

# BERT-Based Embedding Model for Formula Retrieval

Pankaj Dadure, Partha Pakray and Sivaji Bandyopadhyay

Department of Computer Science and Engineering  
National Institute of Technology Silchar  
Assam, India 788010

## Abstract

The web is a rich repository of mathematical information, the task of finding relevant information in such collection is a laborious one. However, numerous efforts have been made to develop web-accessible mathematical and scientific search systems in the recent past. In this paper, we have presented the BERT-based formula embedding model to facilitate formula retrieval in ARQMath2 tasks. To depict the performance of the pre-trained model for mathematical language processing tasks, the proposed BERT-based model is trained on the math exchange corpus of the ARQMath. It takes the  $\LaTeX$  formulas as input and produced the fixed dimensional embeddings for the same. For similarity measure, the cosine similarity has been used. The obtained results have shown that the proposed approach provides better fits than existing embedding approaches and infers the meaningful semantic relationships between equations.

## Keywords

Formula Retrieval, Mathematical Information Retrieval, BERT, Math Stack Exchange, Formula Embedding

## 1. Introduction

The web is a rich repository of mathematical information, including Wikipedia, Arxiv, a growing number of digital libraries, and research publications in science & engineering. In such a collection, the formula is the most common form of mathematical information. Meanwhile, formulae are highly structured and represented in predefined layout structures such as  $\LaTeX$  and MathML. The ongoing development of such information encouraged the implementation of advanced tools and techniques to handle and analyze mathematical formulas effectively. The automatic retrieval of mathematical formulas is significantly beneficial for understanding scientific documents. As a result, the conventional information retrieval (IR) system treated the formulae as a text, however, it is unable to capture the structural and semantic meaning of the formulae. To fill this gap, the mathematical information retrieval (MIR) system comes under the limelight and has drawn increasing attention from researchers. The prime task of a mathematical information retrieval system is to retrieve the scientific document/formulae relevant to the queried formula [1]. In MIR, the sense of the term 'relevant' is a subjective matter [2] and is defined in two ways: the first one finds the relevance of queried formula based on their structural similarities, and the second one finds the relevance of queried formula based semantic

---

*CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania*  
✉ krdadure@gmail.com (P. Dadure); parthapakray@gmail.com (P. Pakray); sivaji.ju.cse@gmail.com (S. Bandyopadhyay)



© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

similarities (formulae hold the same meaning but holds the different representation). In addition to these, it's considered not only the exact matching of queried formula but also those formulae that are partially matched with the queried formula (sub-formula and parent-formula). In the retrieval of a mathematical formula, the effective representation of the formulas is the prime aspect, leading to the preserve the connections between the mathematical symbols and creating the navigator for exploring the scientific documents.

For many applications like information retrieval, document clustering, or classification, the process of capturing semantics is a probabilistic way. The first step is common to incorporate words or documents into a vector space. Neural network approaches like word embedding models [3] acts as state-of-the-art models for representing individual words with semantically fixed-length vectors and provides excellent results compared to the tf-idf or count vectorizer. Word embedding makes it possible to successfully apply deep learning to natural language processing applications such as machine translation, text summarization, and question answering. Recently, several neural network-based retrieval approaches have been introduced, and some of them have made significant improvements in their performance [4]. However, in mathematical language tasks, there have been only a few similar problems.

This paper has implemented the formula embedding approach, which encodes the formula into the embedded vector. For encoding the formula, we have used pre-trained bidirectional encoder representations from transformers model. The proposed embedding model takes the  $\text{\LaTeX}$  formula as input and produces an output as a fixed dimensional embedding representation. Furthermore, the embeddings of the formulae & the queried formula are compared, and cosine similarity is estimated. The performance of the proposed approach has been tested using a math stack exchange corpus of ARQMath 2020, and obtained results have shown a remarkable contribution in the task of formula retrieval.

The paper is structured as follows: Section 2 describes the prior work related to MIR domain. Section 3 gives a detailed account of the dataset. Section 4 provides detailed description about the system architecture. Section 5 & 6 describes the experimental setup and results. Section 7 concludes with summary and directions of further research and developments.

