Keisuke Yokoi; Akiko Aizawa

An Approach to Similarity Search for Mathematical Expressions using MathML

In: Petr Sojka (ed.): Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009. Masaryk University Press, Brno, 2009. pp. 27--35.

Persistent URL: http://dml.cz/dmlcz/702557

An Approach to Similarity Search for Mathematical Expressions using MathML

Keisuke Yokoi[1] and Akiko Aizawa[2]

[1] Department of Computer Science, University of Tokyo
kei-yoko@nii.ac.jp
[2] Department of Computer Science, University of Tokyo
National Institute of Informatics
aizawa@nii.ac.jp

Abstract. The recent global computerization and digitization trend has helped to increase the numbers of documents with mathematical expressions on the Web. These mathematical expressions have their own unique structures, and therefore, it is not an easy task for traditional search systems targeting natural languages to deal with them. We propose a similarity search method for mathematical equations that is particularly adapted to the tree structures expressed by MathML based on this background. The similarity search system helps users acquire additional knowledge, discover concealed relationships to different fields, and compensate for some false recognition. Given an equation as a query, most of the conventional mathematical search systems return corresponding equations that exactly match the query. Contrarily, our proposed system makes it possible to return similar equations by measuring the similarity using tree-matching techniques and also by reforming the structure of Content-based MathML. In this paper, we examine our proposed techniques through preliminary experimentation using a prototype search system, and show this techniques' effectiveness based on some conditions requested by the user.

1 Introduction

Search systems play an essential role in our daily lives because there are a huge number of pages on the Web. However, we sometimes encounter problems with how to search for or what to submit as a query when we want to search for things that cannot be easily expressed in natural language. Among them are mathematical expressions that have unique and complicated tree structures. Therefore, a new search scheme is required that takes the structures of mathematical expressions into account.

Many conventional studies on mathematical expression searching are based on the Mathematical Markup Language (MathML), a worldwide standard defined for describing mathematical contents on the Web. MathML is a kind of XML and is suitable for expressing mathematical expressions with hierarchical

Petr Sojka (editor): DML 2009, Towards a Digital Mathematics Library, pp. 27–35.
 ISBN 978-80-210-4781-5

structures. The existing approaches can be categorized into the following two types: a translation-based approach and a structure-based approach.

With a translation-based approach, the mathematical expressions are parsed and translated into a set of natural language keywords. Then, the generated keywords are thrown to existing search engines. Munavalli et al. analyzed mathematical expressions written in MathML and translated the feature elements into text form in their search system; MathFind [1]. Youssef textualized, serialized, scoped, and normalized mathematical expressions into queries. This concept does not depend on the form of the expressions [2]. Adeel et al. generated keywords by using regular expressions for the expressions written in MathML, and threw them to search systems as queries [3]. A translation-based approach is advantageous in that it can exploit the excellent and up-to-date search capability of contemporary commercial search engines. On the other hand, it is often difficult to create an effective natural language query.

With a structure-based approach, the XML-structures of mathematical expressions are directly indexed and compared with the XML-structures of the queries. Kohlhase et al. developed Content Markup MathML, and made it possible to flexibly search for mathematical expressions in their search system: MathWebSearch [4]. Asperti et al. used a logic-independent metadata model for indexing mathematical expressions in their search system: Whelp [5]. Hashimoto and Hijikata constructed their search system by using XPath and a DOM structure of the mathematical expressions written in MathML [7,8]. Otagiri et al. use their unique query languages and expressions written in Content Markup MathML as targeted expressions [9]. A structure-based approach is capable of taking the structures of mathematical expressions into account. However, this approach has difficulty in ranking or ordering the search results according to their semantic similarity. Most of the previous studies simply enumerate the candidate expressions sharing a common sub-structure with a given query and if not, order them as to rank as only basic methods, and not much attention has been paid to the matching function to calculate the similarity score between the mathematical expressions. However, similarity calculation is particularly important in large scale mathematical search systems; it helps users to acquire additional knowledge, discover concealed relationships to different fields, and compensate for some false recognition.

