ICST Math Retrieval System for NTCIR-11 Math-2 Task

Liangcai Gao
Peking University
glc@pku.edu.cn

Yuehan Wang
Peking University
wangyuehan@pku.edu.cn

Leipeng Hao
Peking University
haoleipeng@pku.edu.cn

Zhi Tang
Peking University
tangzhi@pku.edu.cn

ABSTRACT

In NTCIR-11, the NTCIR-Math-2 Task is organized for mathematical information retrieval. This paper proposes an innovative system for efficient formula index and retrieval. We build a novel indexing and matching model, taking both textual and spatial similarities into consideration. Besides, a hierarchical technique is introduced to generate sub-trees from the semi-operator trees of formulae. The experimental results demonstrate that the method of our system is effective and promising in practical application.

Team Name

ICST

Subtasks

Math-2 Main Task (English)

Keywords

Mathematical Information Retrieval, Structure Matching, Layout Presentation

1. INTRODUCTION

As recently as the 1990s, studies showed that most people preferred getting information from people rather from information retrieval (IR) systems. However, during the last decade, relentless optimization of information retrieval effectiveness has driven search engines to new quality levels at which most people are satisfied. At the same time, different kinds of information retrieval systems are more and more popular, for instance, formulae information retrieval system.

Mathematical formulae are frequently used in various fields, such as Science, Technology, Engineering and Mathematics. Mathematics retrieval, aiming at facilitating the access, retrieval and discovery of math resources, is increasingly demanded in many scenarios. For example, many traditional courses and Massive Open Online Courses (MOOCs) release their resources (books, lecture notes and exercises, *etc*) as digital files in HTML or XML. In order to display mathematical concepts with high quality, mathematical formulae are usually made up in layout presentation (Presentation MathML, LaTeX, *etc*). However, due to the specific characteristics of formula, classic search engines do not work well for indexing and retrieving. The main challenges are query interface, normalization, indexing and ranking [1, 11].

Unlike text-based search, mathematics retrieval takes formulae as queries. Most current engines need users to knew the classification, name or string encoding of a formula. Normalization is a necessity to insure that equivalent math formulae in different presentations or transformations can be recalled. Formula normalization is required to unitize variation in variables, constants, spatial layouts or even semantics among equivalent formulae. For instance, "$x \times y - z$" and "$-a + b \times c$", "a^{-2}" and "$\frac{1}{a^2}$", pairs of them have the same meaning with different layout presentations. For accurately calculation of the structural similarities of formulae, subtree is taken into consideration. For example, "$x \times y$" is a sub-tree of "$x \times y + z$". Nevertheless, the method to figure the similarity according to the attributes of tree structures remains an open problem, cause the motivations of different users according to their background and personal task [12].

In this study, we proposed a mathematics retrieval system, focusing on the challenges mentioned above. The formula format is Presention MathML in our system. For normalization, a semantic enrichment technique is introduced to extract the structural and semantic information from layout presentations of formulae. Meanwhile, we also propose a novel similarity function, based on hierarchy and frequency, to calculate the similarity score of formulae.

2. RELATED WORK

Mathematics retrieval has been researched since 2003 [5], and more than ten systems are developed. We analyze the relevant approaches on formula presentation, indexing and ranking techniques as follows.

Presentation is important since it denotes the formula internal format of a mathematics retrieval system and also determines the compatibility of a mathematics retrieval system to the existing data sources. Some mathematics retrieval researches focus on semantic presentations [4], which mark up the semantic meanings (e.g., Content MathML, Open Math) of formulae. However, in practice, most math formula resources are presented in layout presentations. Since Presentation MathML and LaTeX only contain limited spatial layouts and few semantic information of formulae, existing mathematics retrieval systems based on these layout presentations can only support exact matching [7] or barely consider the structure matching [2, 8, 9].

