An Analysis of Variable-Size Vector Based Approach for Formula Searching

Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay

Department of Computer Science and Engineering
National Institute of Technology Silchar, India
{krdadure, parthapakray and sivaji.cse.ju}@gmail.com

Abstract. The continuously increasing research in the field of science, engineering, and technology has generated textual and mathematical data in huge amounts. The research in retrieval and searching of textual data achieved the state-of-the-art results while searching and retrieval of mathematical information is in the early stage of research and requires significant improvement. Motivated from the concept of formula embedding and term-document matrix, in this paper we have introduced the variable size formula embedding approach where the formula is transformed into the variable size vector. In a vector, each bit represents their occurrence and corresponds to their position in BPIT. The proposed approach has been tested on the Math type data of Math Stack Exchange of ARQMath task and the proficiency of the same are represented in terms of nDCG´, MAP´ and Precision at 10 measures. The obtained results have shown that the approach of variable size formula embedding requires significant improvement to retrieve the syntactically and semantically similar formula.

Keywords: Searching · Math Stack Exchange · Term-Document Matrix · Formula Embedding.

1 Introduction

Math formulas are the key constitutes in any scientific documents or Math-Based Question Answering post to communicate and deliver the idea behind it. Nowadays, the web is a well-known information sharing platform where user's share their knowledge and opinion. The process of information sharing creates a huge amount of data which consists of text, images, videos, etc. However, the data generated from the knowledge sharing sites mainly contains textual and mathematical information. The retrieval of such information based on the mathematical formula is a complex and laborious one. Moreover, the number of textual information retrieval systems have been developed and exhibit their retrieval potential as well. On the contrary, Mathematical Information Retrieval

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

CEUR-WS.org/Vol-2696/paper_150.pdf

(MIR) have faced the many barriers which leads to retrieval of irrelevant information. For instance, the mathematical symbols owned the predefined scripting styles which are qualitatively different from linear textual information. In mathematical information, some mathematical notations posses the alternative representation like a permutation of k objects chosen from n distinct objects have several representations: P_k^n, ${}_nP_k$, $P_{n,k}$, ${}^nP_k$ and P(n, k). In scientific documents, sometimes, some mathematical notation holds multiple meanings. For instance, P(x) depicts either the probability of event "x" or multiplication of variable "P" and "x". In such cases, the context is a significant factor to deliver the actual idea of mathematical notations. In the retrieval of mathematical information, user query depicts the user need and a well-formed user query may lead to a more relevant search. The well-formed user query has the potential to retrieve the relevant search and satisfy the user's needs. To address the above-mentioned obstacles, several approaches have been implemented and light-up the research of mathematical information retrieval. Although still there is a huge scope of improvement in the research field of mathematical information retrieval.

The ARQMath lab at CLEF 2020[1] aims to push-up the state-of-the-art evaluation design of math-aware information retrieval, and which seek to support the development and ultimate deployment of new techniques. There are two tasks namely Answer Finding (Main Task) and Formula Search (Secondary Task) are organized by ARQMath Lab at CLEF 2020. In answer finding task, the system returns the ranked list of answers post for the user query selected from the question post of Math Stack Exchange[2] (MSE) whereas the formula search task returns the ranked list of formulas from MSE which are synthetically or semantically similar to queried formula. From these two tasks, we are mainly focused on the formula search task and contributed a variable-size vector based technique. The data provided by the ARQMath organizer has contained the posts which hold the textual information and mathematical formulas. As per the flexibility provided by the task organizer, the participant can use both textual and formula information or only textual information or only formulas. In our proposed approach, we have used only the formulas extracted from the Math Stack Exchange site.

The major contribution of the proposed work are given as follow:

- The proposed approach transformed the formula into a variable size vector of weight.
- Each weight of the vector represents the occurrence of a particular entity in a formula and correspond to their position in BPIT.
- The proposed approach has been tested on the formulas expressed in Presentation MathML format which are extracted Math Stack Exchange.

