Combining Sparse and Dense Information Retrieval

Soft Vector Space Model and MathBERTa at ARQMath-3 Task 1 (Answer Retrieval)

Vít Novotný[1], Michal Štefánik[1]

[1]Faculty of Informatics Masaryk University, Botanická 554/68a, 602 00 Brno, Czech Republic

Abstract

Sparse retrieval techniques can detect exact matches, but are inadequate for mathematical texts, where the same information can be expressed as either text or math. The soft vector space model has been shown to improve sparse retrieval on semantic text similarity, text classification, and machine translation evaluation tasks, but it has not yet been properly evaluated on math information retrieval.

In our work, we compare the soft vector space model against standard sparse retrieval baselines and state-of-the-art math information retrieval systems from Task 1 (Answer Retrieval) of the ARQMath-3 lab. We evaluate the impact of different math representations, different notions of similarity between key words and math symbols ranging from Levenshtein distances to deep neural language models, and different ways of combining text and math.

We show that using the soft vector space model consistently improves effectiveness compared to using standard sparse retrieval techniques. We also show that the Tangent-L math representation achieves better effectiveness than LaTeX, and that modeling text and math separately using two models improves effectiveness compared to jointly modeling text and math using a single model. Lastly, we show that different math representations and different ways of combining text and math benefit from different notions of similarity between tokens. Our best system achieves NDCG' of 0.251 on Task 1 of the ARQMath-3 lab.

Keywords

information retrieval, sparse retrieval, dense retrieval, soft vector space model, math representations, word embeddings, constrained positional weighting, decontextualization, word2vec, transformers

1. Introduction

State-of-the-art math information retrieval systems use sparse retrieval techniques that can detect exact key word matches with high precision, but fail to retrieve texts that are semantically similar but use different terminology. This shortcoming is all the more apparent with mathematical texts, where the same information can be expressed in two completely different systems of writing and thought: the natural language and the language of mathematics.

Recently, the soft vector space model of Sidorov et al. [25] made it possible to retrieve documents according to both exact and fuzzy key word matches and has outperformed standard sparse retrieval techniques on semantic text similarity [2], text classification [17], and machine translation evaluation [26] tasks. The soft vector space has been used for math information retrieval in the ARQMath-1 and 2 labs [18, 16]. However, it has not been properly compared

CLEF 2022: Conference and Labs of the Evaluation Forum, September 5–8, 2022, Bologna, Italy

witiko@mail.muni.cz (V. Novotný); stefanik.m@mail.muni.cz (M. Štefánik)
0000-0002-3303-4130 (V. Novotný); 0000-0003-1766-5538 (M. Štefánik)

to sparse retrieval baselines. Furthermore, the soft vector space model makes it possible to use different representations of math, different notions of similarity between key words and symbols, and different ways to combine text and math. However, neither of these possibilities has been previously explored.

In our work, we aim to answer the following four research questions:

1. Does the soft vector space model outperform sparse information retrieval baselines on the math information retrieval task?
2. Which math representation works best with the soft vector space model?
3. Which notion of similarity between key words and symbols works best?
4. Is it better to use a single soft vector space model to represent both text and math or to use two separate models?

The rest of our paper is structured as follows: In Section 2, we describe our system and our experimental setup. In Section 3, we report and discuss our experimental results. We conclude in Section 4 by answering our research questions and summarizing our contributions.

2. Methods

In this section, we describe the datasets we used to train our tokenizers and language models. We also describe how we used our language models to measure similarity between text and math tokens, how we used our similarity measures to find answers to math questions, and how we evaluated our system.

2.1. Datasets

In our experiments, we used the Math StackExchange and ArXMLiv corpora:

Math StackExchange The Math StackExchange collection v1.2 (MSE)[1] provided by the organizers of the ARQMath-2 lab [11, Section 3] contains 2,466,080 posts from the Math StackExchange question answering website in HTML5 with math formulae in LaTeX.

ArXMLiv The ArXMLiv 2020 corpus [3] contains 1,571,037 scientific preprints from ArXiv in the HTML5 format with math formulae in MathML. Documents in the dataset were converted from LaTeX sources and are divided into the following subsets according to the severity of errors encountered during conversion: `no-problem` (10%), `warning` (60%), and `error` (30%).

From the corpora, we produced a number of datasets[2] in different formats that we used to train our tokenizers and language models:

[1] An improved Math Stack Exchange collection v1.3 was made available by the organizers of the ARQMath-3 lab [12, Section 3], which we did not use due to time constraints.

[2] See https://github.com/witiko/scm-at-arqmath3, file 01-prepare-dataset.ipynb.