The main indexing techniques of mathematics retrieval include text-based and tree-based. Text-based indexing techniques [5, 6] is to convert math formula markups into plain text strings, so that they can be indexed using existing text-based indexing tools like Lucene. B. Miller, A. Youssef con-

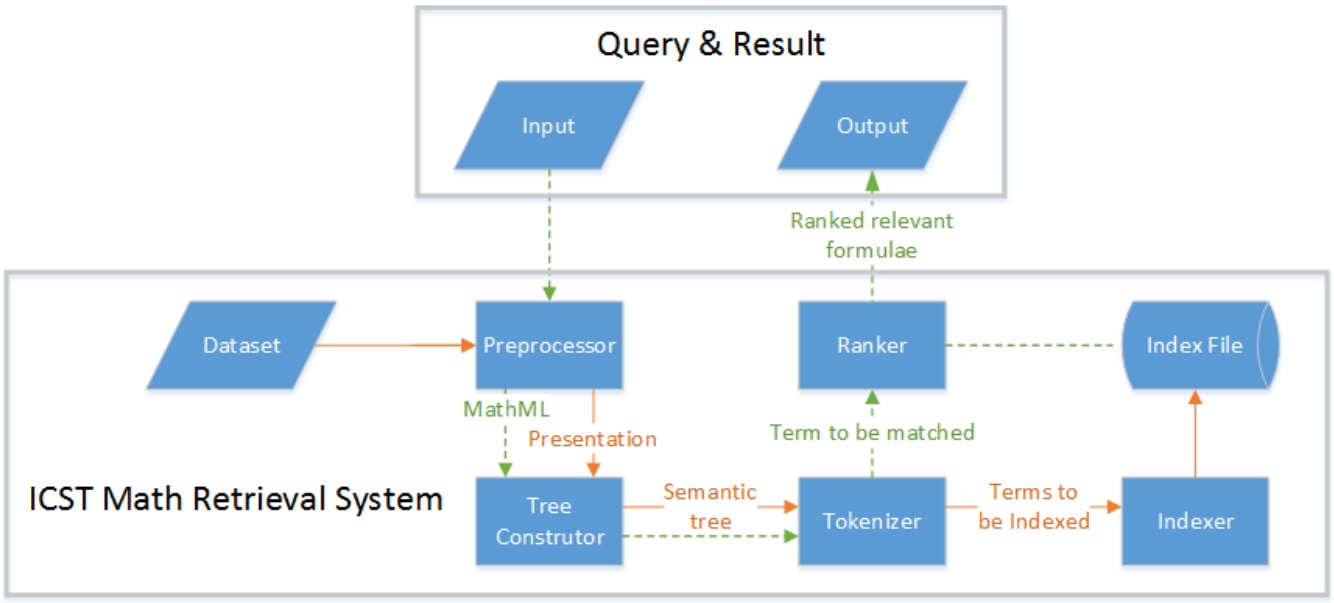


Figure 1: Workflow of the proposed system (Dotted lines denote online query flows and solid lines denote offline index flows).

vert all non-alphanumeric symbols in LaTeX into alphanumeric symbols and normalize the order of operands into a canonical form [5]. The structures of formulae are lost in those methods. In tree-based methods, attributes of formula tree structures are extracted as index terms. A straightforward way is to index all substructures of formulae with term attributes [10]. X. Hu L. Gao *etc* [2] propose to extract substructures from the LaTeX markups considering hierarchical generalization of substructures. In order to avoid explosive growth of index space when indexing all substructures of formulae, M. Kohlhase and I. Sucan [4] apply a substitution tree indexing technique to index substructures of semantic formula presentation. However, the substitution tree in [4] is built upon operator tree, which is difficult to extract from formulae in layout presentations. To overcome this, T.Schellenberg, B. Yuan *etc* [9] employ the substitution tree indexing technique to index layout presentations of formulae. Other indexing methods, which are not text-based or tree-based, have also been proposed [8].

As for ranking, most text-based methods use $tf - idf$ to calculate similarities of formulae. R. Miner and R. Munavalli [6] introduce weights for terms according to their levels, lengths and complexities. Some tree-based methods also use the modified $tf - idf$ to calculate the matching scores of substructures. P. Sojka and M. Lska [10] introduce weights to discriminate substructure matches in different levels based on the assumption that structures in higher level are more important than those at lower levels. Hu et al. [2] introduce weights based upon the distance of matched terms in query and the matched formulae. Besides $tf - idf$, some methods evaluate the similarities of formulae according to the similarity of feature sets. T. T. Nguyen S. C. Hui *etc* [8] index formulae in the mathematical concept lattice structure based on similarities of feature sets of each formula. In the literature [9], each formula can be presented by a set of sub-expressions with their attributes presented in a 5-tuple. S. Kamali and F. W. Tompa [3] calculate the formula similarity using tree edit distance, which cannot evaluate the structure similarity in different levels.