The structure of the paper is as follows: Section 2 describes related work in the domain of formula retrieval. Section 3 illustrates the corpus description and proposed methodology. Section 4 covered the experimental results and their analysis. Section 5 concludes the paper with a key future direction.

[1] https://www.cs.rit.edu/ dprl/ARQMath
[2] https://math.stackexchange.com

2 Related Work

The retrieval of mathematical information has seen growing attention in the last decade and since the mid-1990's the search for Math Formula has been studied. The past research works have addressed the number of mathematical information retrieval challenges. However, there is a huge scope for improvement to mitigate the challenges. In the retrieval of mathematical information, the formula embedding approach [12] transformed the formula into the vector of size 202 where "0" depicts the presence and "1" depicts the absence of a particular entity in the formula. The three layer model of formula representation and searching [4] where the first layer choose candidates using spectral matching over tree node pairs, the second layer aligned a query with candidates and estimates the similarity based on spectral matching, and the third layer estimates the similarity score of two representation using linear regression.

To furnish the retrieval results for formula query, the Tangent-CFT [10] have used the two effective representations of mathematical information i.e. Symbol Layout Trees (SLTs) and Operator Trees (OPTs) which may contribute to the more accurate retrieval. In which, symbol layout tree focused on the appearance of the formula whereas the operator tree focused on the content of the formula. In which, the tuples have been generated based on the path between the pair of symbols and embed them using the fast n-gram embedding model. For state-of-the-art performance, the Tangent-CFT combined the SLTs and OPTs embeddings and uses the structural similarity for partial-match formula retrieval. In the research field of MIR, the transformation of mathematical information language to natural language is an arduous process. To highlight this transformation, AnnoMathTex- a recommender system [14] enables the formula annotation by assigning a meaning to formula identifiers from the surrounded text.

In mathematical information retrieval, relevance measurement is a significant ingredient. For instance, the estimation of the cosine similarity between the indexed formulas and query formula can also lead to some useful relevant results [3]. The state-of-the-art results of deep learning techniques in textual information retrieval motivate the researcher to incorporate it in the retrieval of mathematical information. The positive impact of LSTM in sequence-to-sequence problem, LSTM neural network based formula entailment approach [11] formulated the entailment between the index formulas and query formula. In substructural matching of formulas, formula retrieval requires the high computation time. To speed up the formula search in sub-structural similarity, dynamic pruning strategies, and specialized inverted index produced noticeable outcomes [17]. For further improvement, the query structure representation is associated with posting lists to boots the overall outcomes of the sub-tree matching. To increase the accessibility of scientific documents, the MathAlign system [1] has introduced the rule-based approach which extracts the latex form of formula and linked the identifiers of extracted formula to their text description.

To analyze the frequency distributions of mathematical expressions in the large scientific corpus, André et al. [6] have incorporated the Zipf's law on a text

corpus. Motivated from the similarity in linguistic properties, they have introduced a novel approach to rank formulae by their relevance via a customized version of the ranking function BM25. They have also demonstrated the applicability of the results by presenting auto completion for math inputs as the first contribution to math recommendation systems.

Motivated from the emerging association between mathematical formulas and the textual context in scientific documents, the topic model called TopicEq [15] has generated the context from a mixture of latent topics, and the equation has generated by an RNN that depends on the latent topic activation and enables intelligible processing of equations by considering the relationship between the mathematical equations and topics. To investigate the feasibility of neural representation techniques in MIR, the symbol2vec approach [5] learn the vector representations of formula symbols. For more refined results, the textual information has combined with the Formula2vec model and achieves better retrieval performance. To learn distributed representations of equations, the unsupervised approach called equation embeddings (EqEmb) [9] where the equation has been treated as singleton word. In which, the semantic representations of mathematical equations and their surrounding words have embedded and obtained results are compared with the CBOW, PV-DM, GloVe model where EqEmb-U achieves the highest performance. The exploratory investigation of the effectiveness and use of word embedding [7] where word2vec models have trained on DLMF and arXiv with slightly different approaches for embedding math. The DLMF trained model discovered the mathematical term similarities and term analogies and generates the query expansions. The arXiv trained model has beneficial to extract the textual descriptive phrases for math terms. The word embedding model mainly focused on term similarity, math analogies, concept modeling, query expansion, and knowledge extraction.

