associated 33,482 judgements (85.58%). We used the bigger train subset for training supervised models in systems, where neither hyperparameter optimization nor model selection was required.

Smaller train (44) By taking a simple random sample of the train subset, we produced the smaller train subset of 44 topics (57.14%) and their associated 22,241 judgements (56.85%). We used the smaller train subset for training supervised models in systems, where both hyperparameter optimization and model selection were required.

Smaller validation (11) By taking a simple random sample of the train subset, we produced the smaller train subset of 11 topics (14.29%) and their associated 5,589 judgements (14.29%). We used the smaller validation subset for hyperparameter optimization in systems, where both hyperparameter optimization and model selection were required.

We release our six subsets of topics and judgements in our ARQMATH-eval Python library.[4]

2.4. Evaluation Measures

In hyperparameter optimization, model selection, and performance estimation, we used the *normalized discounted cumulative gain prime (nDCG′)* to estimate information retrieval accuracy.

To determine the diversity of our systems, we used the *Spearman's rank-correlation coefficient (ρ)* between the result lists of our systems averaged across the topics from task 1 of ARQMath 2020 and 2021 to measure the similarity of our systems for clustering. To select the optimal number of clusters, we used the *silhouette score* as a measure of clustering quality.

To measure the speed of our systems, we measured the *wall clock time* on a dedicated machine.

2.4.1. Normalized Discounted Cumulative Gain Prime

The normalized discounted cumulative gain prime (nDCG′ [16]) is an evaluation measure specifically designed for information retrieval with incomplete judgements. nDCG′ is defined as follows:

$$\textsc{ndcg}' = \operatorname*{avg}_{t \in T} \frac{\textsc{dcg}'_t}{\textsc{idcg}_t}, \ \textsc{idcg} = \sum_{i=1}^{|\textsc{rel}_t|} \frac{\text{gain}_t(\textsc{rel}_{t,i})}{\log_2(i+1)}, \text{ and } \textsc{dcg}' = \sum_{i=1}^{|\textsc{res}'_t|} \frac{\text{gain}_t(\textsc{res}'_{t,i})}{\log_2(i+1)}, \tag{1}$$

[4] https://github.com/MIR-MU/ARQMath-eval, files scripts/qrel_task1-⟨*subset name*⟩-pv211-utils.tsv

where T are the topics for a task, $\textsc{rel}_t$ is a list of relevant documents for topic t in the descending order of their gain up to position 1,000, $\textsc{res}_t$ is a list of results produced for topic t our system up to position 1,000, $\textsc{res}'_t = \textsc{rel}_t \cap \textsc{res}_t$, and $\text{gain}_t(R)$ is the gain of result R for topic t.

2.4.2. Spearman's Rank-Correlation Coefficient

Spearman's rank-correlation coefficient (ρ) [17] is a general non-parametric measure of rank correlation. Spearman's ρ between random variables X and Y corresponds to Pearson's correlation coefficient (r) between the rank variables rg_X and rg_Y:

$$\rho = \frac{\text{cov}(\text{rg}_X, \text{rg}_Y)}{\sigma_{\text{rg}_X} \cdot \sigma_{\text{rg}_Y}} \text{ and } \text{cov}(\text{rg}_X, \text{rg}_Y) = \text{E}\big[(\text{rg}_X - \text{E}[\text{rg}_X])\cdot(\text{rg}_Y - \text{E}[\text{rg}_Y])\big]. \tag{2}$$

In our experiments, rg_X and rg_Y correspond to the ranks of the same answer in the result lists of two systems for a single topic from task 1 of ARQMATH 2021.

2.4.3. Silhouette Score

The silhouette score (s) [18] is an intrinsic measure of clustering quality that compares the intra-cluster cohesion (a) and the inter-cluster separation (b). The score s is defined as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}},\ a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j), \text{ and } b(i) = \min_{j, i \neq j} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j), \tag{3}$$

where i is a data point, C_i is the cluster of data point i, and $d(i, j)$ is the distance between data points i and j. In our experiments, data points were the ensembled systems, the distance measure d was $1 - \rho$, and we selected the number of clusters that maximized the expected value $\text{E}[s]$.

2.4.4. Wall Clock Time

The wall clock time is the time experienced by the user. We measure the wall clock time of the preprocessing, training, and information retrieval on a dedicated machine with two NVIDIA Tesla T4 GPUS (16 GB VRAM), 377 GB RAM, and four Intel(R) Xeon™ Gold 6230 CPUS (80 cores at 2.10 GHz).

> "The main thing is not study, but doing."
> Chapters of the Fathers (Pirkei Avot, 1:17)

3. Individual Systems

In this section, we will describe the individual systems submitted by the MIRMU and MSM teams. We describe two best-performing systems of the MSM team (MG and PZ) and the MIRMU team (SCM and COMPUBERT) in detail. We then briefly summarize the architectures and the results of our remaining systems on task 1 of ARQMATH 2021.

3.1. BM25^{+} (MSM – MG)

The BM25 algorithm is often used as the first choice for many information retrieval tasks because of its simplicity and its better performance over TF-IDF systems. In this section, we will describe our system based on the BM25^{+} model. Our experimental code is available online.[5]

BM25^{+} is an improvement over BM25 introduced by Lv and Zhai [19]. Together with other alternatives, such as BM25-L, BM25-adapt, and BM25-T, this improvement surpasses the basic BM25 algorithm on TREC collections. [20] BM25^{+} estimates the relevance of a document d for a query q as follows:

$$\mathrm{BM25}^{+}(d,q) = \sum_{t \in q} \log\left(\frac{N+1}{\mathrm{df}_t}\right) \cdot \left(\frac{(k_1+1)\cdot \mathrm{tf}_{t,d}}{k_1 \cdot \left((1-b)+b\left(\frac{L_\mathrm{d}}{L_\mathrm{avg}}\right)\right) + \mathrm{t}_\mathrm{fd}} + \delta\right), \tag{4}$$

where k_1, b, and δ are hyperparameters, N is the number of documents in the collection, df_t is the number of documents containing the term t, $\mathrm{tf}_{t,d}$ is frequency of term t in document d, L_d the length of document d in words, and L_avg is the expected length of a document in words.

3.1.1. Configuration

In our preprocessing, we tokenized text with math formulae in LaTeX by splitting on sequences of whitespaces. We then stemmed the individual tokens using the English Snowball stemmer available in the NLTK Python library. We represented each answer as the concatenation of its body with the title, body, and tags of its parent question. We used the implementation of BM25^{+} in the rank_bm25 Python library[6].

We optimized the hyperparameters k_1, b, and δ using grid search. The default parameters $k_1 = 1.5, b = 0.75$, and $\delta = 1.0$ achieved the best NDCG′ on the bigger train subset.

3.1.2. Results

On the test subset, BM25^{+} achieved 0.464 NDCG′, which is the best result of all our individual systems. On the judgements for task 1 of ARQMATH 2021, BM25^{+} achieved 0.278 NDCG′, which is again the best result of all our individual systems.

Our preprocessing of the M-SE collection took 50:15 minutes and the indexing took another 65 seconds. The average query time was 69.3 seconds with a minimum of 22.4 seconds for topic A.289 and a maximum of 318.9 seconds for topic A.216.

