The Art of Mathematics Retrieval

Petr Sojka
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
sojka@fi.muni.cz

Martin Líška
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
255768@mail.muni.cz

ABSTRACT

The design and architecture of MIaS (Math Indexer and Searcher), a system for mathematics retrieval is presented, and design decisions are discussed. We argue for an approach based on Presentation MathML using a similarity of math subformulae. The system was implemented as a math-aware search engine based on the state-of-the-art system Apache Lucene.

Scalability issues were checked against more than 400,000 arXiv documents with 158 million mathematical formulae. Almost three billion MathML subformulae were indexed using a Solr-compatible Lucene.

Categories and Subject Descriptors

H.3.7 [**Information Systems**]: Information storage and Retrieval—*Information Search and Retrieval*; I.7 [**Computing Methodologies**]: Document and text Processing—*Index Generation*

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

MIaS, WebMIaS, digital mathematics libraries, information systems, math indexing and retrieval, mathematical content representation

There is no abstract art. You must always start with something. Afterward you can remove all traces of reality.
Pablo Picasso

1. INTRODUCTION

The solution to the problem of mathematical formulae retrieval lies at the heart of building digital mathematical libraries (DML). There have been numerous attempts to solve this problem, but none have found widespread adoption within the wider mathematics community. And as yet, there is no widely accepted agreement on the math search format to be used for mathematical formulae by library systems or by Google Scholar.

MathML standard by W3C has become the standard for mathematics exchange between software tools. Almost no MathML is written directly by authors—they typically prefer a compact notation of some TEX flavour such as LATEX or AMSLATEX. The designer of a search system for mathematics is thus faced with the task of converting data to a unifying format, and allowing DML users to use their preferred notation when posing queries. [AMS]LATEX or similar TEX macropackages are the typical preferences; Presentation MathML or Content MathML are used only when available as outputs of a software system.

During the integration of existing DMLs into larger projects such as EuDML [9], the unsolved math search problem becomes evident—D*M*L without math search support is an oxymoron. We have evaluated several systems for our goals: 1) **MathDex**[1] (formerly MathFind [6]); 2) **EgoMath**[2] developed by Josef Mišutka as an extension of a full text websearch core engine Egothor [5]; 3) **LATEXSearch**[3], a search tool offered by Springer in SpringerLink; 4) **LeActiveMath**[4] search, developed as part of the ActiveMath-EU project and 5) **MathWebSearch**[5] is an MSE developed in Bremen/Saarbrücken by Kohlhase et al. [1] Our evaluation [7] has lead us to conclude that there is no satisfactory, math-aware and scalable solution. For this reason, we designed and implemented [3, 7] a *new* robust solution for retrieving mathematical formulae.

Section 2 presents our design of this scalable and extensible system for searching mathematics, taking into account not only inherent structure of mathematical formulae but also formula unification and subformulae similarity measures. Our evaluation of a prototypical implementation on the Apache Lucene open source full-featured search engine library is presented in Section 3. The paper concludes with a description of the WebMIaS interface and listing future work directions in Section 4 and with a summary in Section 5.

Art is never chaste. It ought to be forbidden to ignorant innocents, never allowed into contact with those not sufficiently prepared. Yes, art is dangerous. Where it is chaste, it is not art.
Pablo Picasso

2. DESIGN OF MIAS

We have developed a math-aware, full-text based search engine called *MIaS* (Math Indexer and Searcher). It processes documents containing mathematical notation in MathML format. MIaS allows users to search for mathematical formulae as well as the textual content of documents.

Since mathematical expressions are highly structured and have no canonical form, our system pre-processes formulae in several steps to facilitate a greater possibility of matching two equal expressions with different notation and/or non-equal, but similar formulae. With

[1] www.mathdex.com/
[2] egomath.projekty.ms.mff.cuni.cz/egomath/
[3] www.latexsearch.com/
[4] devdemo.activemath.org/ActiveMath2/
[5] search.mathweb.org/index.xhtml

DocEng2011, September 19–22, 2011, Mountain View, California, USA

an analogy to natural language searching, MIaS searches not only for whole sentences (whole formulae), but also for single words and phrases (subformulae down to single variables, symbols, constants, etc.). For calculating the relevance of the matched expressions to the user's query, MIaS uses a heuristic weighting of indexed mathematical terms, which accordingly affects scores of matched documents and thus the order of results.