Text + LaTeX To train text & math language models, we combined MSE with the `no-problem` and `warning` subsets of ArXMLiv. The dataset contains text and mathematical formulae in the LaTeX format surrounded by *[MATH]* and *[/MATH]* tags. To validate our language models, we used a small part of the `error` subset of ArXMLiv and no data from MSE.

Example: *We denote the set of branches with [MATH] B_{0},B_{1},\ldots,B{n} [/MATH] where [MATH] n [/MATH] are the number of branches.*

Text To train text language models, we used the same combinations of MSE and ArXMLiv as in the previous dataset, but now our dataset only contains text with math formulae removed.

Example: *(Graphs of residually finite groups) Assume that and are satisfied. Let be a graph of groups. If is infinite then assume that is continuous.*

LaTeX To train math language models, we used the same combinations of MSE and ArXMLiv subsets as in the previous datasets, but now our dataset only contains formulae in the LaTeX format.

Example: *\begin{pmatrix}1&n\0&1\end{pmatrix}\begin{pmatrix}1&p\0&1\end{pmatrix}*

Tangent-L To train math language models, we used the same combinations of MSE and ArXMLiv subsets as in the previous datasets, but now our dataset only contains formulae in the format used by the state-of-the-art Tangent-L search engine from UWaterloo[3] [14].

Example: *#(start)# #(v!△,/,n,-)# #(v!△,/,n)# #(/,v!l,n,n)# #(/,v!l,n)# #(v!l,!0,nn)# #(v!l,!0)# #(end)#*

2.2. Tokenization

In our system, we used several tokenizers:

- To tokenize text, we used the BPE tokenizer of the `roberta-base` language model[4] [9].
- To tokenize math, we used two different tokenizers for the LaTeX and Tangent-L formats:
  - To tokenize LaTeX, we trained a BPE tokenizer[5] with a vocabulary of size 50,000 on our LaTeX dataset.
  - To tokenize Tangent-L, we strip leading and trailing hash signs (#) from a formula representation and then split it into tokens using the `#\s+#` Perl regex.
- To tokenize text and math in the LaTeX format, we extended the BPE tokenizer of `roberta-base` with the *[MATH]* and *[/MATH]* special tokens and with the tokens recognized by our LaTeX tokenizer.

[3] See https://github.com/fwtompa/mathtuples, git commit 888b3d5 from October 25, 2021.

[4] See https://huggingface.co/roberta-base.

[5] See https://github.com/witiko/scm-at-arqmath3, file 02-train-tokenizers.ipynb.

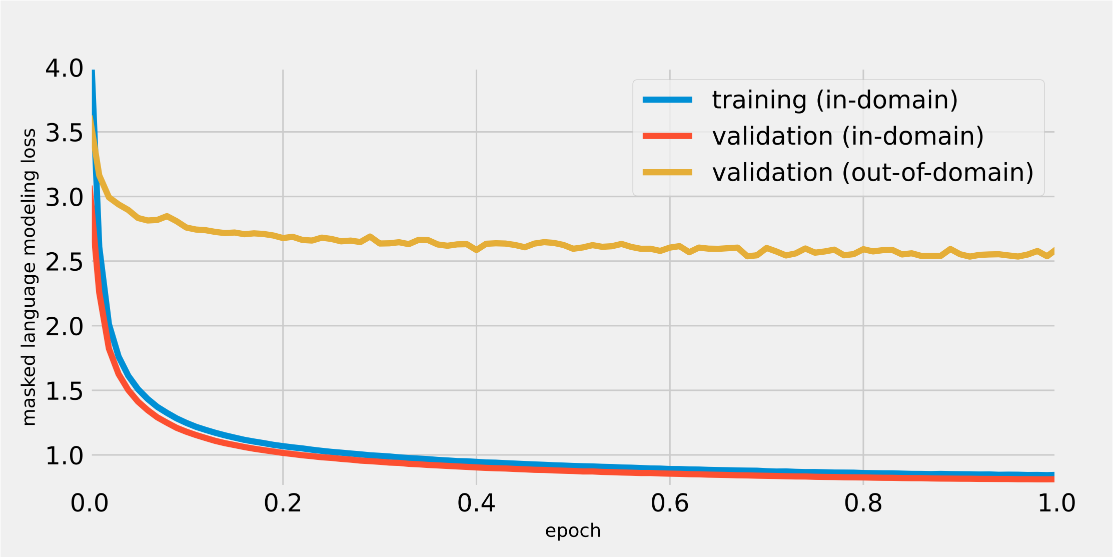

Figure 1: Learning curves of MathBERTa on our text + LaTeX dataset (in-domain) and the European Constitution (out-of-domain). The ongoing descent of in-domain validation loss indicates that the performance of the model improved over time, but has not converged and would benefit from further training. The ongoing descent of out-of-domain validation loss shows that improvements on scientific texts do not come at the price of other non-scientific domains.