3.2. Pyserini (MSM – PZ)

Pyserini [21] is an Apache-licensed Python library for reproducible information retrieval. It uses Anserini [22] for sparse representation-based retrieval and Faiss [23] for dense representation-based retrieval. Pyserini is quite fast, easy to use, and comes with several prebuilt indexes.

[5] https://colab.research.google.com/drive/1lqSx2a4hVHFW9xL2KGiVJMEniVzhQXaO

[6] https://github.com/dorianbrown/rank_bm25

Creating custom indexes is supported as well. In this section, we will describe our system based on Pyserini. Our experimental code is available online.[7]

3.2.1. Configuration

In our preprocessing, we used the LaTeX representation of math formulae. We represented each answer as the concatenation of its body with *three repetitions* of the title and the tags of its parent question. The answers were then preprocessed and indexed by the Pyserini indexer, which is based on Anserini and Lucene, with the default settings: the removal of possessives, lowercasing, the removal of English stopwords, and stemming with the Porter stemmer.

The document relevance was estimated by the `SimpleSearcher` class, which corresponds to the BM25 model with the default hyperparameters $k_1 = 0.9$ and $b = 0.4$. Even though Pyserini supports many other options and extensions, such as the RM3 query expansion, other ranking models, and dense document reranking, our system used the default options.

3.2.2. Results

On the test subset, Pyserini achieved 0.449 nDCG′, which is the second best result of all our individual systems. On the judgements for task 1 of ARQMath 2021, Pyserini achieved 0.275 nDCG′, which is again the second best result of all our individual systems.

Our preprocessing and indexing of the M-SE collection took 3:44 minutes. The average query time was 1.1 seconds with a minimum of 0.5 seconds for topic A.264 and a maximum of 2.6 seconds for topic A.221.

3.3. Soft Cosine Measure (MIRMU – SCM)

Math information retrieval systems often rely on Salton's TF-IDF model [24], which is interpretable, but which also reduces meaningful statements in human and mathematical languages to an unintelligible salad of key words and symbols. At ARQMath 2020, we introduced the soft vector space of Sidorov et al. [25] and its soft cosine document similarity measure (SCM, see Figure 4 on the following page) and we achieved the best nDCG′ on task 1 of ARQMath 2020 of all our individual systems. [9, Section 4] In this section, we will describe our system based on SCM. Our experimental code is available online.[8]

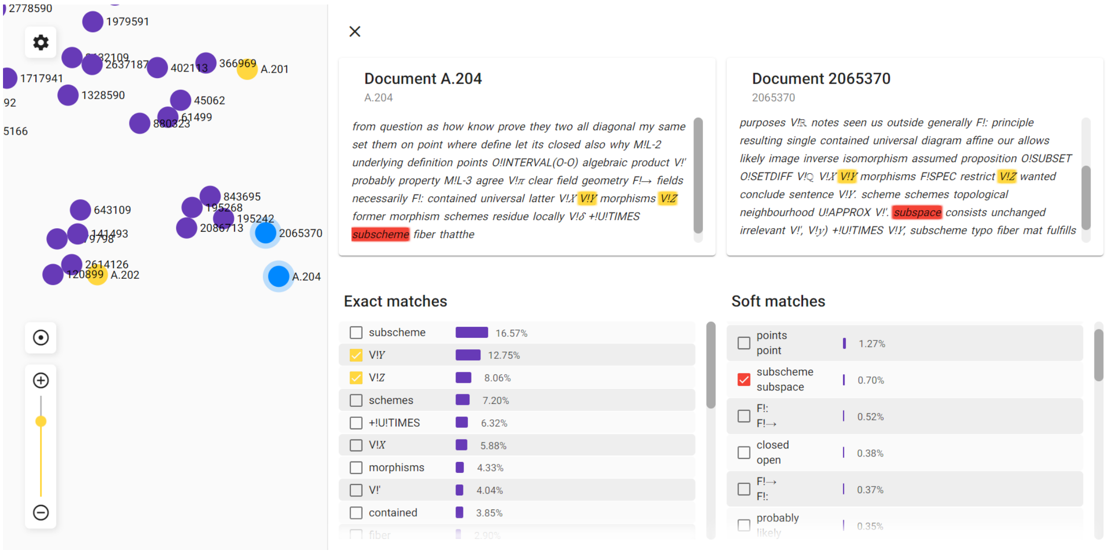
For ARQMath 2021, we also produced an online demo of the SCM,[9] which allows the user to interactively explore a small set of topics[10] and their nearest answers. The demo also allows the user to compare two documents to see why they are considered similar by the SCM, see Figure 3 on the next page.

[7] https://colab.research.google.com/drive/1yzC6p-tkeYIeJlqxA4U75FNsM2_eEWa4

[8] https://colab.research.google.com/drive/1LACGqdkUUeprHGTrEoocWoSOavpfP5Ki

[9] https://mir.fi.muni.cz/document-maps-arqmath-2021

[10] https://mir.fi.muni.cz/document-maps-arqmath-2021/assets/example.json



Document A.204

A.204

from question as how know prove they two all diagonal my same set them on point where define let its closed also why M!L-2 underlying definition points O!INTERVAL(O-O) algebraic product V!′ probably property M!L-3 agree V!π clear field geometry F!→ fields necessarily F!: contained universal latter V!X V!Y morphisms V!Z former morphism schemes residue locally V!δ +!U!TIMES subscheme fiber thatthe

Document 2065370

2065370

purposes V!$\mathbb{R}$ notes seen us outside generally F!: principle resulting single contained universal diagram affine our allows likely image inverse isomorphism assumed proposition O!SUBSET O!SETDIFF V!$\mathbb{Q}$ V!X V!Y morphisms F!SPEC restrict V!Z wanted conclude sentence V!Y. scheme schemes topological neighbourhood U!APPROX V!′. subspace consists unchanged irrelevant V!′, V!y) +!U!TIMES V!Y, subscheme typo fiber mat fulfills

Exact matches

☐ subscheme	16.57%
☑ V!Y	12.75%
☑ V!Z	8.06%
☐ schemes	7.20%
☐ +!U!TIMES	6.32%
☐ V!X	5.88%
☐ morphisms	4.33%
☐ V!′	4.04%
☐ contained	3.85%
☐ fiber	2.90%

Soft matches

☐ points point	1.27%
☑ subscheme subspace	0.70%
☐ F!: F!→	0.52%
☐ closed open	0.38%
☐ F!→ F!:	0.37%
☐ probably likely	0.35%

Figure 3: An online demo of the SCM system, which allows the user to interactively explore a small set of topics and their nearest answers. The demo also allows the user to compare two documents to see why they are considered similar by SCM.

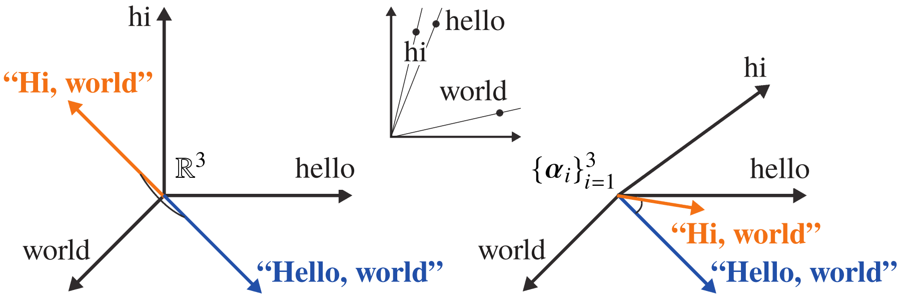


Figure 4: The representation of two documents, "Hi, world" and "Hello, world" in the TF-IDF vector space model (VSM, left) and in the TF-IDF soft vector space model (soft VSM, right). In the VSM, different terms correspond to orthogonal axes, making the document representations distant despite their semantic equivalence. In the soft VSM, different terms correspond to non-orthogonal axes, where the angle between the axes is proportional to the similarity of terms in a word embedding space (middle).

