The Math Retrieval System of ICST for NTCIR-12 MathIR Task

Liangcai Gao ,Ke Yuan , Yuehan Wang, Zhuoren Jiang, Zhi Tang
Institute of Computer Science & Technology of Peking University
{glc, yuanke, wangyuehan, jiangzr, tangzhi}@pku.edu.cn

ABSTRACT

This paper is the summarized experiences of ICST team in the NTCIR-12 MathIR main tasks (ArXiv and Wikipedia main task). Our approach is based on keyword, structure and importance of formulae in a document. A novel hybrid indexing and matching model is proposed to support exact and fuzzing matching. In this hybrid model, both keyword and structure information of formulae are taken into consideration. In addition, the concept of formula importance within a document is introduced into the model. In order to make the ranking results more reasonable, our system (WikiMir) applies the learning to rank algorithm (RankBoost) to rank the retrieved formulae , and then re-ranks the top-k formulae by the regular expressions matching of the query formula. The experimental results show that the method of our system is effective for all metrics and promising in practical application.

Team Name

ICST

Subtasks

MathIR arXiv Main Task, optional MathIR Wikipedia Task

Keywords

Mathematical Information Retrieval, Structure Matching, Learning to Rank, Re-rank, formula importance

1. INTRODUCTION

As an important part of scientific documents, mathematical formulae are usually used to explain the theoretical foundation of these scientific documents. It is not easy for researchers to comprehend the unfamiliar formulae. And a formula-based search engine is expected to help them to find some related formulae resources and understand the formulae further.

Mathematical Information Retrieval (MIR) systems are formula-based search engine. Unlike text, formulae are presented in special formats, such as *LaTeX*, MathML. Meanwhile, formulae are highly structured. For instance, $a + b$ is the sub-structure of y_{a+b} . Due to the characteristics of formulae, the methods of traditional search engines which mainly consider about the plain text retrieval are inapplicable for MIR systems. Moreover, different formulae are usually play different roles in a document, the significant ones should be paid more attention to.

Focus on the problems mentioned above, we proposed a mathematics information retrieval system, namely WikiMir. It is a publicly available hybrid MIR system, based on the keyword, structure and importance of formulae in a document. Meanwhile, WikiMir applies the learning to rank algorithm (RankBoost) to rank the retrieved formulae, and then re-rank the top-k formulae by the regular expressions of query formula to improve the ranking performance. Firstly, WikiMir processes the query formula and the formulae in the dataset into a uniform format (e.g., Presentation mathml). Secondly, a hybrid index consisting of a formula index and a context index is introduced to leverage more comprehensive mathematical information. Furthermore, the importance of formulae in a document is calculated to distinguish the roles of the target formulae in the documents. Finally, we use the supervised method to rank the retrieved formulae, and re-rank the top-k formula by the regular expression of query formula after that.

2. Related Work

2.1 MIR system

Mathematical information retrieval systems are sprouting out since Miller and Youssef [1] first proposed the definition of mathematical information retrieval. An intact MIR system includes interface, tokenizer, indexer and ranker. We analyze the relevant approaches on these parts as follow:

Interface: For MIR systems, formulae are usually taken as queries. The presentation of formula is important since it denotes the formula internal format and determines the compatibility of a mathematics retrieval system to the existing data sources. Most MIR systems use the Presentation MathML format of formulae that can express the structural meaning of formulae [2, 3, 4 and 5].

Tokenizer: Tokenizer is the most challenge part of MIR systems. The tokenizer is used to parse the formulae into terms that are utilized to build the index files. Some MIR systems apply the text-oriented of tokenizers [6, 7], but formulae unlike text, formulae are highly structured which cannot easy to be parsed into terms like text. However, unlike text, formulae can be expressed and parsed as tree structures. Thus, tree-based methods are the most common tokenization approach in recently proposed MIR system [3, 5, 8, and 9].

Indexer: Indexer is utilized to construct index files so as to improve searching efficiency. In traditional information retrieval systems [10], inverted index is widely employed to reduce the time consumption for searching. Based on the novel tokenization methods, MIR systems with the aid of inverted index technique can provide real-time mathematical searching [3, 4, and 5].