3 Methodology

3.1 Data Description

To evaluate the performance of the proposed formula retrieval approach, the task organizer provides the dataset collected from the knowledge sharing platform i.e. Math Stack Exchange (MSE). The dataset comprised the question, answer, and comment posts. As per the flexibility on data, the participant of the task can use the only formula or only textual or both (formula and textual information) to perform the formula search. In the dataset, the formulas are represented in three different formats i.e. L^AT_EX, Presentation MathML, and Content MathML format. These formulas are extracted from the posts of Math Stack Exchange of the year 2010-2018. The number of formulas comprised in L^AT_EX, Presentation MathML and Content MathML formats are 28,320,920, 26,075,012 and 25,366,913 of size 1.5 GB, 11.5 GB and 10.9 respectively. Each format has five distinct attributes namely formula_id, post_id, thread_id, type, and formula. To execute the formula search task, we have used only a formula dataset and out of these three formula representation files, we have used only

the Presentation MathML format. The dataset are available at the official site of ARQMath task[3]. The metadata about the formula dataset is shown in table 1.

Table 1. Data Description

Corpus	Math Stack Exchange
Type	Formula
Format	Presentation MathML
Size	11.5 GB
No. of formulas	26,075,012

3.2 Query Dataset

To estimate the performance of the proposed work, the task organizer provided 87 mathematical formulae, each of the formula is selected from the question's topic of task 1. The topics (queries) for the formula search task provided in an XML file that has a predefined format as shown in figure 1. Each topic in XML file tagged by <topic> and </topic> tag and each topic has a unique topic number i.e. B.x where "x" represents the topic number. Formula_Id shows the id of formula, Latex shows the latex representation of formula, Title shows the question title of the post from which the formula is selected, Question shows the question body from which the formula is selected and Tags shows the comma-separated tags of the question.

```
<Topics>
    <Topic number="B.x">
        <Formula_Id>q_x</Formula_Id>
        <Latex>formula latex representation</Latex>
        <Title> question title </Title>
        <Question> question body </Question>
        <Tags> list of comma separated tags </Tags>
    </Topic>
            ...
</Topics>
```

Fig. 1. Topic representation

[3] https://www.cs.rit.edu/ dprl/ARQMath/

3.3 System Architecture

The system architecture of the proposed approach is shown in figure 2. The proposed system architecture is inspired by the existing formula embedding approach [12] and term-document matrix [2] where each module work interdependently to make the faster retrieval and accurate search.

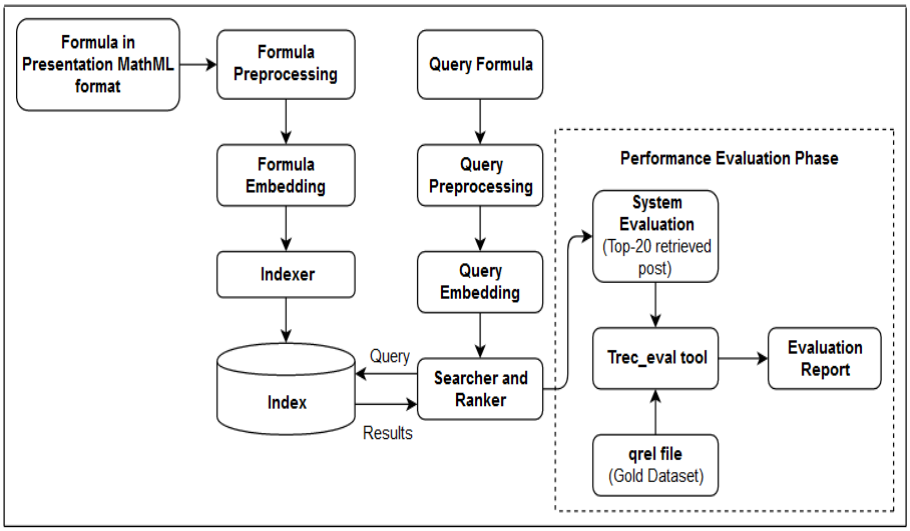

Fig. 2. Proposed system architecture

3.3.1 Formulas: The formulas are extracted from the posts of Math Stack Exchange from 2010-2018 in three different format L^AT_EX, Presentation MathML and Content MathML. To investigate the ability of the proposed approach in the formula search task, we have used only the Presentation MathML format.

