Towards Disambiguation of Mathematical Terms based on Semantic Representations*

Shufan Jiang[1,2], Mary Ann Tan[1,2] and Harald Sack[1,2]

[1]*FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Eggenstein-Leopoldshafen, Germany*
[2]*Karlsruhe Institute of Technology, Karlsruhe, Germany*

Abstract

In mathematical literature, terms can have multiple meanings based on context. Manual disambiguation across scholarly articles demands massive efforts from mathematicians. This paper addresses the challenge of automatically determining whether two definitions of a mathematical term are semantically different. Specifically, the difficulties and how contextualized textual representation can help resolve the problem, are investigated. A new dataset MathD2 for mathematical term disambiguation is constructed with ProofWiki's disambiguation pages. Then two approaches based on the contextualized textual representation are studied: (1) supervised classification based on the embedding of concatenated definition and title and (2) zero-shot prediction based on semantic textual similarity(STS) between definition and title. Both approaches achieve accuracy and macro F1 scores greater than 0.9 on the ground truth dataset, demonstrating the effectiveness of our methods for the automatic disambiguation of mathematical definitions. Our dataset, code, and experimental results are available here: https://github.com/sufianj/MathTermDisambiguation.

Keywords

Entity Linking, Text Similarity, Transformers, Mathematical Definition

1. Introduction

Mathematical scholarly articles contain highly structured statements, such as axioms, theorems, and proofs, which are not easily navigable or explorable through traditional keyword searches. Several initiatives have emerged to enhance the discovery of mathematical definitions. Argot [1][1] is a collection of term-definition pairs automatically extracted from mathematical papers, allowing users to retrieve all definitions of a given term. MathMex [2] [2] is a recent search engine for mathematical definitions based on the semantic similarity between a user's query and the definition. Both projects show promising usage of different word embeddings. However, Argot cannot disambiguate polysemous terms, while MathMex cannot guarantee that the retrieved definitions accurately define the queried term. Both highlight the need for an automatically constructed knowledge base of mathematical definitions from scholarly articles. Such a knowledge base would enable researchers to efficiently look up terms and index relevant mathematical statements and articles.

Existing research in this area focuses on extracting mathematical definitions [3, 4, 5, 6] and identifying the terms defined therein, known as *definienda* (singular: *definiendum*) [1, 7]. These tasks are extended by disambiguating or linking newly extracted definition-term pairs to existing concepts in a reference glossary, or otherwise expanding the glossary. Current work about math term disambiguation shows promising applications of natural language processing for resolving token-level ambiguity in equations, such as the ambiguity of "prime" ($'$) [8], and for linking formulae and identifiers(formula variable without fixed value) in STEM papers to Wikidata [9].

Disambiguating definienda is particularly challenging when identical terms for the same concept are defined in various ways (e.g., "path") or when polysemous terms (e.g., "block") refer to distinct concepts

SCOLIA '25: First International Workshop on Scholarly Information Access (SCOLIA), April 10, 2025, Lucca, Italy

*An extended version of this paper is under review.

shufan.jiang@fiz-karlsruhe.de (S. Jiang); ann.tan@fiz-karlasruhe.de (M. A. Tan); harald.sack@fiz-karlasruhe.de (H. Sack)
0000-0002-8486-3158 (S. Jiang); 0000-0003-3634-3550 (M. A. Tan); 0000-0001-7069-9804 (H. Sack)

[1] https://efedequis.xyz/argot/

[2] https://www.mathmex.com/

Table 1
Definitions extracted from different scholarly articles [7]. The definition of "path" has different formulations. The notion of "block" has different meanings.

Definiendum	Definition and Source Article
path	If the vertices $v_0, v_1, \ldots, v_k$ of a walk W are distinct then W is called a *Path*. A path with n vertices will be denoted by P_n. P_n has length $n-1$. [10]
path	Let $G = (V, E)$ be a graph. A *path* in a graph is a sequence of vertices such that from each of its vertices there is an edge to the next vertex in the sequence. This is denoted by $P = (u = v_0, v_1 \ldots, v_k = v)$, where $(v_i, v_{i+1}) \in E$ for $0 \leq i \leq k-1$. [11]
block	A *block* in H is a maximal set of tightly-connected hyperedges. [12]
block	A *block* of indices is a set of numbers S where every term $SG_{a,b}(s)$ depends on the same value via division, for all $s \in S$. [13]

(see Table 1). A possible heuristic is that if the definienda of two definitions are linked to different concepts in a reference knowledge base, then these two definitions are distinct. This scenario assumes that each definition corresponds to one definiendum.