Based on this background, we used a structure-based approach using Content Markup MathML in this paper, and propose a similarity search system that returns not only the expressions with exactly the same sub-structure, but also the expressions with high similarity scores.

This paper is organized as follows. In the next section, we briefly introduce MathML with its two variations: Presentation Markup and Content Markup. Next, in Sec. 3, we introduce the definition and our adaptation of the similarity measure that is based on the “Subpath Sets” of MathML Content Markup expressions. In Sec. 4, we propose a transformation of the Content Markup expressions to improve the search in both in efficiency and quality. In Sec. 5, the results of preliminary experiments using our prototype search systems are

shown followed by discussions. Finally, our conclusion and future work are presented in Sec. 6.

2 MathML

MathML [11] is a worldwide standard for describing mathematical contents. It is a kind of XML, therefore it can be easily slotted onto Web pages. It is widely used and is compatible with some applications, for example, Mathematica and even some browsers. There are two ways of writing in MathML, Presentation Markup and Content Markup.

The main purpose of Presentation Markup is to display mathematical expressions on Web browsers. Presentation Markup is superior in formatting and displaying mathematical expressions. However, there only about 30 kinds of basic symbols, so it is not easy to search for expressions because this limited number of symbols do not correspond to particular mathematical functions or meanings.

On the other hand, Content Markup focuses on the semantic construction of mathematical expressions, and it has over 100 kinds of semantic symbols corresponding to mathematical functions. Therefore, it is relatively easy to know the mathematical structures of the equations using Content Markup. Since Content Markup is unsuitable for formatting and displaying, it is usually used as an annotation of the expressions written in Presentation Markup.

3 Subpath Set

The definition of a similarity measure is crucial in similarity searching. Understanding that mathematical expressions written in MathML have their own tree structures, we use, in this paper, the degree of similarity between these tree structures as a basic notion of similarity.

As a similarity measure, we use the Subpath Set originally proposed in [6] to measure the distance between two syntactic trees. Here, Subpath is defined as "the path from the root to the leaves and all the sub-paths of that". Trees whose Subpath Sets overlap each other are considered to be similar as well. An example of a Subpath Set is shown in Fig. 1.

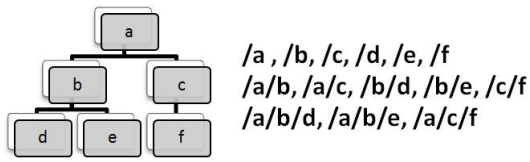

Fig. 1: Example of Subpath Set.

We used the following Jaccard coefficient in the experiments to score the overlap of the Subpath Sets.

$$\frac{\|S(t_1) \cap S(t_2)\|}{\|S(t_1) \cup S(t_2)\|} \tag{1}$$

where t_i is a tree and $S(t_i)$ is the Subpath Set of t_i. The Jaccard coefficient performed better than some of the other indicators, such as Dice coefficient, Simpson coefficient, Cosine coefficient, and some other ways of ranking in our preliminary study.

4 Transformation of Content-based MathML

In Content Markup, the "apply" symbol is the most frequently used of all the symbols. In fact, about 40–50 percent of the symbols in mathematical expressions written in Content Markup are for "apply". The roles of the "apply" symbol are mentioned as follows in the MathML specification by W3C [11].

> The most fundamental way of building up a mathematical expression in MathML content markup is the apply construct. An apply element typically applies an operator to its arguments. It corresponds to a complete mathematical expression. Roughly speaking, this means a piece of mathematics that could be surrounded by parentheses or "logical brackets" without changing its meaning.
>
> In MathML 2.0, the apply construct is used with all operators, including logical operators.

An "apply" symbol is almost always used for applying an operator, which is their first child, to the arguments, which are their other children. In addition, it is used whenever any functions and operators are used. It is useful to know the range a function or operator applies. However, in a search, they consume memory and also disguise meaningful sequences of the function operators on the sub-paths. In order to deal with this issue, we transform the original Content Markup into our new definition of *apply-free* Content Markup where the first children of the "apply" symbols take over their parents positions while other children remain in the same position, for example, as in Fig. 2.