3.3.2 Formula Preprocessing: The prime task of the formula preprocessing module is to transform the formulas into the unified form by removing irrelevant elements and attributes. In this process, the preprocessing module trim the tags to their root form for example <mi mathvariant=““normal””> trimmed to <mi>, <mo rspace=““4.2pt””> trimmed to <mo> etc. Some examples of the preprocessed tags are shown in Table 2. The formula preprocessing module also discarded a few Presentation MathML tags like <mtext>, <mspace>, <mstyle> etc. as these tags does not have much contribution regarding semantic of the mathematical notation.

Table 2. Some examples of preprocessing done by formula Preprocessor

Original	Preprocessed
<mi class= ““ltx_font_mathcaligraphic””>	<mi>
<mi mathvariant=““normal””>	<mi>
<mo movablelimits=““false””>	<mo>
<mo stretchy=““false””>	<mo>
<mo fence=““true”” stretchy=““false””>	<mo>
<mover accent=““true””>	<mo>
<mo rspace=““4.2pt””>	<mo>
<mo largeop=““true”” movablelimits=““false”” symmetric=““true””>	<mo>

3.3.3 Formula Embedding: The proposed formula embedding approach motivates from the existing Bit Position Information Table (BPIT) [12] and Term-Document matrix [2] which transformed the formulas into the variable size vector of weight. Each weight of vector represents the occurrence count of a particular entity in a formula and correspond to entity position in BPIT. The process of formula to variable size vector transformation is shown in figure 2. In the proposed formula embedding approach, entities in a formula are categorized into the three categories based on the tags. The entities tagged by <mi> comes under the first category, entities tagged by <mo> comes under the second category and the third category holds the essential MathML tags which contribute to the semantics of 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 3, 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.

3.3.4 Indexer: In the process of formula searching, the effective indexing is the key constituent to speeding up the searching process. 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.

3.3.5 Query Preprocessing: To test the formula searching effectivity of the proposed approach, the ARQMath task organizer provided the queries (Topics) in L^AT_EX format as described in section 3.2. As we have selected the Presentation MathML format of formula for formula embedding and to maintain the

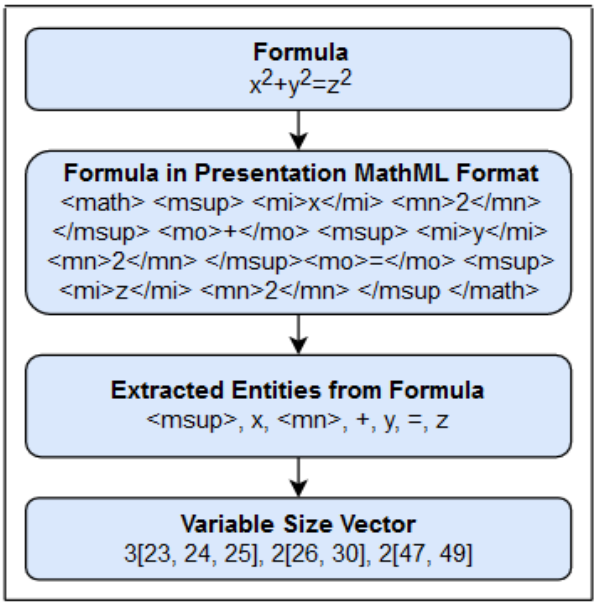

Fig. 3. Formula to Vector Transformation

unified structure between formulas and queries, we have transformed the queries into the Presentation MathML format using the tool Demo MathType[4].