For this study, ProofWiki[3] serves as the reference list. It is a crowd-sourced online collection of mathematical proofs, including 500 disambiguation pages. Similar to Wikipedia, these disambiguation pages list identical terms, each linking to its corresponding definition page. Each definition page includes a unique page title, the definition, and a topic or category where the term can be found (e.g., algebra or geometry). Specifically, the page title contains the definiendum along with its category and serves as the identifier of the definition page within ProofWiki (e.g. "Definition:Bilinear Form (Polynomial Theory)" in table 2).

This work addresses the following research questions: ***RQ1:*** *How well can contextualized word embeddings help the disambiguation of mathematical terms?* ***RQ2:*** *Which pretraining strategies and downstream tasks best suit this task?* The main contributions of this work are:

- **MathD2** - a new dataset for **Math**ematical **D**efiniendum **D**isambiguation.
- Exploration of **two different approaches** demonstrating how the disambiguation task can benefit from contextualized semantic representations.
- **Experiment-supported evidence** highlighting the efficiency of sentence embeddings for the addressed disambiguation task.

2. Related Work

The challenges posed by this task are (a) the lack of labeled datasets for equivalent mathematical definitions, (b) the limited number of disambiguation pages, and (c) the unstructured nature of definitions that combine mathematical notations, formulas, and general discourse [7, 6]. To address (a), entity linking and sentence similarity approaches for mathematical terms are reviewed. To tackle (b) and (c), transformer models [14] are employed for their capabilities to produce rich, contextualized representations.

Contextualized representations produced by BERT (Bidirectional Encoder Representations from Transformers) [15] encode the meaning of a word according to its context. This means that polysemous words have several, more accurate representations depending on the sentence where they appear. BERT is pretrained on two key tasks: Masked Language Modeling (MLM), where random tokens in a sentence are masked and predicted based on context, and Next Sentence Prediction (NSP), which trains BERT to determine whether a sentence logically follows another. Pretraining with MLM is widely applied for domain adaptation , especially when there is a dearth of data for finetuneing [16, 17]. In addition, finetuning BERT for specific downstream tasks and domains is straightforward. For instance, by combining BERT's output with a classification layer, it has been adapted for mathematical notation

[3] https://ProofWiki.org/wiki/Main_Page

Table 2
Data extracted from a ProofWiki disambiguation page.

Definition	Title
Let R be a ring. Let R_R denote the R-module R. Let M_R be an R-module. A bilinear form on M_R is a bilinear mapping $B : M_R \times M_R \to R_R$.	Definition:Bilinear Form (Linear Algebra)
A bilinear form is a linear form of order 2.	Definition:Bilinear Form (Polynomial Theory)

prediction [18], definiendum extraction [7] and mathematical statement extraction [19]. The Natural Language Inferernce (NLI) datasets [20, 21] used by BERT's NSP pretraining are related to the task at hand. A piece of supporting evidence is AcroBERT [22], an entity linker that reuses BERT for NSP's pretrained weights and is finetuned to link acronyms to their long forms. AcroBERT outperforms BERT and other domain-adapted BERT-based models.

However, the nature of the BERT's pretraining tasks makes it unsuitable for measuring semantic similarity. Sentence BERT (SBERT) [4] [23] modifies BERT's architecture to produce semantically meaningful sentence embeddings that can be compared using cosine-similarity. Out-of-the-box SBERT achieves superior performance across varied classification tasks involving mathematical texts [24]. In one such task, the proponents measure the similarity of SBERT embeddings between an input text and the combination of titles and abstracts of mathematical publications in arXiv [5] and zbMATH [6] to predict the classification code of the respective repositories. In the same vein, this study aims to evaluate the effectiveness of semantic textual similarity in linking definitions to titles. Since BERT for NSP and SBERT require different domain adaptation strategies [23, 24], this work first identifies the architecture that performs better for the task.