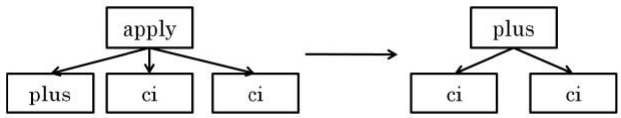

Fig. 2: Expression transformation.

However, there are some exceptional cases; a considerable number of expressions have "apply" symbols whose first children are also "apply" symbols. For example, in the target mathematical expressions we use in our experiments, about 1,100 out of 155,607 expressions have those double-sequence "apply" symbols. These are cases where the functions, i.e., the first children of the "apply" symbols, are composed of more than one word or have additional characters. In these cases, we keep the parent "apply" while removing the child "apply". That is, the "apply" symbols whose parents are also "apply" symbols are translated as in Fig. 3.

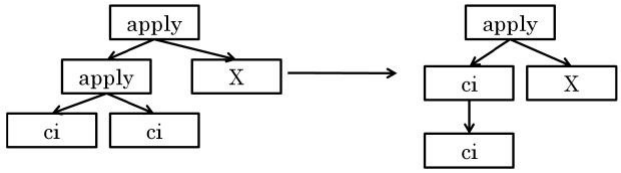

Fig. 3: Transformation for double-sequence "apply" symbols.

5 Experiments

5.1 Targeted formulas

In the experiments, 155,607 mathematical expressions, which were collected in Wolfram Functions Site [10], are used as targeted formulas. They are all written in Presentation Markup and they have Content Markup as annotations. We extracted three different representations for comparison: (i) Presentation Markup, (ii) the original Content Markup, and (iii) the *apply-free* Content Markup we defined in Sec. 4. In addition, we cut out some following symbols in Presentation Markup to improve the accuracy and dispose of vague information, as was the case for Hashimoto [7]; mrow, mstyle, semantics, annotation.

At first, we compare the statistical features of these three tree-forms: Presentation Markup, Content Markup, and *apply-free* Content Markup. The number of children in each symbol and the maximum depth of the tree construction in each expression are shown in Fig. 4, and the maximum and average values of them are listed in Table 1, where the *depth* in the table represents the depth of the tree construction and the *width* represents the number of children in each symbol.

Looking through this figure, we can see that the trees generated by Presentation Markup expressions are totally different from the ones generated by Content Markup or the *apply-free* Content Markup, both in their depths and in their numbers of children. The former group tends to be smaller in depth, but

has more children. In addition, when we compared Content Markup with *apply-free* Content Markup, it turned out that the depths of the two representations are very similar while the latter has less children. This can be easily explained since the translation in Sec. 4 does not change the depth of the tree, but does reduce the number of children. To summarize, tree construction of Presentation Markup is apt to broaden widthwise, and that of Content Markup is apt to broaden lengthwise.

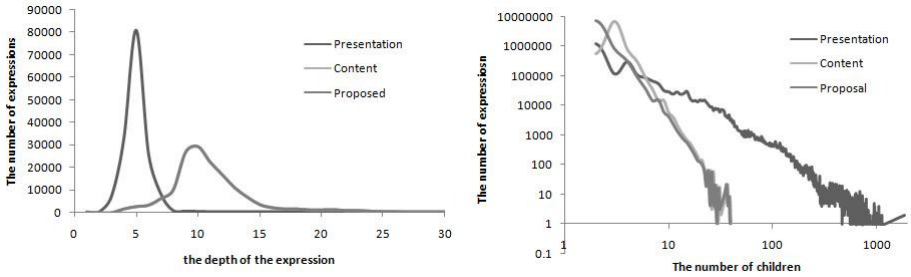

Fig. 4: Comparison of three types of MathML representations.

Table 1: Depth and width data in tree-construction of targeted expression.

Tree-construction form	Max depth	Ave. depth	Max width	Ave. width
Presentation Markup	16	5.000	11,438	7.187
Content Markup	54	11.102	77	2.903
Proposed form	54	11.102	76	2.044

5.2 Search results of example queries

Next, we implemented a prototype search system using Subpath Set. For the purpose of illustration, we selected a few sample queries and checked the rankings from the prototype system. Table 2 shows the top five rankings for the example queries where *apply-free* Content Markup was used as the MathML representation.