3.3.6 Query Embedding: Query embedding module converts the preprocessed Presentation MathML query formula into a variable size vector. For query vector generation, the query embedding module considered the entity position format of BPIT [12]. In a generated vector, each weight represents the occurrence of a particular entity in a query formula and their position in BPIT. In MIR, user's expressed their need in the form of formula and expects that the system returns the documents/posts which contain syntactically and semantically similar formulae.

3.3.7 Searcher and Ranker Module: The main objective of the searcher module is to search for relevant formulas that are syntactically or semantically similar to query formula and satisfy the user's need. Searching for relevant information is a time consuming process and requires the effective formula representation and indexing technique [8]. In the proposed architecture, the searcher module compared the query formula vector with all the indexed formula vectors and computes the similarity for each indexed formula vector. For similarity calculation, the proposed approach compared each bit of query vector with the

[4] http://www.wiris.com/editor/demo/en/developers

bits of formula vector and those formulas contained the maximum number of similar bit that formula have maximum similarity score. The process of similarity calculation is shown in figure 4 where the occurrence count of <mi>, <mo> and essential MathML tags of formula vector are compared with the occurrence count of <mi>, <mo> and essential MathML tags of query vector. The reason behind the consideration of occurrence count is to help in assigning a priority to those formulae which have a similar number of <mi>, <mo>, and essential MathML tags.

After a successful comparison and similarity calculation between the formula vectors and query vector, the ranker module retrieves and ranks the post (the post which contains the search formula) based on the higher similarity score. Those post of MSE contains more than one similar formula with respect to query formula, that post of MSE assigned higher priority as compared to those post which contains the formula only one time. As a final search result with respect to query formula, we have retrieved the top 20 posts of MSE which contains the relevant formulas.

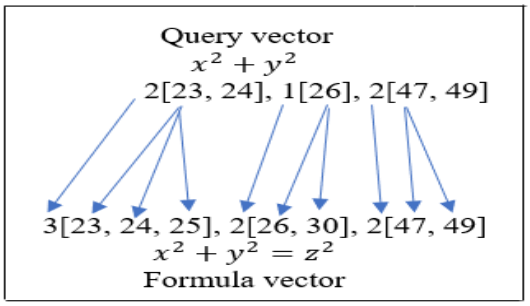

Fig. 4. The process of similarity calculation

4 Results and Discussion

The results of the proposed approach have been estimated using the 45 queries out of provided 87 queries [16]. For each query formula, the proposed approach retrieves the top 20 posts which contain the relevant formulas and ranked them based on the maximum similarity score. Those post holds the more than one similar formulas that post get the higher priority as compared those posts which contained the formula only once. The obtained search results stored in TSV file which have six attributes namely query_id, formula_id, post_id, rank, score and run number where query_id represents the query unique id in the form of "B.x" where x is the query number, formula_id represents the unique identifier for the retrieved formula instance, post_id represents the unique identifier of the post where the formula is contained, rank attributes represents the rank of retrieved

formula, score attribute represents the similarity score between the query formula and index formula and run number represents the number of runs submitted by the participant.

For evaluating the performance of the proposed approach, trec_tool[5] is used, which compared the gold dataset (qrel file) with the result set obtained from the proposed system. The obtained results of the proposed work for the formula search task are shown in table 3 [16]. All the obtained results of the participants have been ranked based on obtained nDCG´ metric [13]. For generic and more comparative analysis, the results are also estimated in terms of MAP´ and P@10. The results of the baseline system have achieved the highest score as compared to all the participant teams. The obtained result disclose that the proposed approach requires significant improvement to generated the state-of-the-art result for formula search and retrieval tasks.

The nDCG´ measure is based on nDCG and is a commonly used scale when ratings for relevance judgments are available and a single value figure is generated over a series of ranking lists. The retrieved document receives a gain value (0, 1, 2, or 3) and the rank of each post is gradually decreasing. The discounted gain values result are obtained and then normalized to [0,1] by dividing the maximum discounted cumulative gain possible which is called normalized Discounted Cumulative Gain (nDCG). The only difference when nDCG's is calculated is that unjudged posts are discarded before the measurement is performed. In addition, the nDCG´ produces a single measure with graded relevance, while Precision@k and Mean Average Precision (MAP) require binarized judgments in terms of relevance [13]. Moreover to nDCG´, the MAP´ is also calculated by removing unassessed posts and precision 10 posts (P@10).