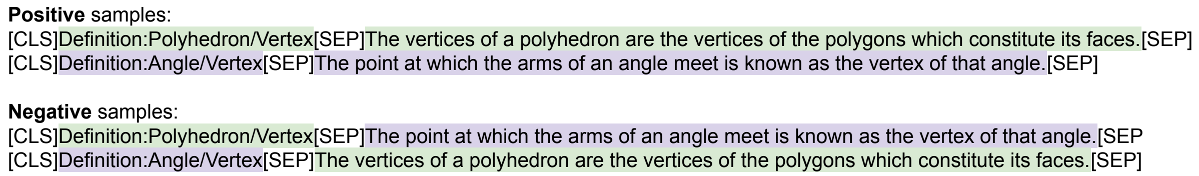
3. Methodology

Term disambiguation is formalized as an entity linking task, where the entities refer to the definition page titles in ProofWiki. That is, given (1) a definition and an ambiguous definiendum and (2) a dictionary that maps the ambiguous definiendum to entities, the goal is to find the title that best matches the definition. The proposed method is described in two steps. First, the ground truth dataset is constructed. Second, two applicable approaches are considered.

3.1. Construction of the MathD2 Dataset

A dump of the whole ProofWiki was extracted on the 9th July, 2024 using WikiTeam [25]. This dump is then parsed to get all the definition statements and titles from all definition disambiguation pages. The extracted definitions is converted to plain text. By mapping ambiguous terms and the corresponding definition titles is finally constructed. Some definitions might contain other definitions(e.g., the definition of "Loop" [7]), which also happens to definitions in scholarly papers. If both definition titles are mapped to a common ambiguous term, only the nested definition and its title are kept, because otherwise the outer definition should be mapped to two titles: its title and the one of the nested definition. Finally, terms mapped to less than two titles are removed. Table 2 shows (definition, title) pairs extracted from the disambiguation page of "Bilinear Form' [8]. For the finetuning in Section 3.2, the dataset is split based on the 343 ambiguous terms at the ratio of 8:2, making a training dataset of 275 ambiguous terms with 1436 (definition, title) pairs and a test dataset of 68 ambiguous terms with 433 (definition, title) pairs. All (definition, title) pairs from one disambiguation page are kept together in either the training or test sets,

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2
[5] https://arxiv.org/
[6] https://zbmath.org/
[7] https://proofwiki.org/wiki/Definition:Loop_(Topology)
[8] https://proofwiki.org/wiki/Definition:Bilinear_Form



Positive samples:
[CLS]Definition:Polyhedron/Vertex[SEP]The vertices of a polyhedron are the vertices of the polygons which constitute its faces.[SEP]
[CLS]Definition:Angle/Vertex[SEP]The point at which the arms of an angle meet is known as the vertex of that angle.[SEP]

Negative samples:
[CLS]Definition:Polyhedron/Vertex[SEP]The point at which the arms of an angle meet is known as the vertex of that angle.[SEP
[CLS]Definition:Angle/Vertex[SEP]The vertices of a polyhedron are the vertices of the polygons which constitute its faces.[SEP]

Figure 1: Sample training data made from the ambiguous term "Vertex". Positive samples are good matches of title and definition.

so that the generalizability of the finetuned model on unseen terms can be evaluated. In the finetuning of Section 3.2, for each ambiguous term, two definitions and their titles are randomly selected to make positive pairs, and the titles of two other definitions to make negative pairs (see example in Figure 1). Both approaches are evaluated on the training and test datasets, except for the finetuned model.

