MIaS: Math-Aware Retrieval in Digital Mathematical Libraries

Petr Sojka
Masaryk University
Faculty of Informatics
Brno, Czech Republic
sojka@fi.muni.cz

Michal Růžička
Masaryk University
Faculty of Informatics
Brno, Czech Republic
mruzicka@mail.muni.cz

Vít Novotný
Masaryk University
Faculty of Informatics
Brno, Czech Republic
witiko@mail.muni.cz

ABSTRACT

Digital mathematical libraries (DMLs) such as arXiv, Numdam, and EuDML contain mainly documents from STEM fields, where mathematical formulae are often more important than text for understanding. Conventional information retrieval (IR) systems are unable to represent formulae and they are therefore ill-suited for math information retrieval (MIR). To fill the gap, we have developed, and open-sourced the MIaS MIR system. MIaS is based on the full-text search engine Apache Lucene. On top of text retrieval, MIaS also incorporates a set of tools for preprocessing mathematical formulae. We describe the design of the system and present speed, and quality evaluation results. We show that MIaS is both efficient, and effective, as evidenced by our victory in the NTCIR-11 Math-2 task.

KEYWORDS

Math Information Retrieval, Digital Mathematical Libraries

ACM Reference Format:
Petr Sojka, Michal Růžička, and Vít Novotný. 2018. *MIaS*: Math-Aware Retrieval in Digital Mathematical Libraries. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3269206.3269233

1 INTRODUCTION

In mathematical discourse, formulae are often more important than text for understanding. As a result, digital mathematical libraries (DMLs) require math information retrieval (MIR) systems that recognize both text and math in documents and queries. Conventional IR systems represent both text, and formulae using the bag-of-words vector-space model (VSM). However, the VSM captures neither the structural, nor the semantic similarity between mathematical formulae, which makes it ill-suited for MIR.

To fill the gap, new math-aware IR systems started to appear after the pioneering workshop on DMLs [18]. Springer's LaTeX Search[1] system takes formulae from papers with available LaTeX sources, and hashes the formulae to obtain a text representation. Zentralblatt Math uses the MathWebSearch system[2] [8], which represents formulae with substitution trees. We have developed and open-sourced the MIaS (Math Indexer and Searcher) system[3] [16, 14] using the robust highly-scalable full-text search engine Apache Lucene [5] and our own set of tools for the preprocessing of mathematical formulae. Since 2012, MIaS has been deployed in the European Digital Mathematical Library (EuDML)[4], making it historically the first system to be deployed in a DML.

2 SYSTEM DESCRIPTION

MIaS processes text and math separately. The text is tokenized and stemmed to unify inflected word forms. Math is expected to be in the MathML format[5]. Open tools such as Tralics[6], LaTeXML[7] convert documents in the popular math authoring language of LaTeX to MathML. Other tools such as InftyReader [21] , and MaxTract [4] convert raster, and vector PDF documents, respectively, to MathML. The math is then canonicalized, ordered, tokenized, and unified (see Figure 1). We will describe each of these processing steps in detail in the following paragraphs.

Canonicalization. As explained above, MathML can originate from multiple sources and each can encode equivalent mathematical formulae a little differently. To obtain a single *canonical* representation, we initially used the third-party MathML canonicalizer from the UMCL library that converts math to a subset of MathML called the Canonical MathML [3]. However, since the conversion speed and accuracy did not match our expectations, we have developed and open-sourced our own MathML canonicalizer[8] [7].

Ordering. MathML canonicalization only affects the encoding of mathematical formulae and does not result in any syntactic manipulation. We go a step further and reorder the operands of commutative operators alphabetically. For example, we convert the formulae $a + b$, and $b + a$ to a single canonical form $a + b$.

Tokenization. A user of our system may not know the precise form of a formula they are searching for. To enable partial matches, we index not only the original formula, but also all its *subformulae*, which correspond to all the XML subtrees of the original formula XML tree. To penalize partial matches, the weight of subformulae is inversely proportional to their depth in the XML tree. [19]

A user is likely interested in documents that contain either the query formula itself, or larger formulae with the query formula as a subformula. On the other hand, a user is unlikely to be interested in documents that contain only small parts of the query formula, such as isolated numbers, and symbols. For that reason, we only tokenize formulae in indexed documents, not in user queries.

[1] https://www.ams.org/notices/201004/rnoti-apr10-cov4.pdf
[2] https://zbmath.org/formulae/
[3] https://github.com/MIR-MU/MIaS

CIKM '18, October 22–26, 2018, Torino, Italy

[4] https://eudml.org/search
[5] https://www.w3.org/TR/MathML3/
[6] https://www-sop.inria.fr/marelle/tralics/
[7] https://dlmf.nist.gov/LaTeXML/
[8] https://github.com/MIR-MU/MathMLCan