2.3. Language Modeling

In our experiments, we used two different types of language models:

Shallow log-bilinear models We trained shallow `word2vec` language models[6] [13] on our text + LaTeX, text, LaTeX, and Tangent-L datasets.

On text documents, a technique known as *constrained positional weighting* has been shown to improve the performance of `word2vec` models on analogical reasoning and causal language modeling [19]. To evaluate the impact of constrained positional weighting on math information retrieval, we trained `word2vec` models both with and without constrained positional weighting for every dataset. For brevity, we refer to `word2vec` with and without constrained positional weighting as *positional* `word2vec` and *non-positional* `word2vec` in the rest of the paper.

Deep transformer models To model text, we used pre-trained `roberta-base` model[7] [9].

Related work shows that accurate domain-specialized representations can be obtained by continuous training, i.e. adaptation, using masked language modeling (MLM) on domain-specific unlabeled texts, in medicine [21], biology [8], and other scientific texts [1]. Previous work [20, 24] performs continuous MLM training on scientific texts, or the math formulae thereof [5]. However, all the aforementioned works treats math as plain

[6] See https://github.com/witiko/scm-at-arqmath3, file 04-train-word2vec.ipynb.

[7] See https://huggingface.co/roberta-base.

text and only few [4] promote math-specific representations in the model adaptation. This motivated us to experiment with adaptation incorporating specific encodings for non-textual expressions.

To model text and math in the LaTeX format, we replaced the tokenizer of `roberta-base` with our text and math tokenizer. Then, we extended the vocabulary of our model with the *[MATH]* and *[/MATH]* special tokens and with the tokens recognized by our LaTeX tokenizer, and we randomly initialized weights for the new tokens. We fine-tuned our model on our text + LaTeX dataset for one epoch using the MLM objective of RoBERTa[8] [9] and a learning rate of 10^{-5} with a linear decay to zero, see the learning curves in Figure 1. We called our model MathBERTa and released it to the HF Model Hub.[9]

2.4. Token Similarity

To determine the similarity of text and math tokens, we first extracted their global representations from our language models:

Shallow log-bilinear models We extracted token vectors from the input and output matrices of our `word2vec` models and averaged them to produce global token embeddings.

Deep transformer models Unlike `word2vec`, transformer models do not contain global representations of tokens, but produce representations of tokens in the context of a sentence. To extract global token embeddings from `roberta-base` and MathBERTa, we decontextualized their contextual token embeddings[10] [26, Section 3.2] on the sentences from our text + LaTeX dataset.

Then, we produced dictionaries of all tokens in our text + LaTeX, text, LaTeX, and Tangent-L datasets,[11] removing all tokens that occurred less than twice in a dataset and keeping only 100,000 most frequent tokens from every dataset. For each dictionary, we produced two types of token similarity matrices[12] that capture the surface-level lexical similarity and the semantic similarity between tokens, respectively:

Lexical similarity We used the method of Charlet and Damnati [2, Section 2.2] to produce similarity matrices using the Levenshtein distance between the tokens.

Semantic similarity We used the method of Charlet and Damnati [2, Section 2.1] to produce similarity matrices using the cosine similarity between the global token embeddings.

For all dictionaries, we produced two matrices using the token embeddings of the positional and non-positional `word2vec` models. For the text and text + LaTeX dictionaries, we also produced an additional matrix using the token embeddings of the `roberta-base` and MathBERTa models, respectively.

[8] See https://github.com/witiko/scm-at-arqmath3, file 03-finetune-roberta.ipynb.

[9] See https://huggingface.co/witiko/mathberta.

[10] See https://github.com/witiko/scm-at-arqmath3, file 05-produce-decontextualized-word-embeddings.ipynb.

[11] See https://github.com/witiko/scm-at-arqmath3, file 06-produce-dictionaries.ipynb.

[12] See https://github.com/witiko/scm-at-arqmath3, file 07-produce-term-similarity-matrices.ipynb.

To ensure sparsity and symmetry of the matrices, we considered only the 100 most similar tokens for each token and we used the greedy algorithm of Novotný [15, Section 3] to construct the matrices. For semantic similarity matrices, we also enforced strict diagonal dominance, which has been shown to improve performance on the semantic text similarity task [17, Table 2].

Finally, to produce token similarity matrices that capture both lexical and semantic similarity between tokens, we combined every semantic similarity matrix with a corresponding lexical similarity matrix as follows:

$$\textbf{Combined similarity} = \alpha \cdot \textbf{Lexical similarity} + (1 - \alpha) \cdot \textbf{Semantic similarity} \quad (1)$$

In our system, we only used the combined token similarity matrices.