Ranker: The capacity of ranker is to sort the retrieved documents that are collected form index files according to the query. Nevertheless, how to rank the retrieved documents is still an open problem for mathematical information retrieved system. It is not easy to rank the documents only by once ranking. Re-ranking the top-k formulae by the Maximum Subtree Similarity is proposed in MIR system [9].

2.2 Learning to Rank

Learning to rank approaches have been successfully applied to information retrieval (IR) tasks. Cao [11] apply Ranking SVM to document retrieval. Neumann [12] uses SVMRank approach to rank query paraphrases in community question answering (cQA). Liu [13] presents an improved ListNet [14] approach to rank figures in a biomedical article. Tan [15] apply the coordinate ascent algorithm to recommend quotes for writing. All these exploratory experiments have achieved significant improvement in ranking results. Therefore, WikiMir tries to apply the learning to rank model to sort the retrieved formulae.

3. The Proposed System

3.1 Overview

WikiMir aims at searching similar mathematical formulae based on the structure, keyword and importance of formulae in a document. In order to improve the ranking performance, a learning to rank model is used in the ranker of WikiMir, and then re-rank the top-1000 results by the regular expressions matching of query formula. The framework of WikiMir is illustrated in Figure 1. It contains two workflows, the green lines denote the offline workflow and the red lines denote the online workflow.

In order to support real-time search, WikiMir builds the indexes and training the learning to rank model in the offline workflow. In the process of building the index files, mathematical information is obtained from the dataset at first. Meanwhile, the importance values of each formula in each document are calculated according to the roles of formulae in the documents and they will be utilized in the online workflow to the ranking of the retrieved documents. Afterwards, both the formula and context terms are extracted by tokenizer from the dataset, indexer stores the features (e.g., tf-idf) of formula, keyword and documents in the index files. In the training process, WikiMir uses the training data to train a ranking model, the training data include features of queries and a list of relevant documents. Meanwhile, WikiMir uses the ranking model to rank the relevant formulae. The procedure of building the training data will be illustrated in details in the next section.

The green lines denote the online workflow, namely searching process. In the process, the query formulae and keywords are parsed into terms by tokenizer. Then, documents containing the terms of the query are collected from the index files that are built in the offline workflow. Then the retrieved documents are sorted by ranker according to features of query formulae and documents. Finally, we automatic generate the regular expressions according the query formula, and use it to match the formulae in the retrieved documents, then re-rank the retrieved documents.

3.2 Interface

In the NTCIR-12 MathIR main tasks, the data of formula is stored in xml files, and the makeups of query formulae are LaTeX and MathML. WikiMir uses the LaTeX format as the presentation of query formulae, so the interface of WikiMir is to identify the LaTeX format of query formulae from the xml files into internal uniform formats, namely Presentation MathML. The interface extracts formulae markups via the pre-defined markup tags, for example" <annotation>", and then the format of formulae are analyzed and outputted as Presentation MathML using the refined formula structure analysis algorithm proposed in [16]. In addition, some of the query formulae contain wildcards which make the searching more difficultly. In our system, if the query formulae include wildcards, we will treat them as a sub-structure of formula and no generalize them again (the generalization process will detail described in the next sections).