3.3.1. Configuration

In our preprocessing, we tokenized text with math formulae in *the prefix notation* by first splitting on whitespaces to separate text and math tokens. Then, we upper-cased the math tokens and we lower-cased the text tokens, so that they contained different subwords for the training of fastText embeddings (described below). We then performed a second tokenization of the text tokens to remove punctuation and numbers using the simple_preprocess function from the Gensim Python library [26]. We represented each answer as the concatenation of its body with *three titles*, the body, and the tags of its parent question.

As our source of similarity between words and symbols, we used our *Medium* fastText embeddings of text and math[11], which were trained on both the M-SE collection and ArXMLiv and which achieved the best NDCG′ on task 1 of ARQMATH 2020. [9, Section 4.4] From the fastText embeddings, we extracted a term similarity matrix[12] using the optimal parameters $\text{Sym} = \checkmark, \text{Dom} = \checkmark$, and $C = 100$ from ARQMATH 2020. [9, Section 4.3]

We optimized the SMART weighting scheme of our TF-IDF model using grid search. The Lnu.ltb weighting scheme with the pivoted document length normalization [27] at slope 0.2 achieved the best NDCG′ on the bigger train subset.

3.3.2. Results

On the test subset, SCM achieved 0.424 NDCG′, which is the third best result of all our individual systems. On the judgements for task 1 of ARQMATH 2021, SCM achieved 0.250 NDCG′, which is the fourth best result of all our individual systems.

Our preprocessing of the M-SE collection took 13:28 minutes. The training of the fastText embeddings took 01:45 hours and the construction of the term similarity matrix took another 32:39 minutes. The hyperparameter optimization took 04:42:34 hours. The average query time was 223.39 seconds with a minimum of 210.68 seconds for topic A.254 and a maximum of 270.18 seconds for topic A.273. Using a single matrix product to retrieve results for all 100 topics from task 1 of ARQMATH 2021 took only 353.0 seconds, which is 63.28× faster than ad-hoc.

3.4. Computational BERT (MIRMU – COMPUBERT)

Our COMPUBERT system aims to utilize the expressive power of pre-trained Transformer models [28] and the results of applying the Transformer architecture to complex math-related tasks, such as computing derivatives and first-order differential equations. [29] In this section, we will describe COMPUBERT and its results on task 1 of the ARQMATH 2020 competition. Our experimental code is available online.[13]

3.4.1. Matching Questions with Answers

In addition to math representation, we have to contend with additional challenges characteristic to information retrieval but alien to Transformers: While the original Transformer architecture [28] builds upon the Wordpiece text segmentation [30] that optimizes the representation of subwords (not unlike fastText in the SCM), we also need to uniformly represent long spans of text.

We address this challenge with an approach introduced by Reimers and Gurevych [31] and shown in Figure 5 on the following page. The underlying idea of their *Sentence Transformers* is to adjust the pre-trained language model so that the pooled representation of longer span of text respect an objective of given task.

[11] https://drive.google.com/file/d/1L6yz4cTyrPZgb-gkpLfAw-XTUVOK4tpZ

[12] https://drive.google.com/file/d/1HIPIwYvEK-HsQgYpZ0lt7KE7L81AkUIQ

[13] https://drive.google.com/drive/folders/1bxYwWzDX3z81S4TwUaTvqZBHtiMOngez

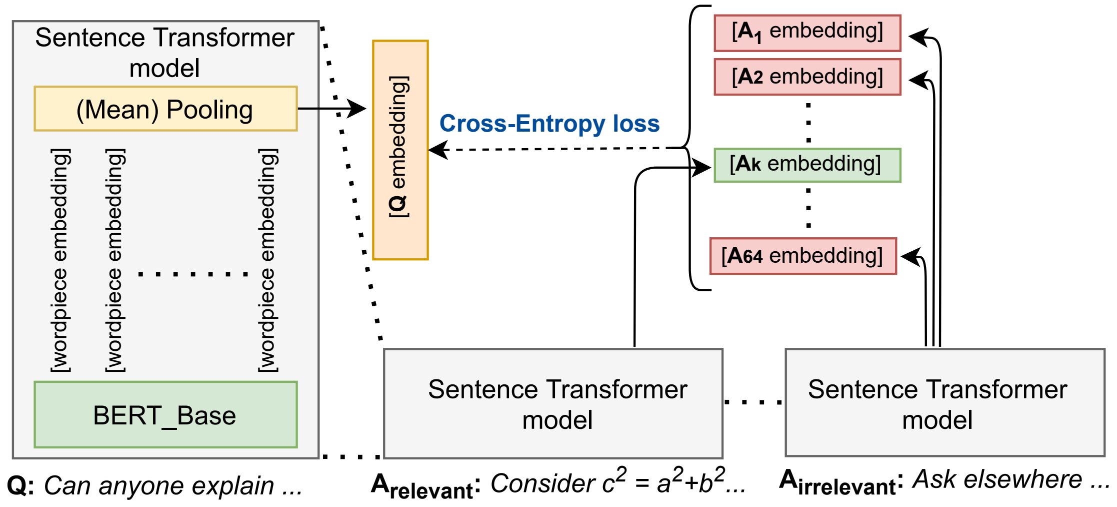


Figure 5: CompuBERT model and Multiple Batch Negatives objective, introduced by Reimers and Gurevych: [31] CompuBERT averages the Wordpiece embeddings [32] into a single representation of 768 floats, that is used similarly to classification: The model is trained to minimize Cross-entropy of softmax on the produced embedding, where an accepted answer holds expected value of one, and a random selection of other accepted answers holds expected value of zero.

We experiment with multiple objectives that are relevant for information retrieval, such as Cosine Contrastive loss on pairs of question and answer, with different representations of formulae part and different selection of positive and negative samples of answers paired with given question. We conclude with a selection of a Multiple Negatives Ranking Loss on a LaTeX math representation, with a selection positive and negative anchors selected as we describe below.

This approach has shown to reach state-of-the-art results on an identification of duplicate questions in Quora,where the unified representations of questions are fine-tuned to be used to classify whether two questions are a duplicate of each other or not.

3.4.2. Model Training

In our approach, we iterate over questions and for each, we pick the positive anchor as the accepted answer of given question, and a batch of negative anchors as random accepted answers of other questions. We skip the questions with no accepted answer. Then, we include the positive anchor into the batch of the negative ones on a random position and we train the system to identify the position of the true accepted answer as a softmax of the embedded representation.

Setting the batch of the fixed size, we adjust the weights of the whole Transformer network by Cross-Entropy loss, as shown on Figure 5. In essence, this approach is similar to a traditional classification training of neural networks.