## 2. Related Work

Mathematical information retrieval is described as the low-hanging fruit of mathematical knowledge management, and people who come from the retrieval community have addressed it in several papers. In mathematical information retrieval, formulae and text both are important factors to achieve the state-of-the-art results and satisfy the user's need based on the formula-based query or text-based query or both (formula+text). For example, the variable typing approach [5] has assigned the meaning to variables based on the text contained in the same sentence. Types are multi-word phrases which normally used to indicate mathematical terminologies such as objects (e.g., "set"), algebraic structures (e.g., "monoid"), and instantiable notions (e.g., "cardinality of a set"). To evaluate the contribution of variable typing approach, two baseline system i.e. nearest type & the SVM proposed by [6][7] and three newly proposed approaches i.e. extended version of SVM baseline, convolutional neural network, and bidirectional LSTM [8][9] have been employed. Among these approaches, the bidirectional LSTM achieved remarkable

results. The signature-based hash indexing approach [10] appears to be a more appropriate alternative to text-based models. In this approach, mathematical formulae have been extracted from scientific documents and transformed to structure encoded strings (SES). These strings served as the input for the hash-based indexing scheme, which aimed to convert these SES into a bit vector/signatures. Finally, these bit vectors have been compared with the user-entered query, and relevant results have been retrieved. In the formula-based engine, the vector-based approach shown remarkable performance. For instance, a Binary Vector Transformation of Math Formula (BVTMF) [11] has attained Presentation MathML formulae from documents and constructs the fairly large-sized binary vectors where '0's represents absence and '1's represents the presence of a particular entity in the formula. The generated formula vector has representative of the information content of the corresponding formula. Moreover, for indexing and searching textual contents, the system relies on Apache Lucene<sup>1</sup>. Text and math search results retrieved have been re-ranked to prioritize the results containing text and math components of the user query. Motivated from promising performances of the LSTM for sequence-to-sequence tasks, an LSTM based Formula Entailment (LFE) approach [12] has successfully identified the entailment between the formula-based user query and formulae contained in the scientific documents. The LFE approach has been trained and validated using a symbol level Math Formula Entailment (MENTAIL) dataset.

At ARQMath-2020 [13] formula search task, our earlier system titled "variable size formula embedding approach" [14] transformed the formula (Presentation MathML Format) into the variable size vector where each bit of vector represents their occurrence and corresponds to their position in Bit Position Information Table (BPIT). At ARQMath-2020 [13], the DPRL has one of the well-performed research teams from Pattern Recognition Lab of Rochester Institute of Technology, which introduced the Tangent-CFT system [15]. The Tangent-CFT system has used both the Symbol Layout Tree (SLT) and Operator Tree (OT) representations of formula to consider both the appearance and the syntax of formulas. Tangent+CFT has the extension of the Tangent-CFT embedding model in which each formula has two vector representations: *Formula Vector*: Vector representation obtained by Tangent-CFT system where vector size is 300. *Text Vector*: Vector representation obtained by considering the formula as a word and trained the fastText model on the surrounding words of the formula. The vector size of 100 is the fastText default value. Moreover, team MIRMU [16] has designed the two systems named Soft Cosine Measure (SCM) and Formula2Vec. The SCM system combines TF-IDF with unsupervised word embeddings to produce interpretable representations of math documents and math formulae that enable fast information retrieval. In Formula2Vec, documents and formulae are represented by document and formula embeddings produced by training the Doc2Vec DBOW model on text and math data.