2.5. Soft Vector Space Modeling

In order to find answers to math questions, we used sparse retrieval with the soft vector space model of Sidorov et al. [25], using Lucene BM25 [6, Table 1] as the vector space and our combined similarity matrices as the token similarity measure. To address the bimodal nature of math questions and answers, we used the following two approaches:[13]

Joint models To allow users to query math information using natural language and vise versa, we used single soft vector space models to jointly represent both text and math.

As our baselines, we used 1) Lucene BM25 with the text dictionary and no token similarities and 2) Lucene BM25 with the text + LaTeX dictionary and no token similarities.

We also used four soft vector space models with the text + LaTeX dictionary and the token similarity matrices from the positional and non-positional `word2vec` models, the `roberta-base` model, and the MathBERTa model.

Interpolated models To properly represent the different frequency distributions of text and math tokens, we used separate soft vector space models for text and math. The final score of an answer is determined by linear interpolation of the scores assigned by the two soft vector space models:

$$\textbf{Interpolated similarity} = \beta \cdot \textbf{Text similarity} + (1 - \beta) \cdot \textbf{Math similarity} \quad (2)$$

As our baselines, we used Lucene BM25 with the text dictionary and no token similarities interpolated with 1) Lucene BM25 with the LaTeX dictionary and no token similarities and with 2) Lucene BM25 with the Tangent-L dictionary and no token similarities.

We also used four pairs of soft vector space models: two pairs with the text and LaTeX dictionaries and two pairs with the text and Tangent-L dictionaries. In each of the two pairs, one used the token similarity matrices from the positional `word2vec` model and the other used the token similarity matrices from non-positional `word2vec` model.

[13] See https://github.com/witiko/scm-at-arqmath3, file 08-produce-arqmath-runs.ipynb.

For our representation of questions in the soft vector space model, we used the tokens in the title and in the body text. To represent an answer in the soft vector space model, we used the tokens in the title of its parent question and in the body text of the answer. To give greater weight to tokens in the title, we repeated them γ times, which proved useful in ARQMath-2 [16, Section 3.2].

2.6. Evaluation

To evaluate our system, we searched for answers to sets of topics provided by the ARQMath organizers for Task 1 (Answer Retrieval) [28, 11, 12, Section 4.1]. As our retrieval units, we used answers from the MSE dataset.

Effectiveness To determine how well the answers retrieved by our system satisfied the information needs of users, we used the normalized discounted cumulative gain prime (NDCG') evaluation measure [23] on the top 1,000 answers retrieved by our system for each topic. As our ground truth, we used the relevance judgements provided by the ARQMath organizers [28, 11, 12, Section 4.3].

To select the optimal values for parameters α, β, and γ, we used the 148 topics from ARQMath-1 and 2, Task 1 and performed a grid search over values $\alpha \in \{0.0, 0.1, \ldots, 1.0\}$, $\beta \in \{0.0, 0.1, \ldots, 1.0\}$, and $\gamma \in \{1, 2, 3, 4, 5\}$. To estimate the effectiveness of our system, we used the 78 topics from ARQMath-3 Task 1.

Due to time constraints, we hand-picked the parameter values $\alpha = 0.1$, $\beta = 0.5$, and $\gamma = 5$ for our submissions to the ARQMath-3 lab. We report effectiveness for both hand-picked and optimal parameter values, and discuss the robustness of our system to parameter variations.

Efficiency Our system is a prototype written in a high-level programming language with emphasis on correctness over efficiency. Furthermore, we computed our evaluation on a non-dedicated computer cluster with heterogeneous hardware, which made it difficult to meaningfully measure the efficiency of our system. Therefore, we have not measured and do not report the efficiency of our system.

3. Results

In tables 1 and 2, we list effectiveness results with hand-picked parameter values submitted to the ARQMath-3 lab for our joint and interpolated soft vector space models. In tables 3 and 4, we list post-competition effectiveness results with optimized parameter values for our joint and interpolated models. In all tables 1–4, we also list the parameter values the we used.

In Figure 2, we visualize the effectiveness of our baseline models with optimized parameter values and how it is affected by our various extensions.

In Table 5, we compare our post-competition effectiveness results with optimized parameter values to the baselines and the best results from other teams on ARQMath-3 Task 1.