Specifically, we set a batch size of 64, meaning the system is required to pick out the accepted answer out of the 64 provided ones. Reimers and Gurevych [31] report that increasing the batch size as high as possible shows to improve a quality of the system, hence this is the biggest batch we can fit into our 15 GB of GPU memory. We measure a significant validation difference depending on the math representation, and we find the LaTeX representation to work the best. Additionally, by an example of BART [33], we prepend the contents of questions and answers by “Question: ” and “Answer: ” respectively, so that the system can recognize the kind of item it aims to represent.

3.4.3. Results

On the test subset, CompuBERT achieved 0.264 nDCG′, which is the fourth best result of all our individual systems. On the judgements for task 1 of ARQMath 2021, CompuBERT achieved 0.262 nDCG′, which is the third best result of all our individual systems.

The training of the CompuBERT model took 20 hours. The subsequent inference and indexing of document vectors took another 1:27:57 hours. The average query time was 4.9 seconds with a minimum of 3.3 seconds for topic A.2921 and a maximum of 5.3 seconds for topic A.20.

> “If what you seek is Truth, there is one thing you must have above all else.” “I know. An overwhelming passion for it.” “No. An unremitting readiness to admit you may be wrong.”
> Anthony de Mello

3.5. Other Systems (MSM – MH, LM, MP, JK, AM, and VS)

The remaining systems of the MSM team in the descending order of their nDCG′ on the test subset, are MH, LM, MP, JK, AM, and VS. In this section, we will briefly summarize their architectures and results. Our experimental code is available online: MH[14], LM[15], MP[16], JK[17], AM[18], and VS[19].

3.5.1. Configuration

In our preprocessing, everyone used the LaTeX representation of math formulae and TF-IDF except for AM, who used the prefix notation for the representation of math formulae and BM25.

All systems were unsupervised except for MP, who optimized the hyperparameters of Roccio’s pseudo-relevance feedback and pivoted document length normalization [27].

3.5.2. Results

On the test subset, our systems achieved between 0.127 and 0.268 nDCG′. On the judgements for task 1 of ARQMath 2021, our systems achieved between 0.066 and 0.159 nDCG′.

The average query time was 22.0 seconds with a minimum of 2.9 seconds by JK for topic A.264 and a maximum of 329.5 seconds by AM for topic A.291.

[14] https://colab.research.google.com/drive/1f726gsoitMqrBeA_loRoDOceMjad8GqW
[15] https://colab.research.google.com/drive/1JUkdLZRF7Qvr7uusg56bQdsnozPhE6mn
[16] https://colab.research.google.com/drive/1iW7qonWsGzjTu8c7R2Ue8qaFVHakJkC3
[17] https://colab.research.google.com/drive/1rcHo2AsJO-XBTd5blRd6ROL4sgIyR2A1
[18] https://colab.research.google.com/drive/1t1ZtuamWdUERcevzSMGEF0s2WtLu9zwo
[19] https://colab.research.google.com/drive/19-LfEQlNwkWvngPkK06Ys-6xrcqjY7Qg

“And as for you all, I will make your reward great as though you had accomplished all the work.” Chapters of the Fathers (Pirkei Avot, 2:2)

4. Ensemble Systems

Different MIR systems can agree on a small portion of the most relevant documents, reflecting different ‘points of view‘ on the search problem. Depending on dozens of parameters, each individual system will miss the great majority of relevant documents. With ensembling and voting techniques, we can combine the strengths of different systems to produce more accurate results. Historically, there is a long tradition of boosting, [34, 35], ensembling [36], data fusion [37] and voting approaches [38, 39] in the information retrieval research.

A successful ensemble requires sufficient *diversity* of the individual systems, and math-aware systems are not exceptional in this behaviour. [40] Using Spearman’s ρ and the silhouette score (see Section 2.4 on page 6), we have clustered all non-baseline primary submissions to task 1 of ARQMATH 2020 except zbMATH who retrieved only a single answer for every topic. Figure 6 on the next page shows that we have received only three clusters of systems, which indicates limited diversity. We have also clustered all our submissions to task 1 of ARQMATH 2021. Figure 7 on the following page shows that we have received six clusters of systems, which indicates a notable increase in diversity. In Figure 8 on page 16, we show that systems from the six clusters have different strengths and weaknesses: For example, COMPUBERT receives the most consistent results across both text- and math-based topics, excels at short topics, but its performance deteriorates for long topics that don’t fit into its context window. By contrast, TF-IDF- and BM25-based systems excel at text-based topics, but their performance deteriorates for math-based topics. Systems that model soft matches, such as the SCM, can exploit both short and long topics. Our experimental code for figures 7 and 8 is available online.[20, 21, 22]

We have implemented, computed, and submitted three ensembling techniques: IBC described in Section 4.2 on page 16, WIBC in Section 4.3 on page 17, and RBC in Section 4.4 on page 18. We have also implemented and computed one out-of-competition ensembling technique: RRF described in Section 4.1 on the following page. We have used our techniques to ensemble the result lists of all ten systems of the MIRMU and MSM teams. In this section, we will describe our ensembling techniques and their results on task 1 of ARQMATH 2021. To show that strength lies in diversity rather than numbers, we also report NDCG′ using randomly selected systems from the six clusters in Figure 7 as an ablation study.

To estimate the performance of our ensemble systems in the absence of judgements for task 1 of ARQMATH 2021, we ensembled all non-baseline primary submissions for task 1 of ARQMATH 2020 for performance estimation and we report NDCG′ on the test subset in addition to NDCG′ on the now-available judgements for task 1 of ARQMATH 2021.

[20] https://colab.research.google.com/drive/1WiuF3oxrFQS387ouly4IrKt1j-Kz5Ucv

[21] https://colab.research.google.com/drive/15A6Qalprjhxi0CpiCQhcYy7N8Q4ryRag

[22] https://colab.research.google.com/drive/13JhPTfe57xVBHMway-594nGgSl8mpCjy

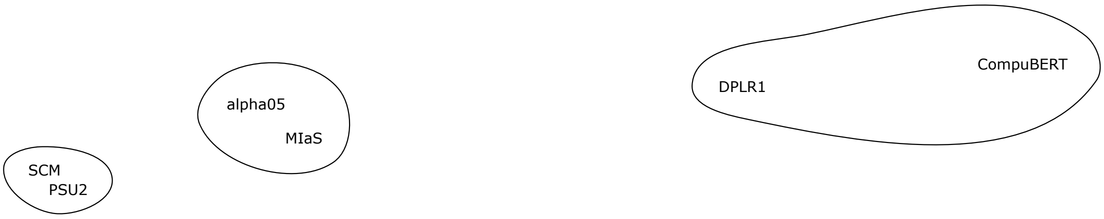


Figure 6: A clustering of all primary submissions to task 1 of ARQMATH 2020 except zbMATH who retrieved only a single answer for every topic. Maximizing the silhouette score produces only three clusters, which indicates small diversity.