Table 3. Formula search task obtained results

Team/Approach	Data	P	nDCG´	MAP´	P@10
Baseline					
Tangent-S	Math	√	0.506	0.288	0.478
DPRL					
TangentCFTED	Math		0.420	0.258	0.502
TangentCFT	Math		0.392	0.219	0.396
TangentCFT+	Both	√	0.135	0.047	0.207
MIRMU					
SCM	Math		0.119	0.056	0.058
Formula2Vec	Math	√	0.108	0.047	0.076
Ensemble	Math		0.100	0.033	0.051
Formula2Vec	Math		0.077	0.028	0.044
SCM	Math	√	0.059	0.018	0.049
NLP_NITS					
FormulaEmbedding	**Math**	√	**0.026**	**0.005**	**0.042**

[5] https://github.com/usnistgov/trec_eval

The results shown in table 4 discovered that the proposed formula embedding approach can retrieve the exact match formula with respect to query formula. The outcomes of $(1 + \sqrt{3}i)^{1/2}$ retrieved the syntactically similar formula and which shows the positive impact of the formula embedding in the retrieval of mathematical information. The proposed formula embedding approach has efficiently retrieved all those formulae which hold the maximum number of similar entities and rank them based on the maximum similarity.

Table 4. Best relevant retrieved search results

Topic id	Query/Topic	Retrieved Formulas
B.12	$(1 + i\sqrt{3})^{\frac{1}{2}}$	$(1 + \sqrt{3}i)^{1/2}$
		$(1 + i\sqrt{3})/2$
		$(1 + i\sqrt{3})^{n}$
		$z_3 = -1/2(1 - i\sqrt{3})$
		$\frac{1}{2}(1 + i\sqrt{3})$

The results shown in table 5 depict that the approach of formula embedding has the potential to retrieve the partial match formula with respect to query formula. The obtained result has shown that the proposed approach retrieves the parent formula or subformula which holds similar entities with respect query formula.

Table 5. Partial relevant retrieved search results

Topic id	Query/Topic	Retrieved Formulas
B.53	$AB = 1 \Rightarrow BA = 1$	$b = 8 \Rightarrow a = 4$
		$ab = b \Rightarrow b = 0$ or $a = 1$
		$b = 0 \Longrightarrow a = 3$
		$ab = a0 \Longrightarrow b = 0$
		$a = 1 \Longrightarrow b = 2$

The frailty of the formula embedding approach is shown in table 6, which depicts that the formula embedding approach sometimes retrieves the totally irrelevant result. The proposed approach effectively retrieves the relevant formulas as well but in some cases, it retrieves irrelevant formulas also.

Table 6. Irrelevant retrieved search results

Topic id	Query/Topic	Retrieved Formulas
B.63	$lcm(n_1, n_2) = \frac{n_1, n_2}{gcd(n_1, n_2)}$	$T_n = (0, \frac{1}{n})$ $B_n = (\frac{1}{n}, 1)$ $A_n = (0, \frac{1}{n})$ $n_2 = (\frac{1}{2}, \frac{1}{2}, 1)$ $I_n = (\frac{1}{2}, \frac{3}{2})$

5 Conclusions and Future Scope

The objective of this paper is to analyze the performance of the formula embedding approach which transformed the formula into the variable size vector. Each weight of vector is an occurrence of a particular entity in a formula and corresponds to their position in BPIT. To estimate relevance between the index formula vector and query vector, each bit of the query vector is compared with bits of index formula vector. The feasibility of this approach has been tested on the Math Stack Exchange corpus of the ARQMath task and the obtained results revealed the robustness and frailness of the same. The proposed approach has the ability to retrieves the syntactically similar formula, subformula, or parent formula. To compared the obtained results, nDCG´ metric is considered as the primary measure and additionally MAP´ and Precision at 10 measure also estimated for more generic comparison.