At the end of all processing methods, formulae are converted from XML nodes to a linear string form, which can be handled by the indexing core.

2.1 Math Indexing Workflow

The top-level indexing scheme with a detailed view of the mathematical part is shown in Figure 1.

MIaS is currently able to index documents in XHTML, HTML and TXT formats. As Figure 1 shows, the input document is first split into textual and mathematical parts. The textual content is indexed in a conventional way, mathematics needs to be processed differently.

2.2 Math preprocessing

Mathematical expressions are pre-analyzed in several steps to facilitate searches not only for exact whole formulae, but also for subparts (tokenization) and for similar expressions (formulae unifications). This addresses the issue of the static character of full-text search engines and creates several representations of each input formula, all of which are indexed.

Tokenization is a straightforward process of obtaining subformulae from an input formula. MIaS makes use of Presentation MathML markup where all logical units are enclosed in XML tags which makes obtaining all subformulae a question of tree traversal.

MIaS performs three types of unification algorithms, the goal of which is to create several more or less generalized representations of all formulae obtained through the tokenization process. These steps allow the system to return similar matches to the user query while preserving the formula structure and α-equality [5]:

1. Ordering—ordering of the operands of the commutative operations;
2. Variables unification—substitution of all variables for unified symbols (ids) while preserving bound variables;
3. Constants unification—substitution of all number constants for one unified symbol (const).

During the searching phase, a query can match several terms in the index. However one match can be more important to the query than another, and the system must consider this information when scoring matched documents. Each indexed mathematical expression has a weight (relevance score) assigned to it describing how far the actual formula is from its original representation. It is computed throughout the whole indexing phase by individual processing steps following this basic rule of thumb—the more modified a formula and the lower the level of a subformula, the less weight is assigned to it.

It is impossible to assemble a weighting function that is exactly right. Such a function needs to consider a document base on which the system will run as well as the established customs in a particular scientific field. We tried to create a complex and robust weighting function that would be appropriate to many fields.

At the end of the preprocessing phase, mathematical formulae are transformed from XML nodes to a compacted string form so it can be handled by a full-text indexing core.

An example of the formula preprocessing is displayed in Figure 2.

2.3 Searching

In the search phase, user input is again split into mathematical and textual parts. Formulae are then preprocessed in the same way as in the indexing phase, except for tokenization—we doubt that users would want to search for subparts of a queried expression rather than the whole.

Art is the elimination of the unnecessary.
Pablo Picasso

3. EVALUATION

For large scale evaluation, we needed an experimental implementation and a corpus of mathematical texts.

3.1 Implementation

The Math Indexer and Searcher is written in Java. The role of full-text indexing and searching core is performed by Apache Lucene 3.1.0. The mathematical part of document processing can be seen as a standalone pluggable extension to any full-text library, however it needs custom integration for each one. In the case of Lucene, a custom Tokenizer (MathTokenizer) has been implemented.

When searching for mathematical formulae, their weights need to be considered in the final score of the document. The scoring function of our MIaS system adds one parameter to the Lucene's standard practical scoring function (described in detail at `http://lucene.apache.org/java/3_1_0/api/all/org/apache/lucene/search/Similarity.html`)—weight w of one matched formula:

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \cdot \sum_{t\ in\ q} \left(tf(t \text{ in } d) \cdot avg(w) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t,d) \right) \quad (1)$$

If a document contains the same formula more than once (each occurrence can have different weight assigned), the average value of all the weights is taken into consideration ($avg(w)$).

3.2 Corpus of Mathematical Documents

A document corpus MREC (Mathematical REtrieval Corpus) with 439,423 scientific documents was used to evaluate the behavior of the system we modelled. The documents come from the arXMLiv project that is converting document sets from arXiv into XHTML + MathML (both Content and Presentation) [8]. The resulting corpus size was 124 GB uncompressed, 16 GB compressed. This corpus of documents (MREC version 2011.4.439) is available for download at `http://nlp.fi.muni.cz/projekty/eudml/MREC`.

3.3 Results

Math Indexer and Searcher demonstrated the ability to index and search a relatively vast corpus of real scientific documents. Its usability is greatly improved thanks to its preprocessing functions together with the formulae weighting model. The ability to search for exact and similar formulae and subformulae, more so with customizable relevance computation, demonstrates an unquestionable contribution to the whole search experience.