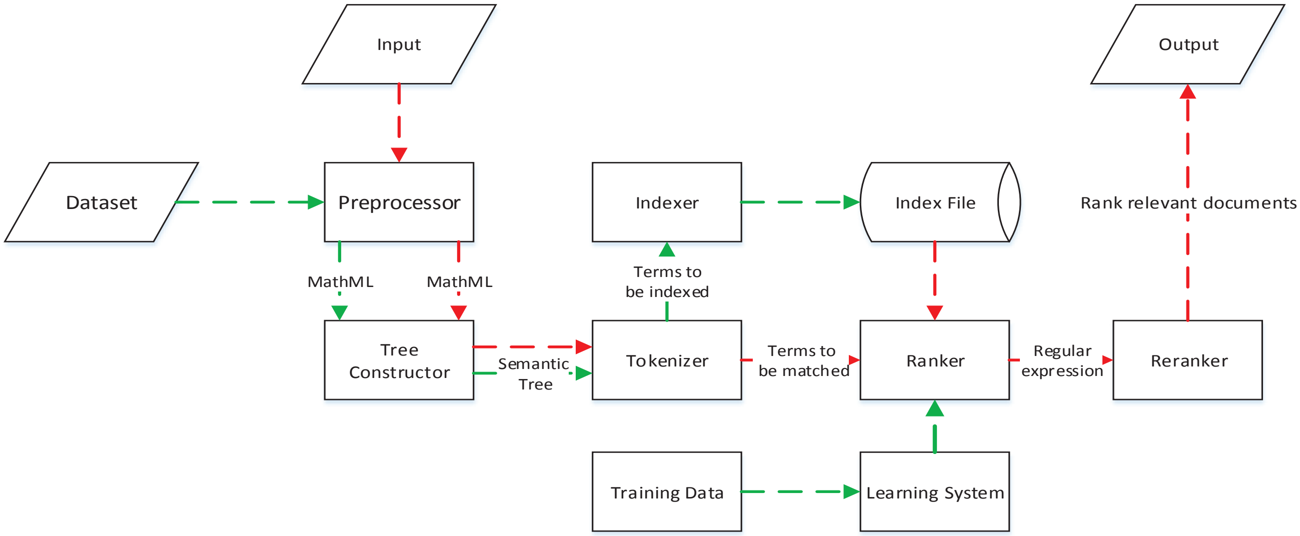
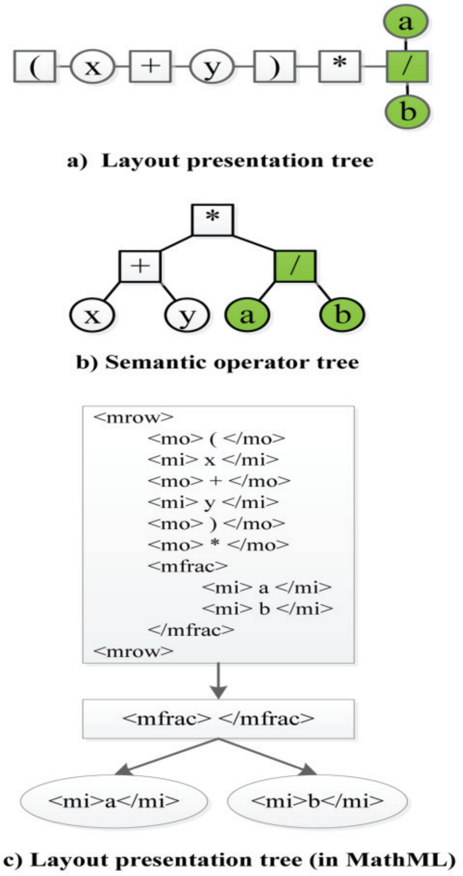


Figure 1. Workflow of WikiMir (green lines denote the offline workflow and red lines denotes the online workflow)

3.3 Tree Constructor

WikiMirs uses the semantic operator tree to represent the formulae. The semantic operator tree is converted from layout presentation by the semantic enrichment technique. The layout presentation tree of formula can easily be extracted directly based on Presentation MathML, See an example in Figure 2 c). But many useful semantic contents of formulae are lost in in layout presentation, due to the symbols in one-dimensional relations are connected using the same horizontal tag (e.g., "<mrow>" tag in Presentation MathML). And the semantic meanings (e.g., operator priority, operands) of one-dimensional relation are unknown in layout presentations. Therefore, the structures and semantics in one-dimensional relations cannot be utilized in mathematics retrieval. WikiMir uses the semantic interpretation technique to convert layout presentation in one-dimensional relation into corresponding semantic presentation.

