Mathematical Equation Retrieval Using Plain Words as a Query

Shinil Kim, Seon yang and Youngjoong Ko
Department of Computer Engineering, Dong-A University
840, Hadan 2-dong, Saha-gu, Busan, 604-714, Korea
{pirate2003, seony.yang}@gmail.com, yjko@dau.ac.kr

ABSTRACT

This paper proposes how to effectively retrieve the mathematical equations when the plain words are given as a query. The proposed system requires no complicated mathematical symbols, no particular input tool and no constraint of query. Users can enter a query with plain words like the traditional Information Retrieval. For this, we extract features from the plain texts that are converted from the real math equations. Experimental results show an outstanding performance, a MRR of 0.6585.

Categories and Subject Descriptors

H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval – *query formulation, retrieval models.*

Keywords

Mathematical equation retrieval, MathML, Identifier & Number, Operator & Structure

1. INTRODUCTION

Information Retrieval (IR) is already popularized through PCs, phones, tablets, etc. Its application areas are recently becoming wider and wider; beyond texts, researchers target voices, images, etc. However, the study of IR for mathematical equations is still in infancy.

One of the initiatives to promote the accessible publication of mathematical contents is MathML (Mathematical Markup Language. http://www.w3.org/Math/), which is focused in this study. MathML helps web documents to easily include mathematical expressions without images. By this reason, the number of documents containing MathML expressions is rapidly increasing.

In this paper, we propose how to easily retrieve math expressions written in MathML. Note that most people do not know much about MathML. If only those who know MathML are allowed to retrieve MathML equations, most people cannot even have the opportunity of equation retrieval. Although there are some math equation input tools available, most users are unfamiliar with the tools. In addition, it is not easy to enter the complicated math symbols in the search box. If there is an IR system that can retrieve math equations with queries with plain words, it could be very useful in many areas, e.g., the field of web-based education. Therefore, the main difference between previous studies and the present one is that the proposed system allows users to enter queries using plain words. To our knowledge, there is not a reported study on this problem so far. For example, we assume that a user is searching for a certain formula. He or she only remembers some part of the formula.

· Target formula: $x = \frac{-b\pm\sqrt{b^2-4ac}}{2a}$

· Part that the user remembers: $\sqrt{b^2-4ac}$

· Example query: "루트 b 의 제곱 마이너스 4ac" (root b squared minus 4ac)

As shown above, users can simply create a query *as if they speak the math equation*.

There are two possible approaches for our purpose: 1) converting queries into MathML expressions, and 2) converting MathML data into plain texts. The former is relatively simple but it is seriously dependent on the query translation performance. The latter requires more work but can alleviate the requirement of high query conversion performance. In addition, since the target consists of plain words, it can be easily combined to many traditional IR techniques. It is the reason why this paper focuses on the latter.

Since features are first extracted from the plain texts converted from real MathML equations, it is needed to convert the math equations into plain sentences. Technically, a "plain sentence" here means "a sequence of normal words" (hereafter, math-sentence). For example, $\sum_{k=1}^{n} k^2$ is converted into "시그마 k 는 1 에서 n 까지 k 의 제곱" (sigma k from 1 to n k squared). Through this conversion process, we construct a wide range of mathematics vocabulary, e.g., a math operator lexicon, synonyms and expressions. Various factors such as identifiers, operators and their order are employed as features and then we index and rank math-sentences of math equations using the features. This method shows a Mean Reciprocal Rank (MRR) of 0.5080. Moreover, we conducted additional experiments by combining the method with the traditional IR techniques. Finally, we achieved a higher performance, a MRR of 0.6585.

The rest of the paper is organized as follows: Section 2 briefly describes the related work. Section 3 describes our proposed system in detail. Section 4 presents experimental results. Finally, we discuss conclusions and future work in section 5.