The mathematical formulae are diverse in nature in terms of syntax and semantic. To integrate this feature in a mathematical information retrieval system, HFS (Hesitation Fuzzy Sets) and BERT (Bidirectional Encoder Representations from Transformer) have examined the mathematical formula and calculates the membership degree of symbolic multi-attributes. With the extraction of the text of the formula, BERT has been used to calculate the context similarity. Then, the documents have been ranked according to the similarity of context & formula, and

---

<sup>1</sup><http://lucene.apache.org/>

**Table 1**  
Math Stack Exchange Corpus Description

<b>Corpus</b>	Math Stack Exchange ARQMath-2020
<b>Type</b>	Formula
<b>Formats</b>	$\LaTeX$ , Presentation MathML, and Content MathML
<b>Size</b>	1.5 GB, 11.5 GB and 10.9 GB
<b>No. of Formulas</b>	28320920, 26075012 and 25366913
<b>No. of Test Queries</b>	45 (ARQMath) and 60 (ARQMath2)

the final retrieval result has been obtained.

### 3. Corpus Description

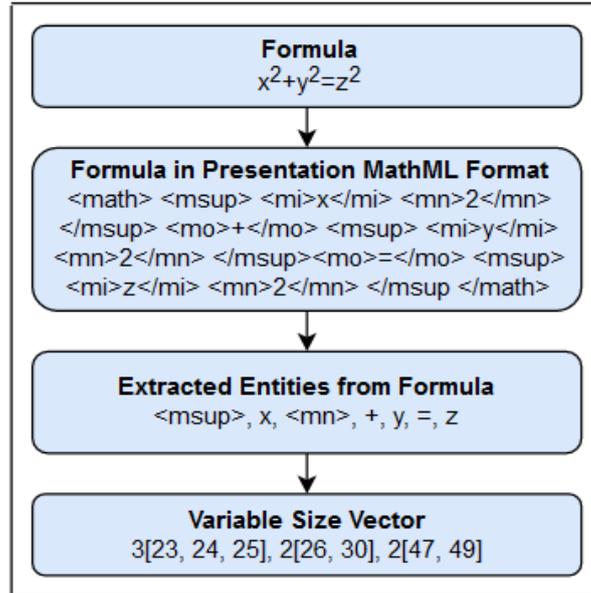
The ARQMath-2021 task [13] provided the formula-based corpus, which is collected from the knowledge-sharing platform, i.e., Math Stack Exchange (MSE). The provided dataset comprised the formulas extracted from the question, answer, and comment posts. In the dataset, the formulas are represented in three different formats, i.e.,  $\LaTeX$ , Presentation MathML, and Content MathML format. The number of formulas comprised in  $\LaTeX$ , Presentation MathML, and Content MathML formats is 28320920, 26075012, and 25366913 of size 1.5 GB, 11.5 GB, and 10.9 GB respectively. Each format has five distinct attributes, namely formula\_id, post\_id, thread\_id, type, and formula. The metadata about the formula dataset is shown in Table 1.

### 4. System Architecture

In ARQMath-2021, we have submitted a total of four runs, one for the topics provided in the ARQMath-2020 and the remaining three-run for the topics provided in the ARQMath-2021. The run submitted for the ARQMath-2020 topics is based on a newly designed BERT-based formula embedding approach. Moreover, the first run of the ARQMath2-2021 has been obtained from our earlier system [14] that we have designed in ARQMath-2020. In addition to this, the second and third runs of the ARQMath2-2021 have been obtained from the BERT-based formula embedding approach.

#### 4.1. Our Earlier System

We have designed the variable-size vector based approach in ARQMath-2020, which is motivated from the existing Bit Position Information Table (BPIT) [17] and Term-Document matrix [18]. The prime objective of this system is to transform the formula (Presentation MathML format) into the variable size vector. Each weight of the vector represents the occurrence count of a particular entity in a formula and corresponds to entity position in BPIT [17]. The process of formula to variable-size vector transformation is shown in Figure 1. In the vector transformation process, entities in a formula are categorized into three categories based on the tags. The entities tagged by  $\langle mi \rangle$  comes under the first category, entities tagged by  $\langle mo \rangle$  comes under the second category, and the third category holds the essential MathML tags, which contribute to