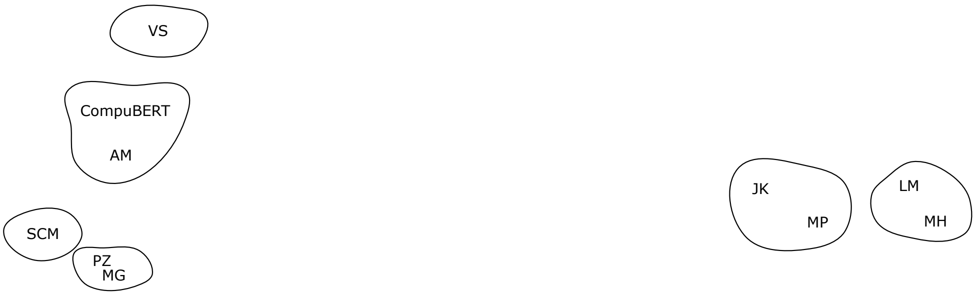


Figure 7: A clustering of all our submissions to task 1 of ARQMATH 2021. Maximizing the silhouette score produces six clusters, which indicates greater diversity compared to Figure 6.

4.1. Non-Weighted Ensemble Baseline (MIRMU – RRF)

The reciprocal rank fusion (RRF) is an ensembling technique, which was shown by Cormack et al. [41] to outperform Condorcet and individual rank learning methods on the LETOR 3 dataset. Our experimental code is available online.[23]

4.1.1. Configuration

The RRF has the parameter k, which mitigates the impact of high rankings by outlier systems. Before applying the RRF to the test subset, we first optimized the value of k on the bigger-train subset, and we received the optimal value $k = 644$. Before applying the RRF to the topics for task 1 of ARQMATH 2021, we first optimized the value of k on the test subset, and we received the optimal value $k = 275$. See Figure 9 on the next page for detailed results of the optimization.

4.1.2. Results

On the test subset, RRF received 0.464 NDCG′, which is the second best result of all our systems, tied with BM25$^+$. On the judgements for task 1 of ARQMATH 2021, RRF achieved 0.309 NDCG′, which is the third best result of all our systems. Ensembling only six systems as part of our ablation study increased NDCG′ from 0.309 to 0.313. Ensembling all non-baseline primary

[23] https://colab.research.google.com/drive/187dVasy8dkKp-JJoLt_VCh_m6HP69NjP

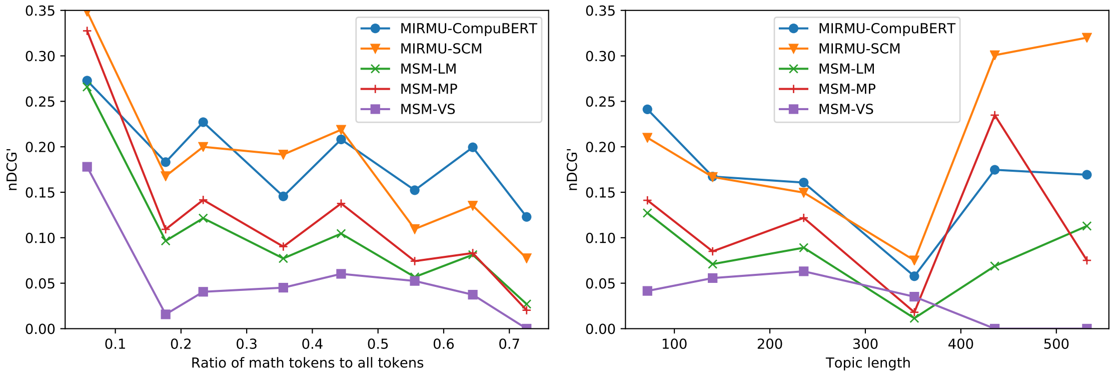


Figure 8: Per-topic nDCG′ (on the y axis) of randomly selected systems from the six clusters in Figure 7 on the topics for task 1 of ARQMath 2021, which we place on the x axis according to the ratio of math tokens to all tokens (left) and according to the topic length in tokens (right) in the prefix math representation.

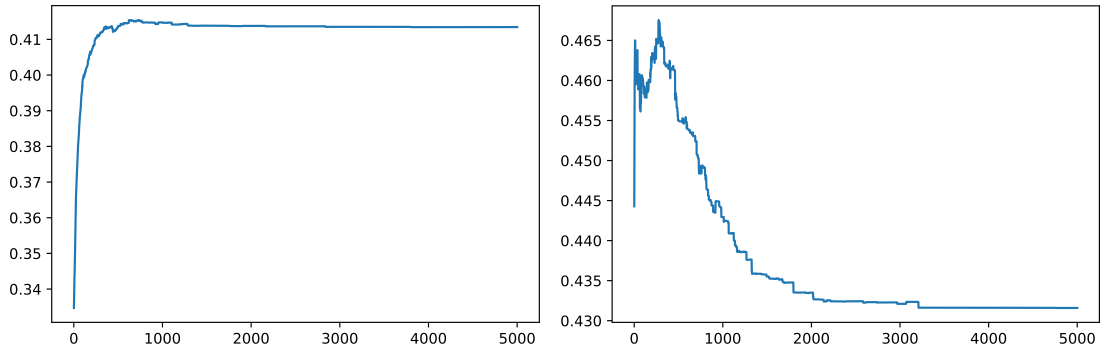


Figure 9: The results of optimizing the parameter k (on the x axis) of the RRF ensemble baseline using nDCG′ (on the y axis) on the bigger-train subset using all non-baseline primary submissions to task 1 of ARQMath 2020 (left) and on the test subset using all ten systems of the MIRMU and MSM teams (right). The optimal values are $k = 644$ on the left and $k = 275$ on the right.

submissions increased nDCG′ from 0.309 to 0.556: *the best result reported in the ARQMath 2021 competition.*

On top of the time to produce the result lists of the ensembled systems, the average query time of RRF was 0.07 seconds with a minimum of 0.02 seconds for topic A.238 and a maximum of 0.59 seconds for topic A.207.

4.2. Non-Weighted Ensemble (MIRMU – IBC)

At ARQMath 2020, we introduced a simple parameter-free algorithm (IBC) for ensembling an arbitrary number of result lists into a single result list, which achieved the best nDCG′ on task 1 of ARQMath 2020 of all our individual systems. [9, Section 7] When we used our algorithm to ensemble the result lists all non-baseline primary submissions to task 1 of ARQMath 2020, we received *the highest* NDCG′ *in the competition* (0.419). Our experimental code is available online.[24]

4.2.1. Configuration

Majority judgement is a single-winner election method by Balinski and Laraki [42], which elects the candidate with the highest median rating. In our IBC ensembling technique, the candidates are all answers in the M-SE collection and their ratings are computed as $^{1000-\text{rank}}/_{1000}$ with ranks taken from the result lists of the individual systems. Ties between several winners are first broken by selecting a random rating out of a uniform distribution of all ratings. Further ties are broken randomly. [9, Section 7]. The result list of IBC consists of the first 1,000 iteratively elected winners.

4.2.2. Results

On the test subset, IBC received 0.452 NDCG′, which is the fourth best result of all our systems. On the judgements for task 1 of ARQMATH 2021, IBC achieved 0.286 NDCG′, which is also the fourth best result of all our systems. Ensembling only six systems as part of our ablation study increased NDCG′ from 0.286 to 0.312. Ensembling all non-baseline primary submissions increased NDCG′ from 0.286 to 0.514: *the second best result reported in the ARQMATH 2021 competition.*

On top of the time to produce the result lists of the ensembled systems, the average query time of IBC was 0.02 seconds with a minimum of 0.01 seconds for topic A.238 and a maximum of 0.24 seconds for topic A.234.