2. RELATED WORK

All After the release of MathML, researchers have started to study math equation retrieval in earnest. Youssef (2006) presented the roles of math search. Miner and Munavalli (2007) described an approach to search for mathematical notation. Adeel et al. (2008) aimed to create a search system enable users to search for mathematical formula contents. Misutka and Galambos (2008) studied to search for a mathematical formula in real-world

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10...$15.00.

mathematical documents, but still offering an extensible level of mathematical awareness. Zhao et al. (2008) first review the current approaches and resources for math retrieval, then report on the interviews of a small group of potential users to properly ascertain their needs. Yokoi and Aizawa (2009) proposed a similarity search method for mathematical equations that are particularly adapted to the tree structures expressed by MathML. Shin and Kim (2010) studied to retrieve mathematical documents using a query written in MathML.

Our work is also related to the research of Ferreira and Freitas (2005). Their goal is to convert MathML expressions into representations of audio version in English and Portuguese. They reviewed the problem of speaking mathematics and presented the tool AudioMath[1]. MathPlayer[2] also converts MathML expressions into representations of English audio version. Based on these previous studies, this study proposes an equation retrieval system that allows plain words as a query.

3. PROPOSED METHOD

3.1 Converting equations into math-sentences

We first convert the math equations, which we collected, into math-sentences. Referring to AudioMath and MathPlayer, we carefully implemented a conversion system (hereafter, MConv-sys). While doing this, we could build some important lexicons. Among them, we briefly introduce the identifier lexicon and the operator lexicon.

- **Identifier lexicon:** An identifier is expressed with the tag <mi> in MathML. It includes all variables and some promised symbols. For example, "sinθ" is represented as "<mi>sin</mi> <mi>&theta;</mi>" in MathML. Table 1 lists some identifier examples.

Table 1. Identifier Examples

Identifier	in MathML	in MConv-sys
a	<mi>a</mi>	"a"
x	<mi>x</mi>	"x"
sine	<mi>sin</mi>	"사인 [*sa-in*]"
log	<mi>log</mi>	"로그 [*ro-geu*]"

- **Operator lexicon:** An operator is expressed with the tag <mo>. Table 2 lists some examples.

Table 2. Operator Examples

Operator	in MathML	in MConv-sys
+	<mo>+</mo>	"플러스 [*peul-reo-seu*]"
=	<mo>=</mo>	"이퀄 [*i-kwol*]"
∫	<mo>&#x222B;</mo>	"인테그랄 [*in-te-geu-ral*]"
∩	<mo>&cap;</mo>	"교집합 [*gyo-jip-hap*]"

We then analyze the rules of reading math equations. The order of reading equations can differ from that of MathML expression. For example, a fraction "one-third" is expressed as "<mfrac><mn>**1**</mn><mn>**3**</mn></mfrac>" whereas it is read by "**3 분의 1**" in Korean; the denominator 3 precedes the numerator 1 in Korean reading. We observed a variety of Korean representations of math expressions. As a result, we could normalize the format of reading and finally implement MConv-sys. We could construct a math-sentence corpus by MConv-sys successfully. An example math-sentence is as follow:

· Example equation:

$$\int e^{\frac{i}{2}\int_0^t(p(s)dq(s)-q(s)dp(2))-i\int_0^t h(p(s),q(s))ds}\ dW_v(p,q)$$

- MathML expression:

```
<math xmlns="http://www.w3. ...">
  <semantics> <mstyle displaystyle='true'>
    <mrow><mo>&#x222B;</mo>
      <mrow><msup><mi>e</mi><mrow>
        <mfrac ><mi>i</mi><mn>2</mn></mfrac>
        <mstyle displaystyle='true'>
          <mrow><msubsup><mo>&#x222B;</mo>
               <mn>0</mn><mi>t</mi></msubsup>
           <mrow><mo stretchy='false'>(</mo>
               <mi>p</mi>
             <mo stretchy='false'>(</mo>
               <mi>s</mi>
          ...
```

· Math-sentence:

"인테그랄 e 위첨자 2 분의 i 인테그랄 0 부터 t 까지 괄호열고 p 괄호열고 s ..." (integral e superscript i by 2 integral from 0 to t open parenthesis p open parenthesis s ...)