The results show that the proposed method is capable of evaluating the structural similarities of the trees rather than the notational similarities of the tokens. For example, cosine's additional theorem was ranked high in the first result listed in Table. 2, allowing for variations in function names.

Table 2: Sample search results.

Query : $\sin(a+b)=\sin(a)\cos(b)+\cos(a)\sin(b)$	
rank	results
1	$\sin(a+b)=\sin(a)\cos(b)+\cos(a)\sin(b)$
2	$\sin(a-b)=\sin(a)\cos(b)-\cos(a)\sin(b)$
3	$\sin(a+ib)=\sin(a)\cosh(b)+i\cos(a)\sinh(b)$
4	$\cos(a-b)=\cos(a)\cos(b)+\sin(a)\sin(b)$
5	$\cos(a+b)=\cos(a)\cos(b)-\sin(a)\sin(b)$

Query : $\tan(z)=\frac{\sinh(iz)}{\sinh(\frac{i\pi}{2}+iz)}$	
rank	results
1	$\tan(z)=\frac{\sinh(iz)}{\sinh(\frac{i\pi}{2}+iz)}$
2	$\tan(z)=\frac{\sinh(iz)}{\sinh(\frac{i\pi}{2}-iz)}$
3	$\sec(z)=\frac{i}{\sinh(\frac{\pi i}{2}+iz)}$
4	$\sec(z)=\frac{i}{\sinh(\frac{\pi i}{2}-iz)}$
5	$\cot(z)=\frac{\sinh(\frac{i\pi}{2}+iz)}{\sinh(iz)}$

5.3 Evaluation by the rank of an expected expression

Next, we compared the different forms of tree constructions, i.e., Presentation Markup, Content Markup, and *apply-free* Content Markup, using the same example queries. For each query, we manually selected an "expected" reference answer so that none of them were too large nor too close to the original queries. Then, we examined the ranks of the expected answers for the different representation forms. The results in which we observed a difference between the representation forms are listed in Table. 3, where the columns P, C, and A correspond to Presentation Markup, Content Markup, and *apply-free* Content Markup, respectively. The value "x" means that the reference expression did not appear in the top 100.

Table 3: Result ranking of expected expressions.

Query	Expected Expression	P	C	A
$\sin(a+b)=\sin(a)\cos(b)+\cos(a)\sin(b)$	$\sin(a-b)=\sin(a)\cos(b)-\cos(a)\sin(b)$	x	6	2
$\int \sin z dz=-\cos z dz$	$\int \sin(az)dz=-\frac{\cos z}{a}$	x	39	23
$\int z e^{az}dz=\frac{e^{az}(-1+az)}{a^2}$	$\int z^3 e^{az}dz=\frac{e^{az}(-6+6az+3a^2z^2+a^3z^3)}{a^4}$	x	17	5
$\int (e^{cz})^v dz=\frac{(e^{cz})^v}{cv}$	$\int \sqrt{e^{cz}}dz=\frac{2\sqrt{e^{cz}}}{c}$	x	5	2
$\mathrm{ArcSin}(z)=\frac{3\pi}{4}-\frac{1}{2}\mathrm{Arctan}(\frac{1-2z^2}{2z\sqrt{1-z^2}})$	$\mathrm{ArcCos}(z)=-\frac{\pi}{4}+\frac{1}{2}\mathrm{Arctan}(\frac{1-2z^2}{2z\sqrt{1-z^2}})$	33	79	16

The results show that Presentation Markup is not suitable for our search system. This may be because all the functions and operators are represented using the same symbol "mo" in Presentation Markup, and it shortens each sub-path that tree constructions in Presentation Markup are apt to broaden

widthwise, and therefore, the Subpath Sets have only a small amount of information. Comparing Content Markup and *apply-free* Content Markup, the latter performed slightly-better performance, which shows the advantage of the proposed free-apply transformation.