Takes " $(x+y)\times\frac{a}{b}$ "as an example, its layout and semantic presentation is illustrated in Figure 2 c)-a). The layout presentation of a fraction (in two-dimensional relation) can be converted into its semantic presentation using straightforward tag conversion as denoted in green nodes in Figure 2 c)-a). The semantic operator tree of the example is illustrated in Figure 1 a)-b), the semantic operator tree is converted from the layout presentation through the semantic enrichment technique.



a) Layout presentation tree

b) Semantic operator tree

```
<mrow>
        <mo> ( </mo>
        <mi> x </mi>
        <mo> + </mo>
        <mi> y </mi>
        <mo> ) </mo>
        <mo> * </mo>
        <mfrac>
                <mi> a </mi>
                <mi> b </mi>
        </mfrac>
<mrow>
```

c) Layout presentation tree (in MathML)

Figure 2. Layout presentation and semantic presentation of $(x+y)\times\frac{a}{b}$

3.4 Tokenizer

In the main tasks of NTCIR-12, the queries are comprised of formulae and keywords. Therefore, there are two tokenizers in WikiMir, one is the formula tokenizer, the other is the context tokenizer.

3.4.1 Formula Tokenizer

The formula tokenizer contains two parts, normalization and term extraction. In the normalization process, different formulae with a same meaning are converted into a uniform formula so as to ensure the high recall of relevant formulae. Variables, constants and the order of operands are all taken into account in the process. Term extraction aims at extracting the index terms from the semi-operator tree of formula. In order to mimic the formula understanding process of users, a term extraction method with generalization is proposed towards the semantic operator tree of formulae. WikiMir extracts two kinds of terms, original terms and generalized terms. The original terms are defined as the substructures of semantic tree presentation formula. The generalized terms are proposed by changing the variables and constants of the original terms into wildcards. For example, the original terms and generalization terms of $(x+3)\times\frac{a}{b}$ are shown in Table 1.

Table 1. Terms of $(x+3)\times\frac{a}{b}$

Level	Original terms	Generalized Terms
1	$(x+3)\times\frac{a}{b}$	$(*)\times *$
2	$(x+3)$	$(*)$
3	$\frac{a}{b}$	$\frac{*}{*}$
4	$x+3$	$*+*$

3.4.2 Context Tokenizer

The queries of NTCIR-12 MathIR main tasks include keyword and formulae. Therefore, WikiMir builds a context tokenzier. The terms of keywords are extracted similarity as the traditional text searching engines [17].

3.5 Indexer

In order to support real-time online searching, the index files are built in the offline workflow by indexer, in which the inverted index data structure is employed. Two index files are built in the offline workflow. One index file is built for formulae terms, the other is constructed for context terms. In the formula index files, the importance of formulae is also recorded, due to the different formulas usually play different roles in a document, the significant ones should be paid more attention to. All the statistics features of formulae and documents are all recorded, and these will be used to rank the retrieved formulae by the ranker which is trained in the offline workflow.

3.6 Ranker

Ranker is the core module of search engine and it directly determines the ranking results. In WikiMir, the retrieved formulae are sorted by ranking model which is trained in the offline workflow. And we apply RankBoost [18] algorithm to training the ranking model. However, if the number of small sub-terms matching is large, the matching accuracy of main structure of queries formulae and the ranking effective will be affected, because users who search mathematical information often need the pages contain expressions that only approximately match the query formula. So we re-rank the top-1000 results through the regular expressions matching of query formulae that are automatic generated. In the process of regular expressions matching, in order to obtain the regular expressions of formulae, WikiMir replaces all the \quar{} of query formulae to the wildcard(*) , and then WikiMir uses the regular expressions to matching the retrieved formulae and re-rank them.

The training data used in the training process include 224 instances, each of which includes the features of a query and the corresponding top-100 retrieved formulae of the searching results. The searching results are calculated by the structure similarity and the regular expression matching of the query formulae. In addition, the relevance of documents with respect to the query is evaluated by six postgraduates and six undergraduates in different majors, such as mathematics, computer science, etc. The features we investigated can be grouped into three categories, containing 255 dimension features as following. Each feature is normalized by the sum of all its values.

Formula-based Features: The semantic and structural information of query formulae are important features to help the final ranking. Moreover, we get the features of formulae in the corpus, because there may be higher probability for popular formulae to be searched.