It is very difficult, if not impossible, to completely verify the correctness of the theoretical considerations made in the previous sections and thus correctness of search results. For this purpose, a sufficiently large corpus of documents with fully controlled content would be needed. For any assembled query, there should exist beforehand a complete list of the documents ordered by their relevance to the query to compare the actual results to. On the other hand, the real world relevance of the results being returned needs to be verified by an extensive user study and perhaps several parameters need to be adjusted for the best results.

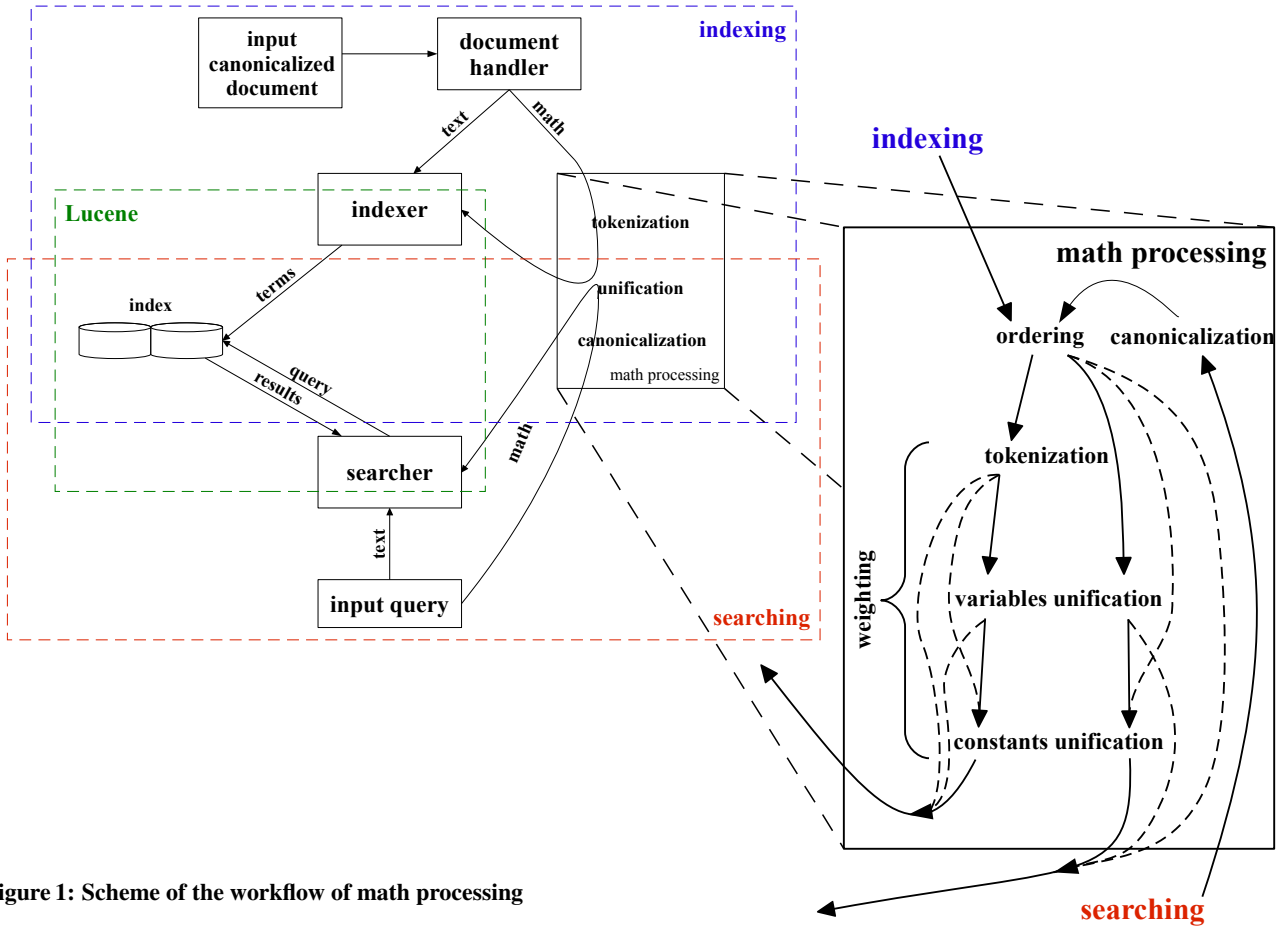

Figure 1: Scheme of the workflow of math processing

We have applied an empirical approach to the evaluation so far. For these purposes we created a demo web interface, see Section 4.