3. THE PROPOSED SYSTEM

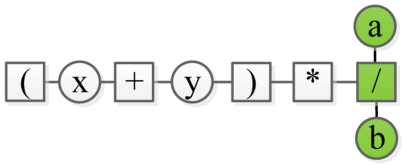
3.1 Overview

Our system aims at searching similar mathematical formulae based on both textual and spatial. The system consists of preprocessor, tree constructor, indexer and ranker, as shown in Figure 1. The system is supported by Lucene. The two frames present Query & Result and ICST Math Retrieval System respectively. The solid lines show the workflow of dataset establishment. Firstly, preprocessor extracts uniform internal format (Presentation MathML) from HTML5 data sources. Then, semantics of formulae in Presentation MathML are enriched and semantic operator trees are constructed by tree constructor. Terms are converted by tokenizer with normalization and generalization. Finally, the indexer calculates and stores the statistical data of each term in the inverted index files. The dotted lines denote the workflow how a query searches formulae. The query is preprocessed into Presentation Math ML and enriched with semantics information. Next terms are got by tokenizer and passed through ranker to find the matched terms in the index files. Meanwhile, The ranked formulae scores are calculated. Lastly, a list of ranked formulae are returned to the user.

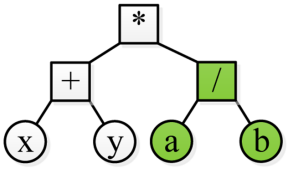
3.2 Preprocessor

In NTRIC-11 Math-2 Task, the data of formula is stored in HTML5 and XHTML5 files. The goal of the preprocessor is to identify formulae markups from the HTML files into internal uniform formats, namely Presentation MathML. The preprocessor extracts formula markups via identifying the

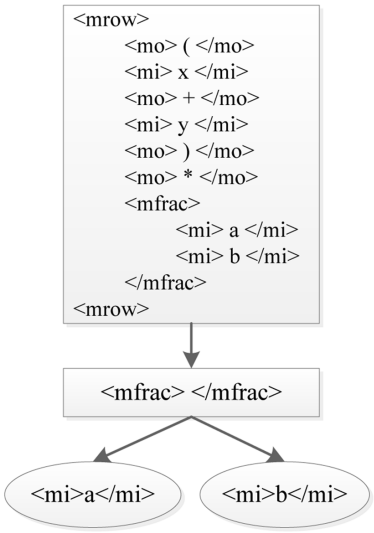
pre-defined markup tags, for example "<math>". For instance, formulae are presented as different format in the given HTML file, such as MathML and LaTeX. The dataset and input queries of our system are based on the format of Presentation MathML, so the queries with LaTeX format can be ignored.



a) Layout presentation tree



b) Semantic operator tree



c) Layout presentation tree (in MathML)

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

3.3 Tree Constructor

Conversion from layout presentation to semantic presentation has been widely researched in the area of formula understanding [11], but satisfactory methods or conversion tools have not been proposed yet. Nevertheless, this limitation is overcome by a semantic enrichment technique in our system. A layout presentation tree can be extracted directly from Presentation MathML. Take " $(x+y) \times \frac{a}{b}$" as an example, its layout and semantic presentation are illustrated in Figure 2 a)-b). The layout presentation of a fraction can be converted into its semantic presentation by using a straightforward tag conversion, as denoted in green nodes in Figure 2 a)-b). In the system, layout presentation is converted into a corresponding semantic presentation, which is called semantic operator tree here by using the semantic enrichment technique.

3.4 Tokenizer

The tokenizer workflow includes two parts, normalization and term extraction. The normalization converts different formulae which have the same meaning into one uniform format, so as to ensure the high recall of relevant formulae. The objects of normalization generally comprise variables, constants and the order of operands. Index terms are extracted from the formula by term extracting processing. It is well known that generalization is an important behavior when people understand formulae. Additionally, generalization of substructures in low level can avoid overrating for the substructures, which is a common problem occurring in tree-based indexing methods. 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. For example, " $(x + y) \times \frac{a}{b}$" can be tokenized into terms, as shown in Table 1.