Table 1
Results with hand-picked parameter values submitted to the ARQMath-3 lab for joint soft vector space models on ARQMath-3 Task 1

Model	α	γ	NDCG'
Joint text + LaTeX (MathBERTa)	0.1	5	0.249
Joint text + LaTeX (non-positional `word2vec`)	0.1	5	0.249
Joint text + LaTeX (positional `word2vec`)	0.1	5	0.248
Joint text (`roberta-base`)	0.1	5	0.188

Table 2
Results with hand-picked parameter values submitted to the ARQMath-3 lab for interpolated soft vector space models on ARQMath-3 Task 1

Model	α_1	γ_1	α_2	γ_2	β	NDCG'
Interpolated text + Tangent-L (positional `word2vec`)	0.1	5	0.1	5	0.5	0.257

3.1. Robustness to Parameter Variations

In tables 1–4, the differences between hand-picked and optimized parameter values for joint models are within 0.002 NDCG' except *Joint text (`roberta-base`)*, which improves effectiveness by 0.041 NDCG' by placing more weight on the lexical similarity of tokens (α: $0.1 \rightarrow 0.6$) and by placing less weight on question titles (γ: $5 \rightarrow 2$). This shows that our joint vector space models are relatively robust to parameter variations.

By contrast, optimizing parameter values for the *Interpolated text + Tangent-L (positional `word2vec`)* model improves effectiveness by 0.098 NDCG'. Compared to the hand-picked parameter values, the optimized parameter values place more weight on the lexical similarity for text tokens (α_1: $0.1 \rightarrow 0.7$), use only semantic similarity for math tokens (α_2: $0.1 \rightarrow 0.0$), place less weight on the text in question titles (γ_1: $5 \rightarrow 2$), and place more weight on math over text (β: $0.5 \rightarrow 0.7$).

3.2. Effectiveness of Baselines and Their Extensions

Figure 2 shows that the *Joint text (no token similarities)* baseline receives NDCG' of 0.235. Using `roberta-base` as the source of semantic similarity between text tokens improves effectiveness by 0.012 NDCG', reaching NDCG' of 0.247. By contrast, including also LaTeX math tokens reduces effectiveness by 0.011 NDCG', reaching NDCG' of 0.224, which we attribute to the difficulty to properly represent the different frequency distributions of text and math tokens in a single joint model. However, when we also use either positional `word2vec` or MathBERTa as the source of semantic similarity between text and math tokens, effectiveness improves by 0.025 NDCG', reaching NDCG' of 0.249. Removing the positional weighting from `word2vec` further improves effectiveness by 0.002 NDCG', reaching NDCG' of 0.251, which is the best result among our joint models.

Figure 2 also shows that the *Interpolated text + LaTeX (no token similarities)* baseline receives NDCG' of 0.257. Using non-positional `word2vec` as the source of similarity between text and

Table 3
Post-competition results with optimized parameter values for joint soft vector space models on ARQMath-3 Task 1

Model	α	γ	NDCG’
Joint text + LaTeX (non-positional `word2vec`)	0.6	5	0.251
Joint text + LaTeX (positional `word2vec`)	0.7	5	0.249
Joint text + LaTeX (MathBERTa)	0.6	4	0.249
Joint text (`roberta-base`)	0.6	2	0.247
Joint text (no token similarities)		2	0.235
Joint text + LaTeX (no token similarities)		3	0.224

Table 4
Post-competition results with optimized parameter values for interpolated soft vector space models on ARQMath-3 Task 1

Model	α_1	γ_1	α_2	γ_2	β	NDCG’
Interpolated text + Tangent-L (positional `word2vec`)	0.7	2	0.0	5	0.7	0.355
Interpolated text + Tangent-L (non-positional `word2vec`)	0.6	2	0.0	5	0.7	0.351
Interpolated text + Tangent-L (no token similarities)		2		4	0.6	0.349
Interpolated text + LaTeX (positional `word2vec`)	0.7	2	1.0	5	0.6	0.288
Interpolated text + LaTeX (non-positional `word2vec`)	0.6	2	1.0	5	0.6	0.288
Interpolated text + LaTeX (no token similarities)		2		5	0.6	0.257

math tokens improves effectiveness by 0.031 NDCG’, reaching NDCG’ of 0.288. Using positional `word2vec` does not further improve effectiveness.

The *Interpolated text + Tangent-L (no token similarities)* baseline receives NDCG’ of 0.349. Using non-positional `word2vec` as the source of similarity between text and math tokens improves effectiveness by 0.002 NDCG’, reaching NDCG’ of 0.251. Enabling the positional weighting of `word2vec` further improves effectiveness by 0.004 NDCG’, reaching NDCG’ of 0.355, the best result among all our models.

3.3. Optimized Parameter Values

Tables 3 and 4 show that all joint models and the interpolated models for text place more weight on the lexical similarity of tokens (α and α_1 of either 0.6 or 0.7). Furthermore, all joint and interpolated models for text place equal weight on question titles (γ and γ_1 of 2). By contrast, all joint models for text and math and the interpolated models for math place comparatively higher weight on the math in question titles (γ and γ_2 between 3 and 5). This indicates that math in question titles is more informative than text in question titles. Additionally, all interpolated models for LaTeX math only used the lexical similarity of tokens (α_2: 1.0). By contract, all interpolated models for Tangent-L math only used the semantic similarity of tokens (α_2: 0.0). Lastly, all interpolated models place more weight on text over math (β of either 0.6 or 0.7).

