Search of Mathematical Contents: Issues and Methods*

Abdou Youssef
Department of Computer Science
The George Washington University
Washington, DC 20052
Email: ayoussef@gwu.edu

Abstract

Efforts are underway worldwide to create Web-accessible mathematical and scientific digital libraries. To benefit fully from such resources, users should be able to search not only for text, but also for equations and other math constructs. This paper will identify major issues that must be addressed in building math search systems, and then present techniques for developing a math search system on top of text search systems. Fundamental to this approach is a process that textualizes, serializes, scopes, and normalizes math expressions in contents and in queries. The performance of the approach is evaluated using a math search system that the author has developed.

1 Introduction

Efforts are underway worldwide to create and codify digital libraries of mathematical, scientific, and engineering contents [3, 4, 6, 10, 11, 12, 13, 14, 18, 20, 22, 23, 25, 28]. Notable examples include the Digital Library of Mathematical Functions (DLMF) at the National Institute of Standards and Technology [8], and the two XML-based mathematical markup languages, MathML [19] and OpenMath [18].

For users to benefit from such digital libraries, they need to be able to search easily and effectively not only for text, but also for equations and other mathematical constructs. Although text Search has reached a high level of maturity [29, 26], mathematical expressions are highly symbolic and structured that current search systems do not recognize. Moreover, mathematical equivalents, which are the counterpart of synonyms in text but much more complex, cannot be identified and utilized in current search engines. Also, math contents and queries involve many levels of abstraction and peculiar notational ambiguities that, though easily understood and resolved by humans, are beyond the current abilities of text search systems to handle.

The immediate goal of research on information retrieval (IR) in math digital libraries is to create a search system that (1) enables users to search not only for text, but also for mathematical expressions; and (2) allows users to express math queries naturally and easily, using the notation and idiom of mathematicians and scientists.

This paper will identify the issues that must be addressed in meeting that goal, and elucidate the text IR systems' limitations to overcome to provide math search and retrieval on those systems. Afterwards, the paper will present general approaches and specific techniques for addressing some of those issues. Finally, the performance of those methods will be evaluated using a math search system that the author has developed.

The paper is organized as follows. The next section surveys related work. Section 3 will identify the main issues and challenges of building a search system for math digital libraries. General approaches for building math search systems are outlined in Section 4, and specific methods and solutions for the issues and challenges identified in Section 3 are presented in detail in Section 5. In Section 6, the performance of those techniques is discussed. Finally, Section 7 concludes the paper.

2 Related Work and State of the Art

Besides the mature area of text search [5, 7, 29, 24, 26], one of the most related efforts is the Digital Library of Mathematical Functions (DLMF) project [15, 16, 21] at the National Institute of Standards and Technology (NIST). It is creating a major new mathematical reference source on the Web for special functions and their applications. It is estimated that over 80% of the DLMF handbook/Website contents are equations. Therefore, a math search system suitable for searching for and retrieving equations is essen-

*This material is based upon work supported by the National Science Foundation under Grant No. 0208818.

tial. The author is developing the math search system for DLMF [21].

In the core area of math search, some other work has started to appear. Recently, Guidi et al published papers on a math query language MathQL [11] and related searching techniques [10], both of which are for RDF metadata repositories, where RDF is the XML-based metadata markup language standard. The MathQL syntax is a markup style that is advanced in its expressive power, but requires the users to be advanced mathematicians. Our math query language is much higher level and easy to use, and does not require RDF contents. Another effort is a searching tool for integral-lookup [9].

There are other mathematical knowledge management research efforts [4, 6, 12, 17, 25]. None of them deals with math search, although MKM systems and math search systems can in the end be synergistically combined.

3 Math Search Issues

From the analysis of text IR systems and the experiences with building a math search system, we present here some of the major issues that have been identified. Methods for addressing most of these issues will be presented later.

1. *Defining an intuitive yet expressive math query language*

   Users of math digital libraries must be able to express their search needs in the "native", concise language of math. For example, to search for a document that contains the expression $(x + 1)^2$, the user need not enter as a query anything more than $(x + 1)^2$, or something very close such as $(x + 1)\hat{}2$.

2. *Bridging the query language with the language of the content files*

   Clearly, the native-math query language is bound to differ significantly from the language of the mathematical contents of digital libraries. The latter language is likely to be HTML, Latex, MathML, MS Word, PDF, or the like. Regardless of what approach is used for indexing and searching, the query language and the contents language must meet. Either the query language is translated to the contents language, or both languages are translated to an intermediary and structure-rich language before indexing and searching take place.