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

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

3.5 Indexer and Ranker

In order to make the retrieval more efficient, the system calculates the statistics of each term and stores them in the inverted index files. Index files for terms and formulae are built to calculate the score, which denotes the similarity between query and formulae. Meanwhile, the score determines the rank of the result. It is defined as follow,

$$score(Q, F) = W_{cover}(Q, F) \times (\sum_{t \epsilon Q} (tf_f(t, F) \times iff^2(t) \times W_{level}(t, Q, F) \times W_{gen}(t)) \quad (1)$$

where Q and F represent a query and a formula. t is a term of Q. $W_{cover}(Q, F)$ denotes the ratio of number of terms in Q matched by F. $tf_f(t, F)$ and $iff^2(t)$ are term frequency and inverse formula frequency. $W_{level}(t, Q, F)$, which is mentioned in Table 1, is introduced to evaluate the distance of the matched terms on different levels. $W_{gen}(t)$ is a penalty of the generalized terms. It is assigned as 1 if t is an ordinary term, otherwise 0.5.

4. EXPERIMENT RESULT

4.1 Dataset and Queries

The dataset, provided by NTCIR, is documents resampled from the same ArXiv dataset. It consists of 105,120

scientific papers. The papers are divided automatically into total 8.301,578 search unit. The documents are converted into HTML5 and XHTML5 formats by the KWARC project (http://kwarc.info/). The size of the dataset is 173G uncompressed.

In the task, there are 50 queries. Each topic include one keyword and one formula at least.

4.2 Accuracy

The main measures for evaluation in this task are MAP (Mean average precision over judgment groups), P-5 (Precision at rank 5), P-10 (Precision at rank 10). The result of ours is show in Table 2 as follow.

Table 2: Result of ICST

	Relevance Level >= 3 (Relevant)	Relevance Level >= 1 (Partially Relevant)
Map avg	0.0700	0.0207
P-5 avg	0.0180	0.0760
P-10 avg	0.0360	0.1520

5. CONCLUSIONS

In this paper, we present out system to NTCIR-11 Math-2 Task. To Facilitate the access and search for mathematical formulae, a mathematics retrieval system focusing on such formulae is proposed, with the following contributions: 1) A semantic enrichment technique is introduced to extract useful semantic information from formulae in layout presentation. 2) Hierarchical generalization of substructures is proposed to generate index terms to support substructure matching and fuzzy matching. Due to the time limitation, our system only dealt with a few files of all so that the performance of it is not good. In the future, we will improve their algorithm and handle all files.

6. ACKOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61202232).

7. REFERENCES

[1] A. Aula and M. Käki. Understanding expert search strategies for designing user-friendly search interfaces. In *ICWI*, pages 759–762. Citeseer, 2003.

[2] X. Hu, L. Gao, X. Lin, Z. Tang, X. Lin, and J. B. Baker. Wikimirs: a mathematical information retrieval system for wikipedia. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 11–20. ACM, 2013.

[3] S. Kamali and F. W. Tompa. Retrieving documents with mathematical content. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 353–362. ACM, 2013.

[4] M. Kohlhase and I. Sucan. A search engine for mathematical formulae. In *Artificial Intelligence and Symbolic Computation*, pages 241–253. Springer, 2006.

[5] 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.

[6] R. Miner and R. Munavalli. An approach to mathematical search through query formulation and data normalization. In *Towards Mechanized Mathematical Assistants*, pages 342–355. Springer, 2007.

[7] J. Mišutka and L. Galamboš. Extending full text search engine for mathematical content. *Towards Digital Mathematics Library. Birmingham, United Kingdom, July 27th, 2008*, pages 55–67, 2008.

[8] T. T. Nguyen, K. Chang, and S. C. Hui. A math-aware search engine for math question answering system. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 724–733. ACM, 2012.

[9] T. Schellenberg, B. Yuan, and R. Zanibbi. Layout-based substitution tree indexing and retrieval for mathematical expressions. In *IS&T/SPIE Electronic Imaging*, pages 82970I–82970I. International Society for Optics and Photonics, 2012.

[10] P. Sojka and M. Líška. Indexing and searching mathematics in digital libraries. In *Intelligent Computer Mathematics*, pages 228–243. Springer, 2011.

[11] R. Zanibbi and D. Blostein. Recognition and retrieval of mathematical expressions. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(4):331–357, 2012.

[12] J. Zhao, M.-Y. Kan, and Y. L. Theng. Math information retrieval: user requirements and prototype implementation. In *Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries*, pages 187–196. ACM, 2008.