3.4 Scalability Testing and Efficiency

We have devised a scalability test to see how the system behaves with an increasing number of documents and formulae indexed. Subsets containing 10,000, 50,000, 100,000 ... and the complete 439,423 documents were gradually indexed and several values were measured: the number of input formulae, the number of indexed formulae, the indexing run-time and indexing CPU-time. Our observation showed that system scales linearly in proportion to the number of documents.

The whole document set contained 158,106,118 formulae, after all the preprocessing was done, our system indexed 2,910,314,146 expressions and the indexing run-time was 1,378 min (almost 23 hours) and the resulting index size was approx. 63 GB. We also measured an average query time by querying the created index with a set of differently complex queries (mixed, non-mixed, more/less complex single/multiple formulae). Resulting average query time was 469 ms.

You don't make art, you find it.
Pablo Picasso

4. WEB INTERFACE AND FUTURE WORK

To allow user evaluation, we have created web interface for MIaS—*the WebMIaS system* [4]—available at `http://nlp.fi.muni.cz/projekty/eudml/mias/`. WebMIaS allows the retrieval of mathematical expressions written in TEX or MathML. TEX queries are converted on-the-fly into tree representations of Presentation MathML, which is used for indexing. WebMIaS allows complex queries composed of plain text and mathematical formulae. It currently works over our complete mathematical corpus, MREC.

As the semantically same formulae can be represented by different MathML notation, it is evident that some kind of normalization of MathML is necessary. We use Canonical MathML [2] as normalization MathML format and are using the software library UMCL supporting it. Canonicalization (converting to a canonicalized form of MathML) is used both during the indexing and querying phases. It not only increases fairness of similarity ranking, but also helps to match a query against the indexed form of MathML. We plan to extend the effect of the commutative ordering part of the normalization mentioned in 2.2 by arranging a full list of commutative operators and all the operators with their priorities so the ordering can be perfected.

Another area of long-term research planned is supporting Content MathML, in a way similar to the current handling of Presentation MathML. The architectural design is open to it, but as most of the math available is in Presentation MathML taken from PDFs, this is not currently a high priority.

Art is the lie that enables us to realize the truth.
Pablo Picasso

5. CONCLUSIONS

We have presented an approach to mathematics searching and indexing—the architecture and design of the MIaS system, and