4.3. Weighted Ensemble (MIRMU – WIBC)

Our IBC ensembling technique assumes that all systems are equally trustworthy and qualified in their rating of the answers. Our WIBC ensembling technique assigns weights to the individual systems. Our experimental code is available online.[25]

4.3.1. Configuration

Instead of electing the candidate with the highest median rating, WIBC elects the candidate with the highest *weighted* median rating. Instead of breaking ties by selecting a random rating out of a uniform distribution of all ratings, we select a random rating out of a *weighted* uniform distribution. We use weights provided by our RBC ensembling technique from in Section 4.4 on the following page.

4.3.2. Results

On the test subset, WIBC received 0.456 NDCG′, which is the third best result of all our systems, slightly below the RRF ensemble baseline. On the judgements for task 1 of ARQMATH 2021, WIBC achieved 0.332 NDCG′, which is the best result of all our systems. Ensembling only six systems

[24] https://colab.research.google.com/drive/1qEfMSE6GgcgF97U7LTx2xEXgt6y74GmV

[25] https://colab.research.google.com/drive/1TZ9IXaQQ28GdzESjKbPDhg6ri5RDINgu

as part of our ablation study slightly decreased nDCG′ from 0.332 to 0.327, indicating that WIBC can utilize both diversity *and* redundancy.

On top of the time to produce the result lists of the ensembled systems, the average query time of WIBC was 0.2 seconds with a minimum of 0.1 seconds for topic A.270 and a maximum of 0.5 seconds for topic A.268.

4.4. Ensembling by Regression (MIRMU – RBC)

Both our IBC and WIBC ensembling techniques decide on the *rank* of the answers from the M-SE collection. In contrast, our RBC ensembling technique directly estimates the relevance of an answer. Our experimental code is available online.[26]

4.4.1. Configuration

First, we trained a number of regression model (linear, SGD, ridge, Bayesian ridge, SVR, kNN, PLS, MLP) to predict the gain of train judgements from the ranks in the result lists of the individual systems. Secondly, we selected the best regression model (linear) using the validation subset. For the performance estimation of RBC, we produced a result list by taking the 1,000 answers with the highest predicted gain for each topic in the test subset. For the performance estimation of WIBC, we used the coefficients of the regression model as system weights.

For the submission of RBC to ARQMATH 2021, we retrained the best regression model to predict the gain of test judgements from the ranks in the result lists of our individual systems. This is necessary, because our regression model has only been trained on the non-baseline primary submissions for task 1 of ARQMATH 2020, nor our ten systems of the MIRMU and MSM teams, and the only subset that had not yet been seen by our ten systems was the test subset. For the submission of WIBC to ARQMATH 2021, we used the coefficients of the retrained regression model as system weights.

4.4.2. Results

On the test subset, RBC received 0.551 nDCG′, which is the best result of all our systems. On the judgements for task 1 of ARQMATH 2021, RBC achieved 0.322 nDCG′, which is the second best result of all our systems, slightly below WIBC. Ensembling only six systems as part of our ablation study increased nDCG′ from 0.322 to 0.328.

On top of the time to produce the result lists of the ensembled systems, the average query time of RBC was 0.04 seconds with a minimum of 0.02 seconds for topic A.201 and a maximum of 0.06 seconds for topic A.212.

[26] https://colab.research.google.com/drive/1cYSl1AymNdZSdjGB20KhemWS0c-HHCD_

“Meaning is only found when you go beyond meaning.”
Anthony de Mello

5. Results

Figure 10 shows that on the judgements for task 1 of ARQMATH 2021, our Ensemble systems received the best NDCG′ out of all our systems.

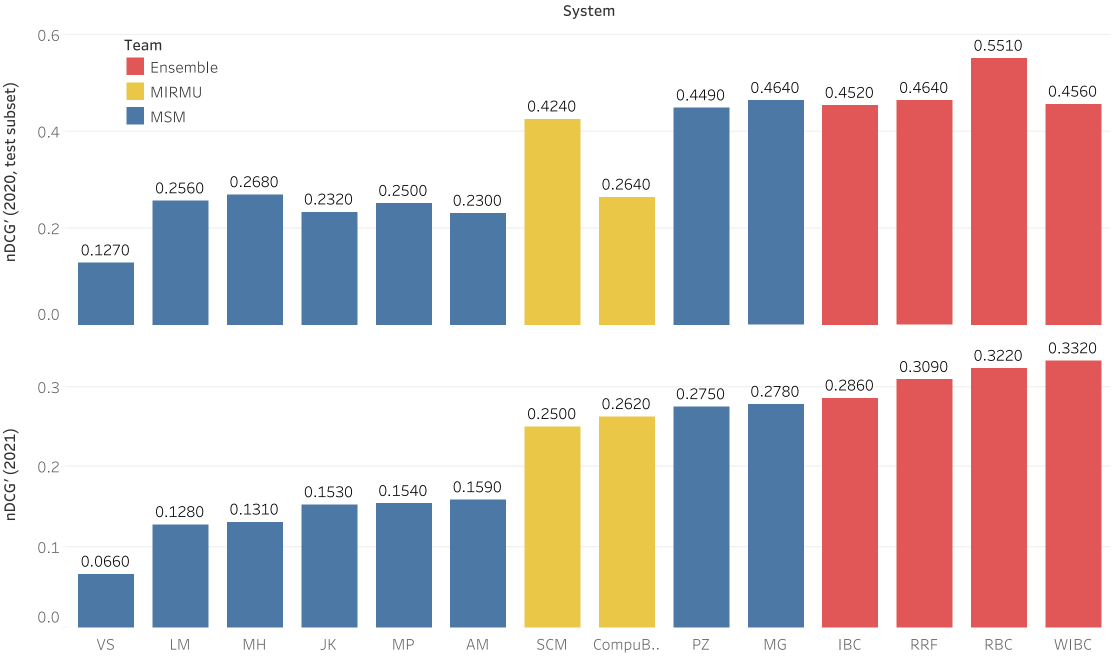


Figure 10: The NDCG′ of our ten individual systems and our four ensemble methods on the test subset (top) and on the judgements for task 1 of ARQMATH 2021 (bottom).

Both on the test set and on the judgements for task 1 of ARQMATH 2021, there is a large gap between the three systems that enriched their answers with the text of their parent questions (SCM, PZ, MG) and the remaining individual systems.

Unlike for other systems, the NDCG′ for COMPUBERT does not significantly decrease from the task subset to the judgements for task 1 of ARQMATH 2021. Due to the unique architecture of COMPUBERT, we theorize that this is because the top 1,000 results of COMPUBERT contain relevant answers, which have not been annotated for ARQMATH 2020.

Figure 11 on the following page shows that the PZ system based on the Anserini library [22] is not only very accurate, but also among our three fastest systems in terms of preprocessing and indexing, and our fastest system in terms of the average search time.