3. *Making IR systems "understand" math symbols & structures*

   Mathematical contents often involve symbols as in “P_n(x)”,“x^”, or “d^2y/dx^2 - x=”. Such symbols are often ignored or wrongly interpreted by current text-IR systems. Also, mathematical equations and other constructs have rich structural semantics. Current search systems do not recognize or index those structures. For example, to current search systems, a query like“sin(x + log x)” is probably the same as “sin x + log x”. Since searchers in technical fields would prefer to search based on term-roles, sub-expressions, and substructures, rather than on just mere occurrence of keywords, text-IR systems cannot meet these requirements. Therefore, math search systems must be able to recognize, process, and search for mathematical symbols, structures and substructures.

4. *Highlighting matched equations*

   Highlighting of matched terms inside retrieved documents is very convenient in general text search. It is even more crucial to highlight matched equations inside retrieved math documents because of the higher importance of equations relative to the rest of the document. Unlike text keywords, highlighting matched equation inside HTML documents, where equations are usually rendered as GIF images, is nontrivial. The string search techniques commonly used to find and highlight keywords inside documents do not apply to locating GIF images inside documents. Instead, new techniques are needed.

This is by no means an exhaustive list of issues. Other issues will be addressed in other papers.

4 General Approaches for Building Math Search Systems

Two broad approaches to building math search systems can be taken. The first is a text-IR-based approach, where new layers are wrapped around a capable but conventional text-IR system to make it math-aware in search. This approach leverages the advanced state of the art in text search, and is much faster to carry out.

The second is a radically different approach based on the emerging XML-based technologies and markup languages. This approach is likely to be much more powerful than the text-IR based approach, but these technologies are still evolving.

In this paper, we adopt the text-IR based approach to building math search systems.

Operator	Meaning
+, −, /, ∗	arithmetic operators
ˆ, **	Superscript or power
_	subscript
=>, <=>	imply (⇒), equivalent (⇔)
! =, *not* =	not equal (≠)
=-	equivalence (≡)
=˜	congruence (≅)

Table 1: A sample of operators in the math query language

Rendered Form	Query Syntax
$\int_0^\infty sin(\frac{1}{3}t^3 + xt)$	integral_ 0ˆinfinity sin((1/3)tˆ3+xt)
$\sqrt{Ai^2 + Bi^2}$	sqrt(Aiˆ2+Biˆ2)
$(\cdots)^{(x+2)}$	ˆ(x+2) // (x+2) as an exponent part
$\frac{\cdots}{(x+2)}$	.../(x+2) // (x+2) as a denominator

Table 2: A sample of math queries

5 Methods for Building a Text-IR Based Math Search Systems

5.1 A Math Query Language

The math query language is similar to but simpler than LaTeX. It includes almost all the standards names of functions in mathematics, including elementary functions and Special Functions, along with their syntax (i.e., number and order of each function's arguments and parameters). Table 1 gives a sample of operators, and Table 2 illustrates math queries. The language also incorporates Boolean, phrase and proximity operators of text-IR systems. For lack of space, the details and the grammar of the query language will be published elsewhere.

5.2 Techniques for Making Text IR Systems Math Capable

As mentioned in Section 3, IR systems should be able to "understand" math symbols and structures, and to bridge between the math query language and the language of the math contents. One technique to do so is to define an intermediary language that is purely textual (i.e., alphanumeric), and to map both the queries and the math content of the digital libraries to it. This will be elaborated in this subsection.

Algorithmically, the mapping process involves the following three principal steps:

1. Textualization of math symbols
2. Serialization and scoping of the various parts of terms and expressions
3. Normalization of the orders of parts into a standard canonical form.

Textualization turns each non-alphanumeric symbol into a unique alphanumeric word, resulting in a purely textual representation of queries and contents. For example,"" maps to the word plus,"-" maps to minus, and "ˆ " (for power or superscript) maps to beginsuperscript expression endsuperscript.

Serialization stacks the structural parts of an equation in a linear sequence, and scoping delineates and surrounds different parts and substructures of expressions with identifying tags. As illustrated in Figure 1, a definite integral is serialized into: (1) integral, (2) the lower limit, and (3) the upper limit. Scoping wraps (1) the lower limit with beginlowlimit endlowlimit, (2) the upper limit with beginupperlimit endupperlimit, and (3) the integrand with beginintegrand endintegrand; it also precedes the variable of integration with an appropriate term such as diff.

Serialization, coupled with textualization, makes it possible to search for math phrases using text-IR systems that support phrase search. Scoping allows users to search for equations by specifying terms in the various structural parts, such as integrands, numerators, denominators, function arguments, summands, and so on.

The third step, normalization, reorders the terms of the serialized & scoped forms into some defined canonical order so that certain variations in notations and writing styles in mathematics will not lead to search misses. The extent of normalization is determined by how much the designer wishes to support notational equivalences and mathematical equivalences. Table 3 illustrates normalization.