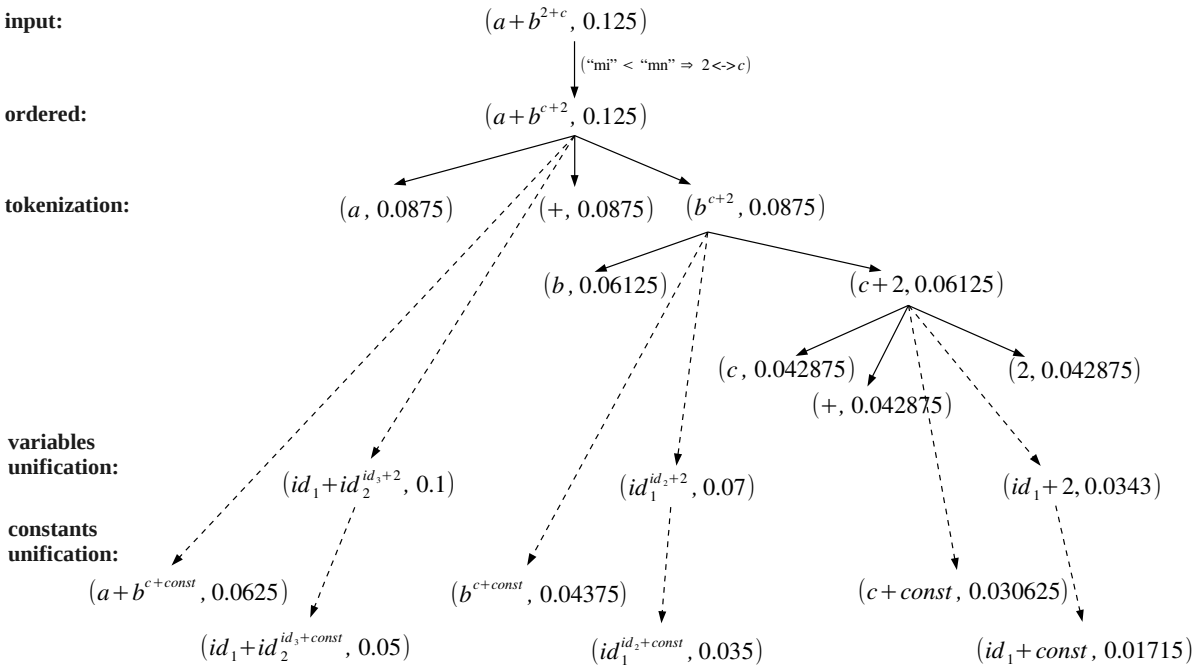

Figure 2: Example of formula preprocessing. Ordered pairs are (<expression written naturally>, <it's weight>). All expressions as shown are indexed, except for the original one.

its WebMIas interface. The feasibility of our approach has been verified on large corpora of real mathematical papers from arXMLiv. Scalability tests have confirmed that the computing power needed for fine math similarity computations is readily available and allows the use of this technology for projects on world-wide scale.

Acknowledgements. This work has been partially supported by the Ministry of Education of CR within the Center of Basic Research LC536 and by the European Union through its Competitiveness and Innovation Programme (Policy Support Programme, 'Open access to scientific information", Grant Agreement No. 250503). We thank Michal Růžička for help with figure drawings and web form of MIaS interface.

Bad artists copy. Good artists steal.
Pablo Picasso

6. REFERENCES

[1] Ş. Anca. Natural Language and Mathematics Processing for Applicable Theorem Search. Master's thesis, Jacobs University, Bremen, Aug. 2009. https://svn.eecs.jacobs-university.de/svn/eecs/archive/msc-2009/aanca.pdf.

[2] D. Archambault and V. Moço. Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations. In K. Miesenberger, J. Klaus, W. Zagler, and A. Karshmer, editors, *Computers Helping People with Special Needs*, volume 4061 of *Lecture Notes in Computer Science*, pages 1191–1198. Springer Berlin / Heidelberg, 2006. http://dx.doi.org/10.1007/11788713_172.

[3] M. Líška. Vyhledávání v matematickém textu (in Slovak), Searching Mathematical Texts, 2010. Bachelor Thesis, Masaryk University, Brno, Faculty of Informatics (advisor: Petr Sojka), https://is.muni.cz/th/255768/fi_b/?lang=en.

[4] M. Líška, P. Sojka, M. Růžička, and P. Mravec. Web Interface and Collection for Mathematical Retrieval. In P. Sojka and T. Bouche, editors, *Proceedings of DML 2011*, pages 77–84, Bertinoro, Italy, July 2011. Masaryk University. http://www.fi.muni.cz/~sojka/dml-2011-program.html.

[5] J. Mišutka and L. Galamboš. Extending Full Text Search Engine for Mathematical Content. In P. Sojka, editor, *Proceedings of DML 2008*, pages 55–67, Birmingham, UK, July 2008. Masaryk University. http://dml.cz/dmlcz/702546.

[6] R. Munavalli and R. Miner. MathFind: A Math-Aware Search Engine. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 735–735, New York, NY, USA, 2006. ACM. http://doi.acm.org/10.1145/1148170.1148348.

[7] P. Sojka and M. Líška. Indexing and Searching Mathematics in Digital Libraries – Architecture, Design and Scalability Issues. In J. H. Davenport, W.M. Farmer, J. Urban and F. Rabe, editors, *Proceedings of CICM Conference 2011 (Calculemus/MKM)*, volume 6824 of *Lecture Notes in Artificial Intelligence, LNAI*, pages 228–243, Berlin, Germany, July 2011. Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-22673-1_16.

[8] H. Stamerjohanns, M. Kohlhase, D. Ginev, C. David, and B. Miller. Transforming Large Collections of Scientific Publications to XML. *Mathematics in Computer Science*, 3:299–307, 2010. http://dx.doi.org/10.1007/s11786-010-0024-7.

[9] W. Sylwestrzak, J. Borbinha, T. Bouche, A. Nowiński, and P. Sojka. EuDML—Towards the European Digital Mathematics Library. In P. Sojka, editor, *Proceedings of DML 2010*, pages 11–24, Paris, France, July 2010. Masaryk University. http://dml.cz/dmlcz/702569.