3.2. Classification Based on One Concatenated Embedding

Following the finetuning setup of AcroBERT [22], BERT for NSP is adapted to build a supervised sentence pair classifier to link definitions to their page titles in ProofWiki. Every pair of (definition, candidate title with the matching ambiguous term in ProofWiki) is concatenated as an input sequence. The sequence begins with a [CLS] token, followed by a candidate title, a [SEP] token, and then the definition, ending with [SEP]. The input sequence passes through BERT's transformer layers. These layers produce contextual embedding for each token in the sequence. Then, the embedding of [CLS] is fed into a softmax classification layer, which outputs a score to judge how coherent the concatenated sequence is. The pair with the highest score is selected as the final predicted output. First the out-of-box BERT for NSP serves as the baseline to see how well the pre-retained natural language inference model can describe the entailment between the titles and definitions. Then the pretrained BERT for NSP is finetuned with the training set using a triplet loss function:

$$\mathcal{L} = \max\left\{0, \lambda - d_{\text{neg}} + d_{\text{pos}}\right\} \tag{1}$$

that aims to assign higher scores to the correct titles that match the input definition while reducing the scores of irrelevant candidates, where $\lambda = 0.2$ is the margin value, and d_{pos} and d_{neg} are the distances for positive and negative pairs, respectively. This approach is implemented with PyTorch [26] and transfomers [27]. A batch size of 16 and Adam optimizer with learning rate 1e-5 are used. The learning rate is exponentially decayed at a rate of 0.95 every 1000 steps. The model is trained with the training dataset for 100 epochs. After each epoch, a checkpoint (copy of the current model weights) is saved. Each checkpoint is then evaluated with the test dataset so that test data do not impact the model weights.

3.3. Zero-shot Prediction Based on Semantic Textual Similarity between Two Embeddings

A shortcoming of the previous solution is that the NSP inference has to be run for every (definition, title) pair mapped to an ambiguous term. Motivated to make a computationally more efficient solution, the sentence embeddings of the definitions and titles are explored. In this setup, the sentence embedding of the titles and the definitions only need to be calculated once. For the definition and each candidate title with the matching ambiguous term, the title with the highest cosine similarity to the embedding of the definition is selected as the final predicted output. To explore the potential benefits of different pretraining corpus and related tasks, the following models are studied:

- The best-performing sentence transformers for Semantic Textual Similarity(STS) tasks for short mathematical text as reported in [24], including out-of-box SBERT [23], math-similarity/Bert-

MLM_arXiv-MP-class_arXiv [24] (noted as Adapted SBERT in Table 3), and mini SBERT models SBERT/all-MiniLM-L6-v2 [28], and SBERT/all-MiniLM-L12-v2 [28].
- Mean pooled out-of-box BERT, to compare with the pretraining of SBERT.
- Mean pooled out-of-box CC-BERT [16], a from-scratch model pretrained with MLM on mathematical papers. This experiment studies the impact of domain-specific MLM pretraining and domain-specific tokenization, comparing to mean pooled out-of-box BERT

Following SBERT's default setting [23], the mean pooling strategy is used to calculate the sentence embeddings with out-of-box BERT and CC-BERT.

4. Results and Discussion

Accuracy and the average of the F_1 score for each ambiguous term (macro F_1 score) are used to measure how well both approaches can link a definition to the correct title. Table 3 shows the experimental results of both methods. Overall, our finetuned NSP model performs best, validating AcroBERT's set-up and the helpfulness of BERT for NSP's pretrained weights. Notably, the out-of-the-box SBERT demonstrated excellent performance with much less inference time. The performance of prediction based on STS with sentence transformers is aligned with the results of [23] and [24]. Given that both BERT for NSP and SBERT are pretrained on NLI tasks [15, 23], it may be deduced that i) compared to using the [CLS] representation of concatenated sequence, using separated sentence embeddings captures more information for our task, and/or ii) SBERT's pretraining on (title, abstract) pairs from S2ORC dataset [29] helps to better understand the entailment between titles and body texts. However, the domain-adapted SBERT model [9] that the authors of [24] finetuned with multiple tasks using titles and abstracts of mathematical papers does not yield better results than general SBERT models. This might be due to the model being solely trained on titles and abstracts, diminishing the model's representational capacity for both formulas and general text. The experiments with the mean pooled out-of-box BERT and CC-BERT show that MLM domain-adaptation over mathematical papers slightly improves this task but is far less efficient than adapted SBERT, which has been pretrained with fewer data but on a better task.