Our future target is to improve the ranking mechanism by assigning priorities to entities while calculating the similarities. To reduce the searching and retrieval time, index optimization is one of our future task which helps to fast the retrieval and formula searching process. To enrich the efficiency of the formula search process and retrieval of semantically similar formula, textual data will be incorporated with the mathematical data.

Acknowledgment

The authors would like to express their gratitude to the Department of Computer Science and Engineering and Center for Natural Language Processing, National Institute of Technology Silchar, India for providing the infrastructural facilities and support.

References

1. Alexeeva, M., Sharp, R., Valenzuela-Escárcega, M.A., Kadowaki, J., Pyarelal, A., Morrison, C.: Mathalign: Linking formula identifiers to their contextual natural

language descriptions. In: Proceedings of The 12th Language Resources and Evaluation Conference. pp. 2204–2212 (2020)
2. Anandarajan, M., Hill, C., Nolan, T.: Term-document representation. In: Practical Text Analytics, pp. 61–73. Springer (2019)
3. Dadure, P., Pakray, P., Bandyopadhyay, S.: An empirical analysis on retrieval of math information from the scientific documents. In: International Conference on Communication and Intelligent Systems. pp. 301–308. Springer (2019)
4. Davila, K., Zanibbi, R.: Layout and semantics: Combining representations for mathematical formula search. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1165–1168 (2017)
5. Gao, L., Jiang, Z., Yin, Y., Yuan, K., Yan, Z., Tang, Z.: Preliminary exploration of formula embedding for mathematical information retrieval: can mathematical formulae be embedded like a natural language? arXiv preprint arXiv:1707.05154 (2017)
6. Greiner-Petter, A., Schubotz, M., Müller, F., Breitinger, C., Cohl, H., Aizawa, A., Gipp, B.: Discovering mathematical objects of interest—a study of mathematical notations. In: Proceedings of The Web Conference 2020. pp. 1445–1456 (2020)
7. Greiner-Petter, A., Youssef, A., Ruas, T., Miller, B.R., Schubotz, M., Aizawa, A., Gipp, B., et al.: Math-word embedding in math search and semantic extraction. Scientometrics pp. 1–30
8. Jansen, B.J., Rieh, S.Y.: The seventeen theoretical constructs of information searching and information retrieval. Journal of the American Society for Information Science and Technology **61**(8), 1517–1534 (2010)
9. Krstovski, K., Blei, D.M.: Equation embeddings. arXiv preprint arXiv:1803.09123 (2018)
10. Mansouri, B., Rohatgi, S., Oard, D.W., Wu, J., Giles, C.L., Zanibbi, R.: Tangent-cft: An embedding model for mathematical formulas. In: Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval. pp. 11–18 (2019)
11. Pathak, A., Pakray, P., Das, R.: Lstm neural network based math information retrieval. In: 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP). pp. 1–6. IEEE (2019)
12. Pathak, A., Pakray, P., Gelbukh, A.: A formula embedding approach to math information retrieval. Computación y Sistemas **22**(3), 819–833 (2018)
13. Sakai, T., Kando, N.: On information retrieval metrics designed for evaluation with incomplete relevance assessments. Information Retrieval **11**(5), 447–470 (2008)
14. Scharpf, P., Mackerracher, I., Schubotz, M., Beel, J., Breitinger, C., Gipp, B.: Annomathtex-a formula identifier annotation recommender system for stem documents. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 532–533 (2019)
15. Yasunaga, M., Lafferty, J.D.: Topiceq: A joint topic and mathematical equation model for scientific texts. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 7394–7401 (2019)
16. Zanibbi, R., Oard, D., Agarwal, A., Mansouri, B.: Overview of arqmath 2020: Clef lab on answer retrieval for questions on math. In: ARQMath Lab @ CLEF 2020. pp. 1–25 (2020)
17. Zhong, W., Rohatgi, S., Wu, J., Giles, C.L., Zanibbi, R.: Accelerating substructure similarity search for formula retrieval. In: European Conference on Information Retrieval. pp. 714–727. Springer (2020)