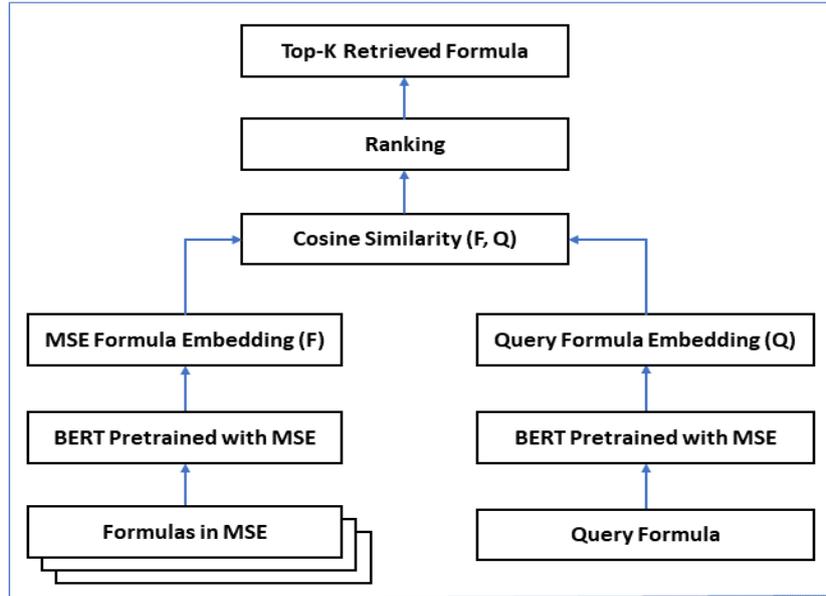
**Figure 1:** Formula to Vector Transformation

the semantics of the formula. As per the position of entities in BPIT, the `<mi>` tags hold the positions 0-25, 57-65 & 71-100, the `<mo>` tags hold the positions 26-45, 66-70 & 101-149 and the positions 46-56 holds the essential MathML tags. As defined in the Figure 1, the generated vector  $3[23, 24, 25], 2[26, 30], 2[47, 49]$  from the formula  $x^2 + y^2 = z^2$  where 3, 2 and 2 defined the occurrence count of `<mi>`, `<mo>` and essential MathML tags respectively and 23, 24, 25, 26, 30, 47, 49 represents the bit position of the `<mi>`, `<mo>` and essential MathML tags in BPIT.

After a successful formula transformation into the variable size vector, the indexer module indexed the formula vector into an index. Each index stored the three different fields, namely embedded formula vector, formula id, and post id from which the formula is originated. Hereafter, the searcher module compared the query formula vector with all the indexed formula vectors and computed the similarity for each indexed formula. For similarity calculation, the system compared each bit of query vector with the bits of formula vector, and those formulas contained the maximum number of similar bits that formula have maximum similarity score. Based on the maximum similarity score, the system retrieves the top-k formulas with respect to the user-entered query formula.

## 4.2. Proposed System in ARQMath2

Word embedding is one of the most common text vocabulary representations. It captures the meaning of words, semantic & syntactic correlation, and similarity within the words. Word embeddings describe the word in low dimensional vector form, and to obtain this, an appropriate composition function is required. The composition function is a mathematical framework that combines multiple words into a single vector. The prior research works have witnessed that



**Figure 2:** Workflow Diagram of the Proposed BERT-Based System

the embeddings of longer size input strings or sentences achieved excellent performance in the Semantic Textual Similarity (STS) [19]. Motivated from this, we have performed formula embedding, which encodes the formulas to embedding vectors. For encoding the formula, we have used bidirectional encoder representations from transformers (BERT) model. The workflow of the BERT-based formula embedding model is shown in Figure 2.

#### 4.2.1. Preprocessing

Formulas in  $\LaTeX$  like  $A^2 + B^2$  and  $a^2 + b^2$  have same syntactic and semantic meaning but when not converted to the lower case those two are represented as two different formulas in the vector space model<sup>2</sup>. To handle this, we have converted all formulas into the lower case.