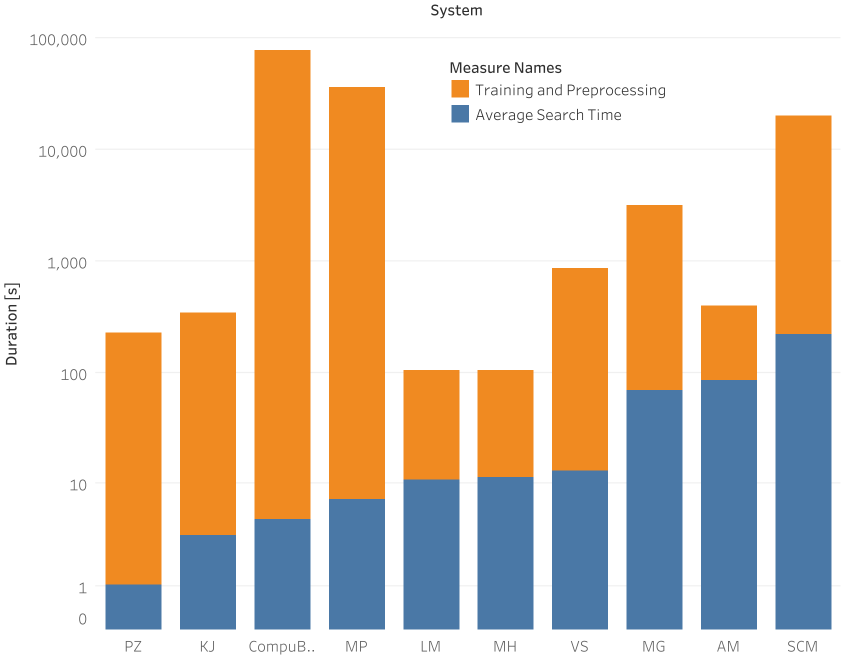


Figure 11: Average query time and the training and preprocessing of the individual systems (log scale).

> “Wisdom tends to grow in proportion to one’s awareness of one’s ignorance.”
> Anthony de Mello

6. Conclusion and Future Work

Motivated by our curiosity to answer our ten research questions, we have introduced ten diverse math-aware information systems. We have data-evidenced using collected systems and available ground truth that good voting strategies of ensembles of baseline systems have the capacity to outperform individual systems.

From our experiments, we formulate ten commandments for math-aware question answering tasks like ARQMath on Math Stack Exchange:

C1: diversity of systems Use as diverse and as many different systems as you can.

C2: diversity of topics Bear in mind the specificity, diachronicity, context of topics, together with relevance judgements of similar topics. If available, take the seeker history and personality into account to disambiguate the information need.

C3: ensembling Diversity is powerful. Clever ensembling of different points of view into account is efficient approach to get better findings and decisions.

C4: ensemble voting strategies Voting strategies are important to weight different aspects of individual systems. It pays off to choose the appropriate ensembling techniques to consistently get better results.

C5: representation Representation matters. The better semantic metric we could design for text and math formulae, the better capture of the topic and answer meaning, and the better the performance!

C6: attention Attention is not all you need, but attention-based models have huge capacity to learn both text and formulae representations and short inferences.

C7: performance Good system performance is a bonus to speed up the development and fine-tuning. Precompute as much as possible, and make the indexes sorted according the semantics. There is always a space for late optimization.

C8: canonicalization Give attention to the diversity of language and mathematical notation and use measures of the semantic similarity as much as you can.

C9: inference To tackle the exponential growth of knowledge, embrace inference into your (deep) models.

C10: explainability Insight, understanding the whole system and explainability for results matters not only to users for result evidence, but also for fine-tuning and system optimization.

Setting the tools, computing environment, and datasets are crucial for studying and researching complex information retrieval methods. [43]. Thanks to the open source tools, ARQMATH competition datasets, and the production environment set at the Faculty of Informatics, Masaryk University, ten student systems were prototyped. We were able to set up a framework for evaluation of not only varieties of individual MIR systems and approaches, but also of their voting and ensembles. This sets the ground to speed up our understanding of innermost features of MIR systems and paves the road to the better fulfilment of information needs of math-aware problem searchers.

Diversity rulez! Looking at complex problem from diverse viewpoints is a good thing and so is wise compounding and merging diverse approaches and their results!

Acknowledgments

First author's work was graciously funded by the South Moravian Centre for International Mobility as a part of the Brno Ph.D. Talent project. We also thank the three anonymous reviewers for their insightful comments. We extend our gratitude to the ARQMATH 2021 organizers for keeping the research of math information retrieval aflame.

References

[1] M. Líška, P. Sojka, M. Růžička, Similarity Search for Mathematics: Masaryk University team at the NTCIR-10 Math Task, in: N. Kando, K. Kishida (Eds.), Proc. of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NII, Tokyo, Japan, Tokyo, 2013, pp. 686–691. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/06-NTCIR10-MATH-LiskaM.pdf.

[2] B. Mansouri, A. Agarwal, D. W. Oard, R. Zanibbi, Advancing Math-Aware Search: The ARQMath-2 Lab at CLEF 2021, in: D. Hiemstra, M. Moens, J. Mothe, R. Perego, M. Potthast, F. Sebastiani (Eds.), Advances in Information Retrieval – 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II, volume 12657 of *LNCS*, Springer, 2021, pp. 631–638. doi:10.1007/978-3-030-72240-1_74.

[3] B. Mansouri, R. Zanibbi, D. W. Oard, A. Agarwal, Overview of ARQMath-2 (2021): Second CLEF Lab on Answer Retrieval for Questions on Math (Working Notes Version), 2021.

[4] P. Sojka, M. Líška, The Art of Mathematics Retrieval, in: Proceedings of the ACM Conference on Document Engineering, DocEng 2011, Association of Computing Machinery, Mountain View, CA, USA, 2011, pp. 57–60. doi:10.1145/2034691.2034703.

[5] P. Sojka, M. Růžička, V. Novotný, MIaS: Math-Aware Retrieval in Digital Mathematical Libraries, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18), ACM, Torino, Italy, 2018, pp. 1923–1926. doi:10.1145/3269206.3269233.

[6] D. Lupták, V. Novotný, M. Štefánik, P. Sojka, Ensembling Ten Math Information Retrieval Systems: MIRMU and MSM at ARQMath 2021, in: F. Kamareddine, C. Sacerdoti-Coen (Eds.), Intelligent Computer Mathematics, CICM 2021, Springer International Publishing Switzerland, Timisoara, Romania, 2021. doi:10.1007/978-3-030-81097-9.

[7] R. Zanibbi, D. W. Oard, A. Agarwal, B. Mansouri, Overview of ARQMath 2020: CLEF Lab on Answer Retrieval for Questions on Math, in: A. Arampatzis, E. Kanoulas, T. Tsikrika, S. Vrochidis, H. Joho, C. Lioma, C. Eickhoff, A. Névéol, L. Cappellato, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction, Springer International Publishing, Cham, 2020, pp. 169–193. doi:10.1007/978-3-030-58219-7_15.

[8] P. Sojka, V. Novotný, E. F. Ayetiran, D. Lupták, M. Štefánik, Quo Vadis, Math Information Retrieval, in: Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2019, 2019, pp. 117–128. URL: https://nlp.fi.muni.cz/raslan/2019/paper11-sojka.pdf.

[9] V. Novotný, P. Sojka, M. Štefánik, D. Lupták, Three is Better than One, in: CEUR Workshop Proceedings: ARQMath task at CLEF conference, volume 2696, CEUR-WS, Thessaloniki, Greece, 2020, pp. 1–30. URL: http://ceur-ws.org/Vol-2696/paper_235.pdf.

[10] B. Mansouri, R. Zanibbi, D. W. Oard, Learning to Rank for Mathematical Formula Retrieval, in: Proceedings of the 2021 ACM SIGIR international conference on theory of Information Retrieval, 2021. doi:10.1145/3404835.3462956.