3.2 Feature Definition

The Two types of features, F1 and F2, are extracted from the math-sentence corpus. Table 3 shows the steps of feature extraction.

F1 (Patterns of Identifier & Number): We first extract identifiers (i) and numbers (n) from each math-sentence. We observe an interesting fact; the order that these two factors appear in math-sentences is relatively consistent. This is the reason why we decide to configure patterns with identifiers and numbers.

F2 (Operator & Structure): The rest words are also used importantly. Except a few stop words, only the operators and some other important equation structure are left. We define them as F2.

Table 3. Features from an example math-sentence

Math-sentence ($D = b^2 – 4ac$)	"D 이퀄 b 의 제곱 마이너스 4ac" (*D equal b squared minus 4ac*)
Identifiers and numbers (with their relative positions)	$D_{(0)}$, $b_{(1)}$, $4_{(2)}$, $a_{(3)}$, $c_{(4)}$
F1 (length=5)	***"iinii"***
The rest (with their relative positions)	'이퀄' in front of $b_{(1)}$, '의 제곱 마이너스' in front of $4_{(2)}$
A stop word	'의 (of)'
F2 (length=3)	***이퀄$_{(1)}$, 제곱$_{(2)}$, 마이너스$_{(2)}$*** ***(equal$_{(1)}$, squared$_{(2)}$, minus$_{(2)}$)***

[1] http://lpf-esi.fe.up.pt/~audiomath

[2] http://www.dessci.com/en/products/mathplayer

3.3 Equation Retrieval

Between the two types of features, we conduct an indexing process only using F1, and ranking process using both F1 and F2. We first extract the math-sentences whose F1 contain the query's F1 as candidates. Then, we calculate the score of each candidate as given in Equation (1):

$$\frac{matched(F1)}{length(candidate\_F1)} + \frac{matched(F2)}{length(candidate\_F2)} \qquad (1)$$

For example, suppose that the equation in Table 3 is an indexed math-sentence and a user enter "x 의 제곱 4bc" ("x squared 4bc") in the search box; F1 and F2 of the query are "*inii*" and "제곱 (squared)", respectively. Since F1 of the math-sentence ("*iinii*") contains that of the query ("*inii*"), it is extracted as a candidate; the length of F1 is 5 and that of F2 is 3 as given in Table 3. In addition, since two elements (*4, c*) of F1 and one element ("제곱 (squared)") of F2 are exactly matched in both value and position, the number of matched element of F1 is 2 and that of matched element of F2 is 1. Thus, the score of the candidate becomes 2/5 + 1/3 = 11/15. In this way, we rank all the candidates by their scores..

4. EXPERIMENTS

4.1 Dataset

We collected a total of 1,800 math equations from the various mathematical manuals that have a high school level or more. After we changed the equations into MathML expressions[3], MConv-sys converted the MathML expressions into math-sentences.

For evaluation, we employed MRR that is a well-known measure in IR. Total 200 equations were given to ten testers at random. The equations played the role of testers' target equations, and the testers created queries and searched the targets by the queries.

4.2 Experimental Results

We first conducted experiments in three different ways. The results are listed in Table 4[4].

Table 4. Results of the three methods

Method	MRR
TFIDF	0.3293
Okapi-BM25	0.3631
F1&F2	**0.5083**

As shown above, F1&F2, the method using F1 and F2, shows better performance than TFIDF or Okapi-BM25. We found that the performances of TFIDF and Okapi-BM25 are very dependent on operators, while one of F1&F2 is relatively dependent on identifiers and numbers.

To compensate for these biased phenomena, we did experiments of linear combination as given in Equation (2):

$$\alpha \cdot Score_{F1\&F2} + (1-\alpha) \cdot Score_{other} \qquad (2)$$

Here, $Score_{F1\&F2}$ denotes the score calculated by F1 and F2, and $Score_{other}$ denotes that generated by TFIDF or Okapi-BM25. Figure 1 and Table 5 show the results.