Here, we used only one reference answer for the evaluation, and a more comprehensive study for a qualitative evaluation will be necessary in the future. Since there has not been any reasonable scale of datasets for evaluation purposes in previous works on mathematical search systems, we believe such an effort would be worthwhile particularly for the enrichment of contents in today's digital mathematics libraries.

6 Conclusion & Future Work

In this paper, we proposed a new similarity search scheme for mathematical expressions. We started by introducing a similarity measure based on Subpath Set and proposed a MathML conversion that is apt for it. Based on the preliminary experiments, we believe that the proposed scheme has the potential to provide a flexible interface for searching for mathematical expressions on the Web. However, some important issues are left for future study.

First is the scalability issue. The proposed search system responds to a submitted query within a feasible amount time by using 155,607 formulas collected in Wolfram Functions Site [10]. However, with the variety and number of equations currently on the Web, the similarity calculation may become the bottleneck of the search. Therefore, some kinds of ingenious methods are needed, such as clustering targeted expressions, to pick up speed.

Second is the consideration of symbol values. Our current implementation recognizes only symbols and does not perceive the actual values or strings assigned to them. For example, expressions with different coefficients are treated as the same. While the simplification is effective process simplification, it sometimes degrades the performance. Therefore, in order to improve the accuracy of the similarity calculation, the values of the symbols should be taken into account to some extent.

Lastly, it is not easy to decide the similarity between two mathematical expressions using only their MathML descriptions. We are looking into the possibility of integrating other techniques including the existing translation-based approaches.

References

1. Rajesh Munavalli and Robert Miner: MathFind: A Math-Aware Search Engine. SIGIR. pp. 735–735, 2006.
2. Abdou Youssef: Information Search And Retrieval of Mathematical Contents: Issues And Methods. the ISCA 14th Int'l Conf. on Intelligent and Adaptive Systems and Software Engineering (IASSE-2005), July 20–22, Toronto, Canada, 2005.

3. Muhammad Adeel, Hui Siu Cheung, and Sikandar Hayat Khiyal: Math GO! Prototype of A Content Based Mathematical Formula Search Engine. Journal of Theoretical and Applied Information Technology, Vol4, No10, pp. 1002–1012, 2008.
4. Michael Kohlhase and Ioan A. Sucan: A Search Engine for Mathematical Formulae. Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006, Springer Verlag, pp. 241–253, 2006.
5. Andrea Asperi, Ferruccio Guidi, Claudio Sacerdoti Coen, Enrico Tassi, and Stefano Zacchiroli: A Content Based Mathematical Search Engine: Whelp Proceedings of TYPES 2004 conference: Types for Proofs and Programs, LNCS 3839, Springer Berlin / Heidelberg, ISBN 3-540-31428-8, pp. 17–32, 2006.
6. Hiroshi Ichikawa, Taiichi Hashimoto, Takenobu Tokunaga, and Hozumi Tanaka: New methods of retrieve sentences based on syntactic similarity. IPSJ SIG Technical Reports, DBS-136, FI-79, pp. 39–46, 2005.
7. Hideki Hashimoto, Yoshinori Hijikata, and Shogo Nishida: A Survey of index formats for the search of MathML objects. IPSJ SIG Technical Reports, DBS-142, FI-87, pp. 55–59, 2007.
8. Yoshinori Hijikata, Hideki Hashimoto, and Shogo Nishida: An Investigation of Index Formats for the Search of MathML Objects. Proc. of Intelligent Web Interaction Workshop (IWI 2007), pp. 244–248, DOI 10.1109/WI-IATW.2007. 121, Silicon Valley, USA, November, 2007.
9. Kenichi Otagiri and Tsuyoshi Murata: Search of Mathematical Formulas using MathML. The 22nd Annual Conference of the Japanese Society for Artificial Intelligence, 1F1-3, 2008.
10. The Wolfram Functions Site, Wolfram Research Inc. `http://functions.wolfram.com`.
11. Mathematical Markup Language (MathML) Version 2.0 (Second Edition), World Wide Web Consortium. `http://www.w3.org/TR/MathML2/`.