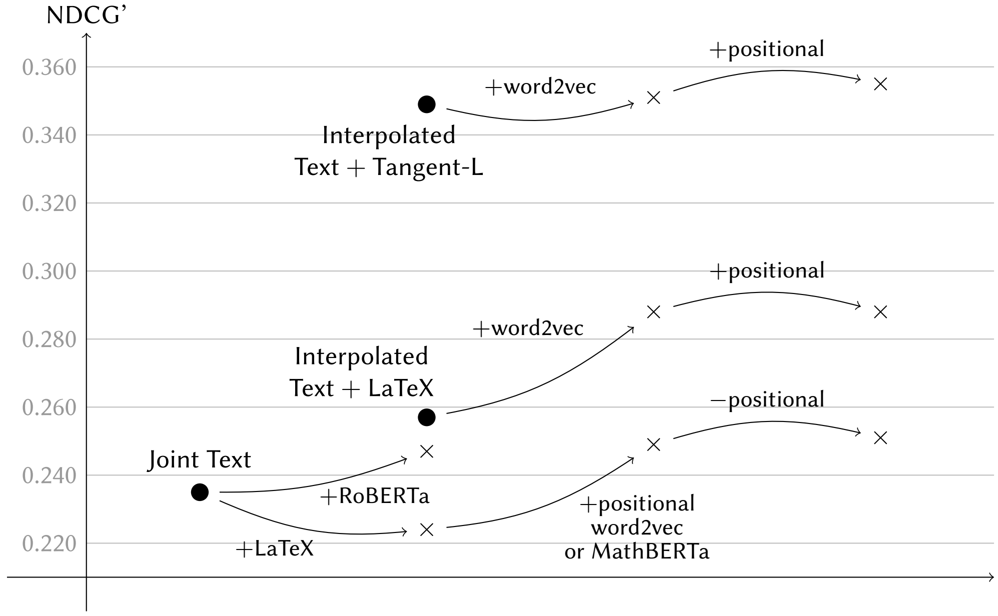

Figure 2: The extensions of the baseline soft vector space models and their impact on the effectiveness with optimized parameter values

3.4. Comparison to Results from Other Teams

Our submission to the ARQMath-3 lab with hand-picked parameter values placed last in effectiveness among the teams that participated in Task 1. However Table 5 shows that our *Interpolated text + Tangent-L (positional `word2vec`)* model with optimized parameter values achieves better effectiveness than the best *SVM-Rank* system from the DPRL team [10] by 0.011 NDCG'.

4. Conclusion

In this paper, we aimed to answer the following research questions:

1. Does the soft vector space model outperform sparse information retrieval baselines on the math information retrieval task?
2. Which math representation works best with the soft vector space model?
3. Which notion of similarity between key words and symbols works best?
4. Is it better to use a single soft vector space model to represent both text and math or to use two separate models?

Using our experimental results, we can answer our research questions as follows:

Table 5
Comparison of our post-competition effectiveness results with the baselines and the best results from other teams on ARQMath-3 Task 1

Model	NDCG’
fusion_alpha05 from approach0 [29]	0.508
Ensemble_RRF from MSM [27]	0.504
MiniLM+RoBERTa from MIRMU [27]	0.498
L8_a018 from MathDowsers [7]	0.474
math_10 from TU_DBS [22]	0.436
Interpolated text + Tangent-L (positional `word2vec`)	0.355
Interpolated text + Tangent-L (non-positional `word2vec`)	0.351
Interpolated text + Tangent-L (no token similarities)	0.349
Interpolated text + LaTeX (positional `word2vec`)	0.288
Interpolated text + LaTeX (non-positional `word2vec`)	0.288
SVM-Rank from DPRL [10]	0.283
TF-IDF (Terrier) baseline [12]	0.272
Interpolated text + LaTeX (no token similarities)	0.257
Joint text + LaTeX (non-positional `word2vec`)	0.251
Joint text + LaTeX (positional `word2vec`)	0.249
Joint text + LaTeX (MathBERTa)	0.249
Joint text (`roberta-base`)	0.247
Joint text (no token similarities)	0.235
TF-IDF (PyTerrier) + TangentS baseline [12]	0.229
Joint Text + LaTeX (no token similarities)	0.224
TF-IDF (PyTerrier) baseline [12]	0.190
Tangent-S baseline [12]	0.159
Linked MSE Posts baseline [12]	0.106

1. Yes, using the soft vector space model to capture the semantic similarity between tokens consistently improves effectiveness on ARQMath-3 Task 1, both for just text and for text combined with different math representations.