Variable & constant: The number of variables and respective constant in the query formula.

Two-dimensional relation: The layer number of the semi-operator tree which is used to represent the structural information of formula. The semi-operator tree of formula is described in literature [2,4].

Frequency of the term: The frequency of the term in the inverted formula file.

Relevance-based Features: The relevance of query formula and the retrieved documents is the similarity between them. In order to evaluate it, we extract terms from the semi-operator tree which is used to represent formula, and then use several metrics to calculate the relevance. The computing methods below are detailed described in literature [2,4].

Frequency of the term occurring in a formula: The frequency of the query formula term occurring in a formula.

Formulae Level Distance: The distance of the matched terms on different levels between the query formulae and the related formulae in documents.

The match ratio between query and the formula: It is the matching score which is calculated by the total number of the terms in the query and a formula.

Independent score: The score which represents the similarity between the query and a single formula.

Comprehensive score: The score which is calculated by the similarity between a query and a document.

Document-based Features: Users who are searching formulae in MIR system may be interested in the articles which not only include the related formulae, but also the related formulae applied in the similar application scenarios. Therefore, features of document are useful to help ranker sort the retrieved documents.

Formula importance value: The importance of formula in a document.

Link number: Number of inlinks and outlinks in the document. We hypothesize that the stronger the association between a document and other documents, the more important of the document. The intensity of association can be reflected from link number.

The ratio of depictive text: The length ratio of the preceding paragraph and the following paragraph of a formula to the total length of the article. In a document, if we use more text to describe a formula, maybe the formula is more important to the document.

Picture number: Number of pictures in the document. We speculate that users tend to read the illustrated documents.

4. Experiment Result

4.1 Dataset and Queries

The datasets are provided by NTCIR-12, and it contains two parts, one is the ArXiv dataset, the other is the Wikipedia dataset.

The size of ArXiv dataset is 173 uncompressed. And there are include 29 queries in ArXiv main task. Each topic includes one or more keyword and one or more formulae at least.

The Wikipedia dataset contains 319689 HTML articles, and it includes 592443 formulae inside <math> tags in these HTML articles, 580068 marked formulae are in MathTagArticles and 12375 marked formulae are in TextArticles. The size of the Wikipedia dataset is 417MB uncompressed. There are 30 queries in Wiki main task, and each topic includes one or more keyword and one or more formulae.

4.2 Accuracy

The main measures for evaluation in ArXiv main task are MAP (Mean average precision over judgment groups), P-5(Precision at rank 5), and P-10 (Precision at rank 10), and P-20(Precision at rank 20). The evaluation results [19] of the two main tasks are shown as follow:

Table 2. ArXiv main task results of ICST

	Relevance Hits>3 (Relevant)	Relevance Hits>=1 (Partially Relevant)
P-5 avg	0.2276	0.5517
P-10 avg	0.1862	0.4966
P-15 avg	0.1632	0.4299
P-20 avg	0.1362	0.4000

Table 3. Wikipedia main task results of ICST by the trec_eval method

	Relevance Hits>3 (Relevant)	Relevance Hits>=1 (Partially Relevant)
P-5 avg	0.4733	0.8533
P-10 avg	0.3767	0.7900
P-15 avg	0.2978	0.7133
P-20 avg	0.2617	0.6600

Table 4. Wikipedia main task results of ICST by the rank_based method

	Relevance Hits>3 (Relevant)	Relevance Hits>=1 (Partially Relevant)
P-5 avg	0.4733	0.8533
P-10 avg	0.3763	0.7900
P-15 avg	0.2978	0.7133
P-20 avg	0.2617	0.6600

From the results above we can see that, WikiMir is effective in ArXiv and Wikipedia main task. From the results we also can find that the ranking performance of Wikipedia is better than ArXiv. The situation is caused by the training data, the training data of WikiMir come from Wikipedia. The training data is important in the process of building learning to rank model.