Limitations: An interesting finding is that SBERT for STS and the finetuned BERT for NSP make some common mistakes, indicating the limits of using only semantic representations. The most common error is when the definition statement includes nested definitions. Another typical error is that the predicted result is in the correct category but not the definiendum, mainly when the definition contains morphemes in the predicted title or when the definition does not contain some morphemes in the expected title. For example, the definition of "Consequence Function" starts with "Let $\mathbf{G}$ be a game..." [10], and the predicted title is "Definition:Consequence(Game Theory)' [11]. Thus, enhancing sentence embedding's comprehension of semantic and syntactic knowledge of mathematical definitions is still worth investigating. Other common mistakes reveal the noises in the dataset due to automatic scrapping and LaTeXconversion of irregular ProofWiki pages.

5. Conclusion and Future Works

This work introduces a new dataset for mathematical term disambiguation with ProofWiki. Two entity linking approaches have been implemented and shown to yield advantages in the usage of contextualized embeddings to differentiate mathematical definitions. The experimental results proved the efficiency and effectiveness of using out-of-the-box SBERT. Further work is planned on applying the proposed approaches on scholarly papers. In addition, the current approach is to be extended to include document-level representation and citation information to differentiate definitions in scholarly papers. This work also indicates the need for further study on building sentence transformers that benefit from domain-specific MLM and task-related pretraining.

[9] https://huggingface.co/math-similarity/Bert-MLM_arXiv-MP-class_arXiv

[10] https://proofwiki.org/wiki/Definition:Consequence_Function

[11] https://proofwiki.org/wiki/Definition:Consequence_(Game_Theory)

Table 3

Accuracy and macro F_1 scores. Models convert the input sentence to embeddings, and the embeddings are fed to different calculations according to the approach. Values are reported as $\rho \cdot 100$. Only the finetuning of BERT with NSP has been trained with the Train Set, the rest approaches are unsupervised. Thus, the unsupervised approaches are also evaluated over the Train Set to have more results.

Model	Approach	Test		Train	
		F_1	Acc.	F_1	Acc.
BERT [15]	NSP	80.9	84.8	79.8	83.9
finetuned BERT	NSP	**92.1**	**93.8**	-	-
Mean Pooled BERT	STS	26.1	35.3	30.2	40.3
Mean Pooled CC-BERT [16]	STS	28.2	37.9	35.6	45.4
SBERT/all-MiniLM-L6-v2 [28]	STS	90.1	92.4	88.1	91.0
SBERT/all-MiniLM-L12-v2 [28]	STS	91.2	93.3	**89.4**	**92.1**
SBERT-all-mpnet-base-v2 [23]	STS	**91.4**	**93.5**	89.0	91.6
Adapted SBERT [24]	STS	43.8	52.2	54.0	61.5

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: grammar and spelling check, paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] L. Berlioz, ArGoT: A Glossary of Terms extracted from the arXiv, Electronic Proceedings in Theoretical Computer Science 342 (2021) 14–21. URL: http://arxiv.org/abs/2109.02801v1. doi:10.4204/EPTCS.342.2.

[2] S. Durgin, J. Gore, B. Mansouri, Mathmex: Search engine for math definitions, in: European Conference on Information Retrieval, Springer, 2024, pp. 194–199.

[3] L. Berlioz, Hierarchical Representations from Large Mathematical Corpora, Ph.D. thesis, University of Pittsburgh, 2023.

[4] K. Nakagawa, A. Nomura, M. Suzuki, Extraction of Logical Structure from Articles in Mathematics, in: A. Asperti, G. Bancerek, A. Trybulec (Eds.), Mathematical Knowledge Management, volume 3119, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 276–289. URL: http://link.springer.com/10.1007/978-3-540-27818-4_20. doi:10.1007/978-3-540-27818-4_20, series Title: Lecture Notes in Computer Science.

[5] Y. Sun, H. Zhuge, Discovering patterns of definitions and methods from scientific documents, arXiv preprint arXiv:2307.01216 (2023).

[6] N. Vanetik, M. Litvak, S. Shevchuk, L. Reznik, Automated discovery of mathematical definitions in text, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 2086–2094.