2. Among LaTeX and Tangent-L, our soft vector space models using Tangent-L achieve the highest effectiveness on ARQMath-3 Task 1.

3. Among lexical and semantic similarity, all joint models and the interpolated models for text reach their highest effectiveness on ARQMath-3 Task 1 by combining both lexical and semantic similarity, but place slightly more weight on lexical similarity. The interpolated models for math gave mixed results: The model for Tangent-L reaches the highest efficiency by using only semantic similarity, whereas the model for LaTeX reaches the highest efficiency by using only lexical similarity.

   Among sources of semantic similarity, joint models achieve comparable effectiveness on ARQMath-3 Task 1 with non-positional `word2vec`, positional `word2vec`, and MathBERTa, and interpolated models achieved comparable effectiveness with non-positional `word2vec` and positional `word2vec`. This may indicate that the soft vector space model does not fully exploit the semantic information provided by the sources of semantic similarity and therefore does not benefit from their improvements after a certain threshold.

4. All our interpolated models achieved higher effectiveness on ARQMath-3 Task 1 than our joint models. This shows that it is generally better to use two separate models to represent text and math even at the expense of losing the ability to model the similarity between text and math tokens.

Our answers to research questions 2 and 3 also provide the following new questions:

2. Are there other math representations besides LaTeX and Tangent-L that may work better with the soft vector space model?
3. How can the soft vector space model be parametrized or improved, so that it can benefit from improved measures of similarity between tokens?

These questions should provide a fruitful venue for future work.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the LINDAT-CLARIAH-CZ project LM2018101.

References

[1] Iz Beltagy, Kyle Lo, and Arman Cohan. “SciBERT. A Pretrained Language Model for Scientific Text”. In: *Proceedings of EMNLP-IJCNLP 2019*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3615–3620. DOI: 10.18653/v1/D19-1371. URL: https://aclanthology.org/D19-1371.

[2] Delphine Charlet and Geraldine Damnati. “Simbow at SemEval-2017 Task 3. Soft-Cosine Semantic Similarity Between Questions for Community Question Answering”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017, pp. 315–319. URL: https://aclanthology.org/S17-2051/ (visited on 10/16/2021).

[3] Deyan Ginev. *arXMLiv:2020 dataset, an HTML5 conversion of arXiv.org*. SIGMathLing – Special Interest Group on Math Linguistics. 2020. URL: https://sigmathling.kwarc.info/resources/arxmliv-dataset-2020/.

[4] Zheng Gong et al. “Continual Pre-training of Language Models for Math Problem Understanding with Syntax-Aware Memory Network”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 5923–5933. DOI: 10.18653/v1/2022.acl-long.408. URL: https://aclanthology.org/2022.acl-long.408.

[5] Hwiyeol Jo et al. “Modeling Mathematical Notation Semantics in Academic Papers”. In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 3102–3115. DOI: 10.18653/v1/2021.findings-emnlp.266. URL: https://aclanthology.org/2021.findings-emnlp.266.

[6] Chris Kamphuis et al. “Which BM25 do you mean? A large-scale reproducibility study of scoring variants”. In: *European Conference on Information Retrieval*. Springer. 2020, pp. 28–34.

[7] Andrew Kane, Yin Ki Ng, and Frank Tompa. “Dowsing for Answers to Math Questions. Doing Better with Less”. In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.

[8] Jinhyuk Lee et al. “BioBERT. A pre-trained biomedical language representation model for biomedical text mining”. English. In: *Bioinformatics* 36.4 (Feb. 2020), pp. 1234–1240. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682.

[9] Yinhan Liu et al. *RoBERTa. A Robustly Optimized BERT Pretraining Approach*. 2019. URL: https://arxiv.org/abs/1907.11692v1 (visited on 05/27/2022).

[10] Behrooz Mansouri, Douglas Oard, and Richard Zanibbi. “DPRL Systems in the CLEF 2022 ARQMath Lab. Introducing MathAMR for Math-Aware Search”. In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.

[11] Behrooz Mansouri et al. “Overview of ARQMath-2. Second CLEF Lab on Answer Retrieval for Questions on Math”. Working Notes Version. In: *Proceedings of the Working Notes of CLEF 2021* (Sept. 22–23, 2021). Ed. by Guglielmo Faggioli et al. Vol. 2936. Bucharest, Romania: CEUR-WS, 2021, pp. 1–24. URL: http://ceur-ws.org/Vol-2936/paper-01.pdf (visited on 10/23/2021).