5. Conclusions

In this paper, we present our system (WikiMir) for NTCIR-12 MathIR tasks. The system takes keyword of formulae into account. A context index is constructed to match the keywords in queries and the keywords in documents. What' more, the importance of formulae in a document is introduced, This system uses the supervised method to rank the retrieved documents. Finally, we re-rank top-k formulae by the regular expressions of query formulae which make the order of the results becomes more reasonable.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61472014), the Natural Science Foundation of Beijing (No.4142023) and the Beijing Nova Program (2015).

6. REFERENCES

[1] Miller, B., and Youssef, A. Technical aspects of the digital library of mathematical functions. In *Annals of Mathematics and Artificial Intelligence*.121-136. 2003.

[2] Wang, Y., Gao, L., Liu, X., Wang, S., Yuan, K., and Tang, Z. WikiMirs 3.0: A Hybrid MIR System Based on the Context, Structure and Importance of Formulae in a Document. In *Proceeding*s of the 15th ACM/IEEE-CS joint conference on Digital libraries. 173-182. ACM. 2015

[3] Hu, X., Gao, L., Lin, X., Tang, Z., Lin, X., and Baker, J. B. Wikimirs: a mathematical information retrieval system for wikipedia. In *Proceedings of the 15th ACM/IEEE-CS joint conference on Digital libraries(JCDL).* 11-20.ACM. 2013.

[4] Lin, X., Gao, L., Hu, X., Tang, Z., Xiao, Y., and Liu, X. A mathematics retrieval system for formulae in layout presentation. In *Proceedings of Annual International ACM SIGIR Conference on Research & Development in Information Retrieval.* 697-706. ACM, 2014.

[5] Stalnaker, D., and Zanibbi, R. Math expression retrieval using an inverted index over symbol pairs. IS&T/SPIE Electronic Imaging International Society for Optics and Photonics. 2015.

[6] B. R. Miller and A. Youssef. Technical aspects of the digital library of mathematical functions. Annals of Mathematics and Artificial Intelligence, 38(1-3):121-136, 2003

[7] R. Miner and R. Munavalli. An approach to mathematical search through query formulation and data normalization. In Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008, pages 55-67.2008.

[8] P. Sojka and M. Liska. Indexing and searching mathematics in digital libraries. In Intelligent Computer Mathematics, pages 228-243. Springer, 2011.

[9] Zanibbi R, Davila K, Kane A, et al. The Tangent Search Engine: Improved Similarity Metrics and Scalability for Math Formula Search[J]. arXiv preprint arXiv:1507.06235, 2015.

[10] A. Singhal. Modern information retrieval: A brief overview. IEEE Data Eng. Bull., 24(4):35-43,2001.

[11] Cao Y., Xu J., Liu T., Li H., Huang Y., and Hon H. Adapting Ranking SVM to Document Retrieval. In *Proceedings of Annual International ACM SIGIR Conference on Research&Development in Information Retrieval.pages:*186-193. ACM, 2006.

[12] Neumann, G., Figueroa, A. Learning to Rank Effective Paraphrases from Query Logs for Community Question Answering. *In 27th AAAI Conference on Artificial Intelligence.*2013.

[13] Liu, F.; and Yu, H. Learning to Rank Figures within a Biomedical Article. *Plos One* 9(3): e61567. 2014.

[14] Cao Z, Qin T, Liu TY, Tsai MF, Li H. Learning to rank: from pairwise approach to listwise approach. Proceedings of the 24th international conference on Machine learning. Pages: 129–136. 2007.

[15] Tan, J., Wan, X., and Xiao, J. Learning to Recommend Quotes for Writing. In *29th AAAI Conference on Artificial Intelligence.*2453-2459. 2015.

[16] R. Zanibbi and D. Blostein , and J. R. Cordy. Recognizing mathematical expressions using tree transformation. IEEE Trans. On Pattern Analysis and Machine Intelligence, 24(11): 1455-1467, 2002.

[17] C. D. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval, volume 1, Cambridge university press Cambridge,2008.

[18] Freund, Y., Iyer, R., Schapire, R., E., et al. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research* 4: 933-969. 2003.

[19] Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, el al. NTCIR-12 MathIR Task Overview, NTCIR, National Institute of Informatics (NII), 2016.