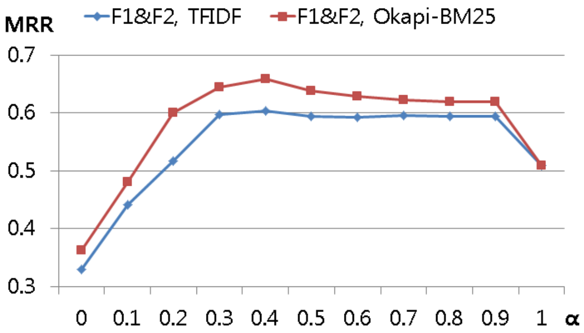

Figure 1. Performance comparison according to α

Table 5. Performances of the combined system; α value is in parentheses.

Method	MRR
F1&F2 (α = 1)	0.5083
F1&F2, TFIDF (α = 0.4)	0.6039
F1&F2, Okapi-BM25 (α = 0.4)	**0.6585**

As can be seen, the overall performance was increased when we combined the methods. In particular, α option of 0.4 showed the best performance. Finally, we achieved an outstanding performance, a MRR of 0.6585, by combining F1&F2 with Okapi-BM25. It could be an important evidence that the usage of the traditional IR techniques by combining with the proposed method are effective on equation retrieval.

5. CONCLUSIONS

This paper has presented how to retrieve math expressions written in MathML using plain words as a query. We first converted MathML equations into math-sentences, and then extracted features from the math-sentences. Our experimental results showed that the proposed method can be effectively used for equation retrieval.

In our future work, we have the following plans. Our first plan is to build a larger corpus. We have been already collecting a considerable amount of equations. This new corpus contains many of the more complex equations. Our second plan is to conduct more experiments for obtaining better performance. One of them is to combine our method (F1&F2) to the state-of-the-art IR techniques instead of TFIDF or Okapi-BM25. The third one is to convert queries into MathML expressions and retrieve MathML equations directly. We will compare its performance with the present one. Our final plan is to apply our method to English and other languages.

6. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded

[3] We used a tool of MathType (http://www.dessci.com/en/products/mathtype).

[4] Since this is the first study that uses plain words as a query, we could not find any comparable previous experiments.

by the Ministry of Education, Science and Technology (2011-0003780).

7. REFERENCES

[1] Adeel, M., Cheung, H. S., and Khiyal, S. H. 2008. Math GO! Prototype of A Content Based Mathematical Formula Search Engine. *Journal of Theoretical and Applied Information Technology*. 4(10):1002-1012.

[2] Ferreira, H. and Freitas, D. 2005. AudioMath: Towards Automatic Readings of Mathematical Expressions. In *Proceedings of Human-Computer Interaction International (HCII 2005)*

[3] Miner, R. and Munavalli, R. 2007. An Approach to Mathematical Search Through Query Formulation and Data Normalization. In *Proceedings of Towards Mechanized Mathematical Assistants (MKM 2007)*. 342-355.

[4] Misutka, J. and Galambos, L. 2008. Extending Full Text Search Engine for Mathematical Content. In *Proceedings of Towards Digital Mathematics Library: DML 2008 workshop*. 55-67.

[5] Shin, J. and Kim, H. 2010. An Equation Retrieval System Based on Weighted Sum of Heterogenous Indexing Terms. *Journal of Korean Institute of Information Scientists and Engineers (KIISE): Software and Applications*. 27(10):745-750.

[6] Yokoi, K. and Aizawa, A. 2009. An Approach to Similarity Search for Mathematical Expressions using MathML. In *Proceedings of Towards Digital Mathematics Library (DML 2009)*. 27-35.

[7] Youssef, A. 2006. Roles of Math Search in Mathematics. In *Proceedings of International Conference on Mathematical Knowledge Management (MKM 2006)*. 2-16.

[8] Zhao, J., Kan, M., and Theng, Y. 2008. Math Information Retrieval: User Requirements and Prototype Implementation. In *Proceedings of Joint Conference on Digital Libraries (JCDL 2008)*