[12] Behrooz Mansouri et al. “Overview of ARQMath-3 (2022). Third CLEF Lab on Answer Retrieval for Questions on Math”. Working Notes Version. In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.

[13] Tomáš Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013, pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[14] Yin Ki Ng et al. “Dowsing for Answers to Math Questions. Ongoing Viability of Traditional MathIR”. In: *Proceedings of the Working Notes of CLEF 2021* (Sept. 22–23, 2021). Ed. by Guglielmo Faggioli et al. Vol. 2936. Bucharest, Romania: CEUR-WS, 2021, pp. 63–81. URL: http://ceur-ws.org/Vol-2936/paper-05.pdf (visited on 10/13/2021).

[15] Vít Novotný. “Implementation Notes for the Soft Cosine Measure”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 1639–1642. DOI: 10.1145/3269206.3269317.

[16] Vít Novotný et al. "Ensembling Ten Math Information Retrieval Systems. MIRMU and MSM at ARQMath 2021". In: *Proceedings of the Working Notes of CLEF 2021* (Sept. 22–23, 2021). Ed. by Guglielmo Faggioli et al. Vol. 2936. Bucharest, Romania: CEUR-WS, 2021, pp. 82–106. URL: http://ceur-ws.org/Vol-2936/paper-06.pdf (visited on 10/13/2021).

[17] Vít Novotný et al. *Text classification with word embedding regularization and soft similarity measure.* 2020. URL: https://arxiv.org/abs/2003.05019v1 (visited on 10/15/2021).

[18] Vít Novotný et al. "Three is Better than One. Ensembling Math Information Retrieval Systems". In: *Proceedings of the Working Notes of CLEF 2020* (Sept. 22–25, 2020). Ed. by Linda Cappellato et al. Vol. 2696. Thessaloniki, Greece: CEUR-WS, 2020, pp. 93–122. URL: http://ceur-ws.org/Vol-2696/paper_235.pdf (visited on 10/23/2021).

[19] Vít Novotný et al. "When FastText Pays Attention. Efficient Estimation of Word Representations using Constrained Positional Weighting". In: *Journal of Universal Computer Science (J.UCS)* 28 (2 Feb. 28, 2022), pp. 181–201. ISSN: 0948-6968. DOI: 10.3897/jucs.69619.

[20] Shuai Peng et al. "MathBERT. A Pre-Trained Model for Mathematical Formula Understanding". In: *ArXiv* abs/2105.00377 (2021).

[21] Laila Rasmy et al. "Med-BERT. Pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction". In: *NPJ digital medicine* 4.1 (2021), pp. 1–13.

[22] Anja Reusch, Maik Thiele, and Wolfgang Lehner. "Transformer-Encoder and Decoder Models for Questions on Math". In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.

[23] Tetsuya Sakai and Noriko Kando. "On information retrieval metrics designed for evaluation with incomplete relevance assessments". In: *Information Retrieval* 11.5 (2008), pp. 447–470.

[24] Jia Tracy Shen et al. "MathBERT. A Pre-trained Language Model for General NLP Tasks in Mathematics Education". In: *ArXiv* abs/2106.07340 (2021).

[25] Grigori Sidorov et al. "Soft similarity and soft cosine measure. Similarity of features in vector space model". In: *Computación y Sistemas* 18.3 (2014), pp. 491–504.

[26] Michal Štefánik, Vít Novotný, and Petr Sojka. "RegEMT. Regressive Ensemble for Machine Translation Quality Evaluation". In: The 2021 Conference on Empirical Methods in Natural Language Processing EMNLP 2021 (Nov. 10, 2021). Nov. 10, 2021, pp. 1041–1048. URL: https://aclanthology.org/2021.wmt-1.112 (visited on 05/27/2022).

[27] Martin Geletka abd Vojtěch Kalivoda et al. "Diverse Semantics Representation is King. MIRMU and MSM at ARQMath 2022". In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.

[28] Richard Zanibbi et al. "Overview of ARQMath 2020. CLEF Lab on Answer Retrieval for Questions on Math". Updated Working Notes Version. In: *Proceedings of the Working Notes of CLEF 2020* (Sept. 22–25, 2020). Ed. by Linda Cappellato et al. Vol. 2696. Thessaloniki, Greece: CEUR-WS, 2020, pp. 1–27. URL: http://ceur-ws.org/Vol-2696/paper_271.pdf (visited on 10/23/2021).

[29] Wei Zhong, Yuqing Xie, and Jimmy Lin. "Applying Structural and Dense Semantic Matching for the ARQMath Lab 2021, CLEF". In: *Proceedings of the Working Notes of CLEF 2022* (Sept. 5–8, 2022). Ed. by Guglielmo Faggioli et al. Bologna, Italy: CEUR-WS, 2022.