[7] S. Jiang, P. Senellart, Extracting definienda in mathematical scholarly articles with transformers, in: Proceedings of the Second Workshop on Information Extraction from Scientific Publications, 2023, pp. 31–36.

[8] R. Shan, A. Youssef, Towards Math Terms Disambiguation Using Machine Learning, in: F. Kamareddine, C. Sacerdoti Coen (Eds.), Intelligent Computer Mathematics, Springer International Publishing, Cham, 2021, pp. 90–106.

[9] P. Scharpf, M. Schubotz, B. Gipp, Towards explaining stem document classification using mathematical entity linking, arXiv preprint arXiv:2109.00954 (2021).

[10] S. J. Kalayathankal, et al., Operations on covering numbers of certain graph classes, arXiv preprint arXiv:1506.03251 (2015).

[11] K. Perera, Y. Mizoguchi, Bipartition of graphs based on the normalized cut and spectral methods, arXiv preprint arXiv:1210.7253 (2012).

[12] B. Ergemlidze, E. Győri, A. Methuku, 3-uniform hypergraphs without a cycle of length five, arXiv preprint arXiv:1902.06257 (2019).

[13] E. Kupin, Subtraction division games, arXiv preprint arXiv:1201.0171 (2011).

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023. URL: https://arxiv.org/abs/1706.03762. arXiv:1706.03762.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[16] S. Mishra, L. Pluvinage, P. Senellart, Towards extraction of theorems and proofs in scholarly articles, in: P. Healy, M. Bilauca, A. Bonnici (Eds.), DocEng '21: ACM Symposium on Document Engineering 2021, Limerick, Ireland, August 24-27, 2021, ACM, 2021, pp. 25:1–25:4. URL: https://doi.org/10.1145/3469096.3475059. doi:10.1145/3469096.3475059.

[17] S. Jiang, R. Angarita, S. Cormier, J. Orensanz, F. Rousseaux, Choubert: Pre-training french language model for crowdsensing with tweets in phytosanitary context, in: International Conference on Research Challenges in Information Science, Springer, 2022, pp. 653–661.

[18] H. Jo, D. Kang, A. Head, M. A. Hearst, Modeling mathematical notation semantics in academic papers, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 3102–3115.

[19] S. Mishra, Y. Brihmouche, T. Delemazure, A. Gauquier, P. Senellart, First steps in building a knowledge base of mathematical results, in: Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024), 2024, pp. 165–174.

[20] S. R. Bowman, G. Angeli, C. Potts, C. D. Manning, A large annotated corpus for learning natural language inference, in: L. Màrquez, C. Callison-Burch, J. Su (Eds.), Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 632–642. URL: https://aclanthology.org/D15-1075. doi:10.18653/v1/D15-1075.

[21] A. Williams, N. Nangia, S. Bowman, A broad-coverage challenge corpus for sentence understanding through inference, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1112–1122. URL: https://aclanthology.org/N18-1101. doi:10.18653/v1/N18-1101.

[22] L. Chen, G. Varoquaux, F. M. Suchanek, GLADIS: A general and large acronym disambiguation benchmark, in: Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 2073–2088. URL: https://aclanthology.org/2023.eacl-main.152/. doi:10.18653/v1/2023.eacl-main.152.

[23] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://aclanthology.org/D19-1410. doi:10.18653/v1/D19-1410.

[24] C. Steinfeldt, H. Mihaljević, Evaluation and domain adaptation of similarity models for short mathematical texts, in: International Conference on Intelligent Computer Mathematics, Springer, 2024, pp. 241–260.

[25] WikiTeam, Wikiteam, 2024. URL: https://github.com/WikiTeam/wikiteam, original-date: 2014-06-25T10:18:03Z.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).

[27] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[28] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, Advances in Neural Information Processing Systems 33 (2020) 5776–5788.

[29] K. Lo, L. L. Wang, M. Neumann, R. Kinney, D. Weld, S2ORC: The semantic scholar open research corpus, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4969–4983. URL: https://www.aclweb.org/anthology/2020.acl-main.447. doi:10.18653/v1/2020.acl-main.447.