#### 4.2.2. Formula Embedding

After preprocessing of the formulas, the preprocessed formulas are fed into the BERT model. The BERT model takes the preprocessed  $\LaTeX$  formula as input and produces an output as a fixed dimensional embedding vector. Furthermore, the embedding vectors of the formulae and embedding vector of the queried formula is compared and computes the cosine similarity [20][21]. Based on the highest similarity score, the top-K formulas have been retrieved.

**BERT: Bidirectional Encoder Representations from Transformers:** BERT [22] is a context-independent word representational model based on the masked language model and pre-trained using bidirectional transformers [23]. BERT uses a masked language model that predicts

<sup>2</sup><https://thehelloworldprogram.com/python/python-string-methods/>

**Table 2**  
Experimental Environment

Name	Features
CPU	Intel(R) Xeon(R) W-2155 CPU @ 3.30GHz
Number of CPU	20
L1d cache	32K
L1i cache	32K
L2 cache	1024K
L3 cache	14080K
RAM	64 GB
Operating System	Ubuntu 18.04 LTS
HDD	2 TB
Programming Language	Python
Version of the language	3.7

randomly masked words in a sequence and hence can be used for learning bidirectional representations. Also, it obtains state-of-the-art performance on most NLP tasks while requiring minimal task-specific architectural modification. As the BERT integrates the information from bidirectional representations, we believed that such bidirectional representations are crucial in mathematical information retrieval as complex relationships between mathematical terms often exist in scientific documents. Primarily, the BERT model pretrained using English Wikipedia and BooksCorpus and proposing new pre-training objectives: the Masked Language Model (MLM) and Next-sentence Prediction (NSP). The MLM task randomly replaces 15% of the input words into “masked” tokens and predicts them; 80% and 10% of the masked tokens are replaced by [MASK] and random tokens, respectively, while 10% remains unchanged. The BERT model enables contextualized formula embedding by learning deep bidirectional representations through the MLM task. The proposed formula embedding module is mapping formulas into vectors to enable computers to understand mathematical language. The NSP task determines whether two formulas are mathematically associated. Through this module, BERT learned to understand the relationship among formulas and mapped their similarities. Several BERT models are available depending on their size, including BERT-Base (12 layers, 768 hidden size, 12 attention heads, and 110 million parameters) and BERT-Large (24 layers, 1024 hidden size, 16 attention heads, and 340 million parameters). In this work, we have used the BERT-Base model.

## 5. Experimental setups

The experimental platform is a standalone Ubuntu 18.04 desktop to validate our claim. The configuration of the experimental environment is demonstrated in Table 2. At the time of the experiment, we have carefully validated our approach and avoided any kind of noise.

**Table 3**  
Evaluation Results

Query	Approach	Data	nDCG'	MAP'	P@10'
ARQMath-2020	Formula Embedding	Math	0.233	0.140	0.271
ARQMath2-2021	Variable-Size Vector Based Approach	Math	0.091	0.032	0.151
	Formula Embedding_A	Math	0.114	0.039	0.152
	Formula Embedding_P	Math	0.161	0.059	0.197

## 6. Experimental Results

The experimental results of the BERT-based formula embedding approach revealed several characteristics of mathematical formulae, which indicated that the natural language embedding models are potentially useful for the formula embedding task. The results value for the topics released in ARQMath and ARQMath2 are shown in Table 3. The obtained results have shown that the pre-trained embedding model can handle the mathematical representation and able to preserve their syntactic meaning. In ARQMath-2021, we have submitted a total of four runs, one for the topics provided in the ARQMath-2020 and the remaining three-runs for the topics provided in the ARQMath-2021. The run submitted for the ARQMath-2020 topics is based on the BERT-based formula embedding approach, which has been trained only on the 20 million  $\LaTeX$  formula. Moreover, the first run of the ARQMath2-2021 has been obtained from our earlier system [14] that we have designed in ARQMath-2020. This system has used the formula in Presentation MathML format, which is almost 26075012. In addition to this, the second and third runs of the ARQMath2 have been obtained from the BERT-based formula embedding approach. The second run has been trained only on the 10 million  $\LaTeX$  formula, and the third run has been trained only on the 20 million  $\LaTeX$  formula. The reasons behind these formula selections are inaccessible computing power and unavailability of training time.