[11] K. Davila, R. Zanibbi, Layout and Semantics: Combining Representations for Mathematical Formula Search, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 1165–1168. doi:10.1145/3077136.3080748.

[12] D. Formánek, M. Líška, M. Růžička, P. Sojka, Normalization of Digital Mathematics Library Content, in: J. Davenport, J. Jeuring, C. Lange, P. Libbrecht (Eds.), 24th OpenMath Workshop, 7th Workshop on Mathematical User Interfaces (MathUI), and Intelligent Computer Mathematics Work in Progress, number 921 in CEUR Workshop Proceedings, Aachen, 2012, pp. 91–103. http://ceur-ws.org/Vol-921/wip-05.pdf.

[13] D. Ginev, arXMLiv:08.2019 dataset, an HTML5 conversion of arXiv.org, 2019. URL: https://sigmathling.kwarc.info/resources/arxmliv-dataset-082019/, SIGMathLing – Special Interest Group on Math Linguistics.

[14] J. R. Landis, G. G. Koch, The measurement of observer agreement for categorical data, Biometrics (1977) 159–174.

[15] B. Mansouri, ARQMath: Additional Information on Topics 1 and 2, 2021. URL: https://drive.google.com/drive/u/1/folders/1rE9f_xheR1RRKQYLlfYzV0-kDXEnRNae, visited on 2021-07-03.

[16] T. Sakai, N. Kando, On information retrieval metrics designed for evaluation with incomplete relevance assessments, Information Retrieval 11 (2008) 447–470.

[17] C. Spearman, The Proof and Measurement of Association between Two Things, The American Journal of Psychology 15 (1904) 72–101. URL: http://www.jstor.org/stable/1412159.

[18] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Journal of computational and applied mathematics 20 (1987) 53–65.

[19] Y. Lv, C. Zhai, A Log-Logistic Model-Based Interpretation of TF Normalization of BM25, in: R. Baeza-Yates, A. P. de Vries, H. Zaragoza, B. B. Cambazoglu, V. Murdock, R. Lempel, F. Silvestri (Eds.), Advances in Information Retrieval, Springer, Berlin, Heidelberg, 2012, pp. 244–255. doi:10.1007/978-3-642-28997-2_21.

[20] A. Trotman, A. Puurula, B. Burgess, Improvements to BM25 and language models examined, in: Proceedings of the 2014 Australasian Document Computing Symposium, ADCS '14, ACM, New York, NY, USA, 2014, pp. 58–65. doi:10.1145/2682862.2682863.

[21] J. Lin, X. Ma, S. Lin, J. Yang, R. Pradeep, R. Nogueira, Pyserini: An Easy-to-Use Python Toolkit to Support Replicable IR Research with Sparse and Dense Representations, CoRR abs/2102.10073 (2021). URL: https://arxiv.org/abs/2102.10073.

[22] P. Yang, H. Fang, J. Lin, Anserini: Enabling the Use of Lucene for Information Retrieval Research, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '17, ACM, New York, NY, USA, 2017, pp. 1253–1256. doi:10.1145/3077136.3080721.

[23] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with GPUs, CoRR abs/1702.08734 (2017). URL: https://arxiv.org/abs/1702.08734.

[24] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information processing & management 24 (1988) 513–523.

[25] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, D. Pinto, Soft similarity and soft cosine measure: Similarity of features in vector space model, Computación y Sistemas 18 (2014) 491–504.

[26] R. Řehůřek, P. Sojka, Software Framework for Topic Modelling with Large Corpora, in: Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks, ELRA, Valletta, Malta, 2010, pp. 45–50. doi:10.13140/2.1.2393.1847.

[27] A. Singhal, G. Salton, M. Mitra, C. Buckley, Document length normalization, Information Processing & Management 32 (1996) 619–633.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is All you Need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[29] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, T.-Y. Liu, Understanding and improving transformer from a multi-particle dynamic system point of view, arXiv abs/1906.02762 (2019). URL: https://arXiv.org/abs/1906.02762.

[30] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, in: Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers), ACL, Berlin, Germany, 2016, pp. 1715–1725. doi:10.18653/v1/P16-1162.

[31] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in NLP and the 9th International Joint Conference on NLP (EMNLP-IJCNLP), ACL, Hong Kong, China, 2019, pp. 3982–3992. URL: https://www.aclweb.org/anthology/D19-1410. doi:10.18653/v1/D19-1410.

[32] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv 1810.04805 (2018). URL: https://arXiv.org/abs/1810.04805.

[33] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, arXiv e-prints (2019) arXiv:1910.13461. URL: https://ui.adsabs.harvard.edu/abs/2019arXiv191013461L. arXiv:1910.13461.

[34] A. Gulin, I. Kuralenok, D. Pavlov, Winning The Transfer Learning Track of Yahoo!'s Learning To Rank Challenge with YetiRank, in: O. Chapelle, Y. Chang, T.-Y. Liu (Eds.), Proceedings of the Learning to Rank Challenge, volume 14 of *Proceedings of Machine Learning Research*, PMLR, Haifa, Israel, 2011, pp. 63–76. URL: http://proceedings.mlr.press/v14/gulin11a.html.

[35] Q. Wu, C. J. C. Burges, K. M. Svore, J. Gao, Adapting boosting for information retrieval measures, Information Retrieval 13 (2010) 254–270. doi:10.1007/s10791-009-9112-1.

[36] Y. Wang, I.-C. Choi, H. Liu, Generalized Ensemble Model for Document Ranking in Information Retrieval, Computer Science and Information Systems 14 (2017) 123––151. doi:10.2298/csis160229042w.

[37] R. Nuray, F. Can, Automatic Ranking of Information Retrieval Systems Using Data Fusion, Information Processing and Management 42 (2006) 595–614. doi:10.1016/j.ipm.2005.03.023.

[38] M. Mosbah, B. Boucheham, Majority Voting Re-ranking Algorithm for Content Based-Image Retrieval, in: E. Garoufallou, R. J. Hartley, P. Gaitanou (Eds.), Metadata and Semantics Research, Springer International Publishing, Cham, 2015, pp. 121–131.

[39] A. T. Albaham, N. Salim, Quality Biased Thread Retrieval Using the Voting Model, in: Proceedings of the 18th Australasian Document Computing Symposium, ADCS '13, ACM, New York, NY, USA, 2013, pp. 97–100. doi:10.1145/2537734.2537752.

[40] L. I. Kuncheva, C. J. Whitaker, Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, Machine Learning 51 (2003) 181–207. doi:10.1023/A:1022859003006.

[41] G. V. Cormack, C. L. A. Clarke, S. Buettcher, Reciprocal Rank Fusion Outperforms Condorcet and Individual Rank Learning Methods, in: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, ACM, New York, NY, USA, 2009, pp. 758–759. doi:10.1145/1571941.1572114.

[42] M. Balinski, R. Laraki, A theory of measuring, electing, and ranking, Proceedings of the National Academy of Sciences 104 (2007) 8720–8725. doi:10.1073/pnas.0702634104.

[43] Z. Akkalyoncu Yilmaz, C. L. A. Clarke, J. Lin, A Lightweight Environment for Learning Experimental IR Research Practices, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, ACM, New York, NY, USA, 2020, pp. 2113–2116. doi:10.1145/3397271.3401395.