The above three steps give rise to an intermediary language, called TexSN (from Textualization, Serialization/scoping, and Normalization). The database

$$\int_a^b (x^2 + x/2)dx$$

(a)

```
integral
    beginlowerlimit a endlowlimit
    beginupperlimit b endupperlimit
    beginintegrand
        x beginsuperscript 2 endsuperscript
            plus
        frac
            beginnumerator x endnumerator
            begindenominator 2 enddenominator
    endintegrand
diff x
```

(b)

Figure 1: An Illustration of Textualization, Serialization and Scoping. Part (a) is a math expression, and part (b) is its textualized+serialized+scoped counterpart

Math Expression	**Normalized Form (with Comments**
x_2^3 expressed as x_ 2ˆ3 or xˆ3_ 2	*x beginsubscript* 2 *endsubscript* *beginsuperscript* 3 *endsuperscript* (Comment: Subscripts come before superscripts)
c+a+d+b	a plus b plus c plus d (Comment: We order the terms alphabetically when valid)
$ab^{-1}cd^1$	frac *beginnumerator* *a* times *c* *endnumerator* *begindenominator* *b* times *d* *enddenominator* (Comment: We get rid of negative powers)

Table 3: Illustration of normalization

files are converted to the TexSN form offline before they are indexed by a text search system. Also, every math query is translated online into a TexSN form Boolean query before it is handed to the text search system for searching and retrieval.

This three-step approach achieves several objectives at once. First, it renders math expressions, equations and other constructs indexable and searchable with a standard text search system. Second, it bridges the query language with the language of the math library, reconciling disparities in notation and mathematical idioms and idiosyncrasies between queries (users) on the one hand, and the library (authors), on the other hand. Third, it offers users a good measure of specificity about which part or structure of a mathematical expression that the keywords (or phrases) must be in.

5.3 Techniques for Equation Highlighting

To achieve equation highlighting, discussed in Section 3, a dual data model is defined. This entails the following:

- Each equation is given an identifier that is unique across the whole database.
- Each equation ID contains the name of the document containing the equation.
- The ID of each equation must be inserted in the vicinity of the equation inside the document that contains the equation.
- Although equations are embedded inside documents, they are extracted (but not deleted) from the documents, and individually stored as separate units (e.g., files), called equation storage units, somewhere in a file system or a database system.
- The name (or ID) of each equation storage unit must contain the ID of the corresponding equation, directly or in some codified form.
- The text IR system creates two index files: one for the document database, and another for the equation storage units. Thus the duality of the model.

For example, if E is the 12th equation in a document XYZ, then E must be given an ID of the form XYZ.12 (or something equivalent), and the equation storage unit for E must be given a name/ID of the form EQ.XYZ.12 (or something equivalent), and the ID XYZ.12 must be inserted next to the equation E inside the document XYZ.

When equation highlighting is to be performed (at hit-browsing time), the IDs of the matched equations and their native documents are extracted from the hit list of matching equations, and used to locate those equations in their native documents, and then to add highlighting markup (or stylesheet instructions, or the like) to the documents right before handing the documents to the browser for display.

6 Performance Evaluation

The math search system has been implemented on top of a text-IR system that supports Boolean operators, single- and multi-character wildcards, the phrase (adjacency) operator, and the proximity operator. We tested it and evaluated its performance on a collection of 300 mathematical documents containing about 2000 equations, measuring precision and recall.

The major problem in measuring performance of math search systems is the lack of any math query benchmarks because this area of search is quite new. For the system at hand, we tested it using about 50 queries believed to be quite representative for the time being. It was found that the system delivered 100% recall on all the queries, and that the precision ranged from 60% to 100%, averaging around 80%. Table 4 shows a small cross-section of the queries tested, and the corresponding precision and recall recorded.

In terms of query processing speed and turnaround time, it was found that the front-end processing (i.e., TexSNization) of a query before it is handed to the text IR system takes on the order of milliseconds on a Pentium 1.4GHz PC. Also, the time to distill the queries and the time to add equation highlights is in the order of tens of milliseconds (the latter case depends on the size of the document). All in all, the math-related components of the query processing is a tiny percentage of the overall search and retrieval time performed by the text-IR system. The whole turnaround from the moment of submitting the query to the moment of seeing the hit equation list is about 1 second, and the time to display a document with all the highlights is also about one second.

7 Conclusions and Future work

This paper has identified the major issues involved in creating a math search system, and presented a number of techniques, methods and ideas for building a math search system on top of a conventional text search system. The powers and limitations of this approach were highlighted, and potential alternative techniques for future enhancements were pointed out. The tests show that the performance of text-IR based math search is very good, and is likely to yield high user satisfaction. The search capabilities are expected to be adequate for the majority of math & science users, from middle school students all the way to graduate students and professionals in math and science.