The aim of the test queries is to verify the properties of math-aware search engine like retrieval of sub-formula, parent-formula, similar formula and nearly-similar formula which are briefly explained as follows:

- Sub-formula is a part of a formula. For example, the obtained results of query B.48, which are depicted in Table 4 where BERT-based formula embedding model effectively handles the retrieval of sub-formula.
- Parent formula is a formula that holds the existence of the queried formula. For example, the obtained results of query B.30 shown the evaluation of the parent formula search for queried formula  $a^3 + b^3 + c^3 - 3abc$ .
- The nearly-similar formula is a formula that has a similar meaning to the formula in the dataset. For example, the obtained results of the queried formula B.12 depict the retrieval accuracy of the nearly-similar search.
- A similar formula is a formula which is semantically similar or considerably similar to a queried formula, such as retrieved results of query B.60.

Table 4 shows the sample test queries with their relevant formulas present in MSE corpus. The relevant formulas are ordered by their similarities to the queries.

**Table 4**  
Retrieved Search Results

Query ID	Query Formula	Retrieved formula
B.12	$(1 + i\sqrt{3})^{1/2}$	$(2 + \sqrt{2})^{1/3}$
		$(1 - \sqrt{3}i)^{1/2}$
		$(1 + i\sqrt{3})/2$
		$(-1 + \sqrt{3}i)/2$
		$(-1/2 + \sqrt{3}i/2)$
B.30	$a^3 + b^3 + c^3 - 3abc$	$a^3 + b^3 + c^3 - 3abc$
		$a^3 + b^3 + c^3 - 3abc?$
		$n = a^3 + b^3 + c^3 - 3abc$
		$3 (a^3 + b^3 + c^3 - 3abc)$
		$p = a^3 + b^3 + c^3 - 3abc$
B.48	$(x + y)^k \geq x^k + y^k$	$x^k \geq y^k$
		$(x^k + y^k) < (x + y)^k$
		$(x + y)^k \geq 0$
		$(x + y)^k \geq x + ky$
		$(x + y)^n \geq x + ny$
B.60	$\lim_{n \rightarrow \infty} a_n$	$\limsup_{n \rightarrow \infty} A_n$
		$\lim_{n \rightarrow \infty} a_n = L$
		$\limsup_{n \rightarrow \infty} a_n$
		$\lim_{n \rightarrow \infty} a_n$
		$\lim_{n \rightarrow \infty} a_n = 0$
B.80	$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \{4\}, \dots$	$\emptyset, \{1\}, \{2\}, \{1, 2\}, \{3\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}, \{4\}, \dots$
		$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$
		$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$
		$\{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{3, 2\}, \{1, 2, 3\}, \emptyset\}$
		$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, S$

## 7. Conclusions and Future Scope

Mathematical information retrieval has been researched consciously in recent years, and there have been many productive results. In this paper, we have reported the contribution of our BERT-based formula embedding model in ARQMath2-2021. The proposed BERT-based model trained on the math exchange corpus of the ARQMath where it takes the  $\LaTeX$  formulas as input and produced the fixed dimensional embeddings for the same. For similarity computation, cosine similarity has been used. The experimental results support the hypothesis that a pre-trained model for natural language processing tasks positively responds to mathematical language processing tasks. The obtained results have shown that the proposed model has better retrieval accuracy than our earlier system and shows a remarkable contribution.

In future studies, we will integrate the textual information with a formula to achieve better retrieval accuracy for semantically similar formulas. Moreover, we aim to test other recently developed embedding models like Sentence Deep Bidirectional Transformers (SBERT) and Deep Contextualized Word Representations (ELMo), which are computationally more expensive but have the better embedding formation ability.

## Acknowledgments

The authors would like to express gratitude to the Department of Computer Science and Engineering and Center for Natural Language Processing, National Institute of Technology

Silchar, India for providing infrastructural facilities and support.

## References

- [1] F. Guidi, C. S. Coen, A survey on retrieval of mathematical knowledge, *Mathematics in Computer Science* 10 (2016) 409–427.
- [2] Q. Zhang, A. Youssef, An approach to math-similarity search, in: *International Conference on Intelligent Computer Mathematics*, Springer, 2014, pp. 404–418.
- [3] S. Lai, K. Liu, S. He, J. Zhao, How to generate a good word embedding, *IEEE Intelligent Systems* 31 (2016) 5–14.
- [4] J. Guo, Y. Fan, Q. Ai, W. B. Croft, A deep relevance matching model for ad-hoc retrieval, in: *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016, pp. 55–64.
- [5] Y. Stathopoulos, S. Baker, M. Rei, S. Teufel, Variable typing: Assigning meaning to variables in mathematical text, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 303–312.
- [6] G. Y. Kristianto, M.-Q. Nghiem, Y. Matsubayashi, A. Aizawa, Extracting definitions of mathematical expressions in scientific papers, in: *Proc. of the 26th Annual Conference of JSAI*, 2012, pp. 1–7.
- [7] G. Y. Kristianto, G. Topić, A. Aizawa, Exploiting textual descriptions and dependency graph for searching mathematical expressions in scientific papers, in: *Ninth International Conference on Digital Information Management (ICDIM 2014)*, IEEE, 2014, pp. 110–117.
- [8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, *arXiv preprint arXiv:1603.01360* (2016).
- [9] M. Rei, G. K. Crichton, S. Pyysalo, Attending to characters in neural sequence labeling models, *arXiv preprint arXiv:1611.04361* (2016).
- [10] S. Dhar, S. Roy, Mathematical document retrieval system based on signature hashing, *Aptikom Journal on Computer Science and Information Technologies* 4 (2019) 45–56.
- [11] A. Pathak, P. Pakray, A. Gelbukh, Binary vector transformation of math formula for mathematical information retrieval, *Journal of Intelligent & Fuzzy Systems* 36 (2019) 4685–4695.
- [12] A. Pathak, P. Pakray, R. Das, Lstm neural network based math information retrieval, in: *2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*, IEEE, 2019, pp. 1–6.
- [13] R. Zanibbi, D. W. Oard, A. Agarwal, B. Mansouri, Overview of arqmath 2020: Clef lab on answer retrieval for questions on math, in: *International Conference of the Cross-Language Evaluation Forum for European Languages*, Springer, 2020, pp. 169–193.
- [14] P. Dadure, P. Pakray, S. Bandyopadhyay, An analysis of variable-size vector based approach for formula searching, in: *Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum*, 2020.
- [15] B. Mansouri, D. W. Oard, R. Zanibbi, Dprl systems in the clef 2020 arqmath lab, in: *Working Notes of CLEF 2020-Conference and Labs of the Evaluation Forum*, 2020.

- [16] V. Novotný, P. Sojka, M. Štefánik, D. Lupták, Three is better than one: Ensembling math information retrieval systems, Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum Thessaloniki, Greece (2020).
- [17] A. Pathak, P. Pakray, A. Gelbukh, A formula embedding approach to math information retrieval, *Computación y Sistemas* 22 (2018) 819–833.
- [18] M. Anandarajan, C. Hill, T. Nolan, Term-document representation, in: *Practical Text Analytics*, Springer, 2019, pp. 61–73.
- [19] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, Semeval-2012 task 6: A pilot on semantic textual similarity, in: *First Joint Conference on Lexical and Computational Semantics*, 2012, pp. 385–393.
- [20] F. Rahutomo, T. Kitasuka, M. Aritsugi, Semantic cosine similarity, in: *The 7th International Student Conference on Advanced Science and Technology ICAST*, volume 4, 2012, p. 1.
- [21] P. Dadure, P. Pakray, S. Bandyopadhyay, An empirical analysis on retrieval of math information from the scientific documents, in: *International Conference on Communication and Intelligent Systems*, 2020, pp. 301–308.
- [22] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, *arXiv preprint arXiv:1706.03762* (2017).