Query	Precision	Recall
x^2	91%	100%
$\Gamma(1/2+\mu-\kappa)$	100%	100%
$\Gamma(1/2+*-*$	100%	100%
Ai^2+Bi^2	75%	75%
$\int^x$	62%	100%
$\int_x$	100%	100%
$\int_0^\infty sin(\frac{1}{3}t^3+xt)$	100%	100%

Table 4: System performance on a sample of queries

References

[1] I. M. Author, An Article I Wrote,” *The Journal Name*, Vol. 11, pp. 3-14, 1982.

[2] Next Author, *Title of Book*, the Publisher, 1982.

[3] The ActiveMath Project, http://www.mathweb.org/activemath/

[4] A. Asperti et al, “Mathematical Knowledge Management in HELM [Italy],” *1st International Workshop on Mathematical Knowledge Management,* Schloss Hagenberg, Austria, September 24-26, 2001.

[5] N. J. Belkin et al, “Rutgers' TREC-7 Interactive Track Experience” *TREC-7,* NIST, page 275, 1999.

[6] B. Buchberger,“Mathematical Knowledge Management Using Theorema”, *1st International Workshop on Mathematical Knowledge Management,* Schloss Hagenberg, Austria, September 24-26, 2001.

[7] C. Buckley, J. Walz, M. Mitra, and C. Cardie, “SMART High Precision”, *TREC 7,* NIST, page 285, 1999.

[8] The Digital Library of Mathematical Functions (DLMF), the National Institute of Standards and Technology, http://dlmf.nist.gov/

[9] T. H. Einwohner and R. Fateman, "Searching techniques
for integral tables," *International symposium on Symbolic and algebraic computation,* ACM, 1995. (http://torte.cs.berkeley.edu:8010/tilu)

[10] F. Guidi, *Searching and Retrieving in Content-based Repositories of Formal Mathematical Knowledge,* Ph.D. Thesis in Computer Science, University of Bologna, March 2003. Technical report UBLCS 2003-06.

[11] F. Guidi and I. Schena, "A Query Language for a Metadata Framework about Mathematical Resources," *the 2nd International Conf. Mathematical Knowledge Management,* Bertinoro, Italy, Feb. 2003.

[12] T. Hardin, "Mathematical Knowledge Management in FOC [France]", Schloss Schloss Hagenberg, Austria, September 24-26, 2001.

[13] An Hypertextual Electronic Library of Mathematics, http://helm.cs.unibo.it/.

[14] M. Kohlhase, "MBase: Representing Knowledge and Context for the Integration of Mathematical Software Systems", to appear in *the Journal of Symbolic Computation.*

[15] Daniel W. Lozier, "The DLMF Project: A New Initiative in Classical Special Functions," *International Workshop on Special Functions - Asymptotics, Harmonic Analysis and Mathematical Physics,* Hong Kong, June 21-25, 1999.

[16] D.W.Lozier, B.R.Miller, B.V.Saunders, "Design of a Digital Mathematical Library for Science, Technology and Education," *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries; IEEE ADL '99,* Baltimore, Maryland , May 1999.

[17] MathWeb.org, http://www.mathweb.org/.

[18] The OpenMath Standard, 1998, http:///www.openmath.org/

[19] MathML 2.0, a W3C Recommendation, October, 2003, http://www.w3.org/Math/.

[20] E. Melis et al, "ActiveMath: A Generic and Adaptive Web-Based Learning Environmen", *Artifical Intelligence in Education,* vol 12, no 4, Winter 2001.

[21] Bruce Miller and Abdou Youssef, "Technical Aspects of the Digital Library of Mathematical Functions" *Annals of Mathematics and Artificial Intelligence,* Vol. 38, pp. 121-136, 2003.

[22] MathWeb.org, http://www.mathweb.org/.

[23] The Omega Group, http://www.ags.uni-sb.de/ omega/.

[24] S.E. Robertson, S. Walker, and M. Beaulieu,"Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactiv", *TREC-7*, NIST, page 253, 1999.

[25] P. Rudnicki and A. Trybulec, "Mathematical Knowledge Management in MIZAR", *1st International Workshop on Mathematical Knowledge Management,* Schloss Hagenberg, Austria, September 24-26, 2001.

[26] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval,* McGraw Hill, New York, 1993.

[27] T. Saracevic,"Relevance: A Review of and a Framework for the Thinking on the Notion," *Journal of the American Society of Information Science*, 26(4), 1975, pp. 321-343.

[28] Theorist Interactive LiveMath, http://www.livemath.com/

[29] Baeza-Yates, R. and B. Ribeiro-Neto (1999). *Modern information retrieval,* Reading, MA: Addison-Wesley.