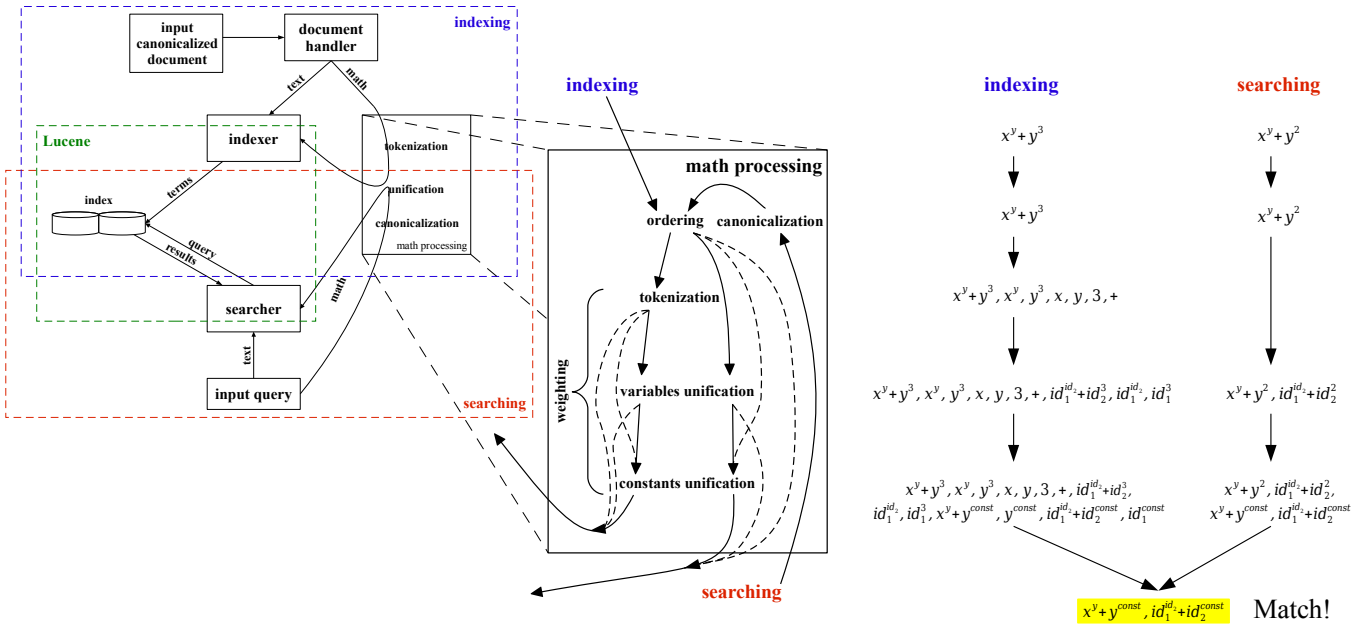

Figure 1: The preprocessing of mathematical formulae in indexed and query documents

Unification. In theory, the naming of variables does not affect the meaning of formulae. To match formulae in different notations, we replace each variable with a numbered identifier. For example, we convert the formulae $a + b^a$, and $x + y^x$ to a single *unified* form $\mathrm{id}_1 + \mathrm{id}_2^{\mathrm{id}_1}$. In practice, many fields have an established notation and variable names are meaningful. To encourage precise matches, we keep the original formulae in addition to the unified formulae.

Two formulae that only differ in numeric constants are often related. For example, both $3x^2-2x+2$, and $8x^2-3x+6$ are quadratic polynomials. We replace every numeric constant with a constant identifier. For example, we convert the above formulae to a single unified form $\mathrm{const}x^2-\mathrm{const}x+\mathrm{const}$. To encourage precise matches, we keep the original formulae in addition to the unified formulae.

In predicate logic, a variable can represent an arbitrary formula. For example, the formulae $a^2+\frac{\sqrt{b}}{c}$, and $a^2+\frac{x}{y}$ are equivalent if x equals $\sqrt{b}$. Starting with the deepest subformulae, we replace all subformulae at a given depth with a unifying identifier. [15] For example, we convert the formula $a^2+\frac{\sqrt{b}}{c}$ to a sequence of *structurally unified* formulae $a^2+\frac{\sqrt{\square}}{c}$, $\square^{\square}+\frac{\square}{\square}$, and $\square+\square$ and the formula $a^2+\frac{x}{y}$ to a sequence of structurally unified formulae $\square^{\square}+\frac{\square}{\square}$, and $\square+\square$. To penalize partial matches, the weight of the formulae is proportional to the depth of replacement. To encourage precise matches, we keep the original formulae in addition to the unified formulae. We have open-sourced the MathML structural unificator[9].

After preprocessing, a query consists of a weighted set of terms, and formulae. Since we are now going to search for documents that match at least one term, and at least one formula from the query, ill-posed terms, and formulae will negatively impact the recall of our system. To overcome this problem, we remove selected terms and formulae to produce a set of *subqueries*. Figure 2 shows an example strategy for producing subqueries. Líška, Sojka, and Růžička [10] describe other strategies that we use. We then submit the subqueries to Apache Lucene and receive ranked lists of resulting documents. Since the scores of the resulting documents are incomparable between subqueries, we cannot merge and rerank the individual result lists. Instead, we interleave them to obtain the final search results that we present to the user.

Subquery 1:	f_1	f_2	t_1	t_2	t_3
Subquery 2:	f_1	f_2	t_1	t_2	
Subquery 3:	f_1	f_2	t_1		
Subquery 4:	f_1	f_2			
Subquery 5:	f_1		t_1	t_2	t_3
Subquery 6:			t_1	t_2	t_3

Figure 2: The subqueries produced from the original query $f_1 f_2 t_1 t_2 t_3$ with mathematical formulae f_1, and f_2 and terms t_1, t_2, and t_3 using the Leave Rightmost Out (LRO) strategy.

To provide a web user interface to MIaS, we have developed and open-sourced WebMIaS[10, 11] [16, 11]. Users can input their query in a combination of text, and math with a native support for LaTeX provided by Tralics, and MathJax [6]. Matches are conveniently highlighted in the search results. The user interface of WebMIaS is shown in Figure 3. We have deployed a demo of the latest development version of WebMIaS[12] using the Apache Tomcat[13] implementation of the Java Servlet. The demo uses an index

[9] https://github.com/MIR-MU/MathMLUnificator

[10] https://mir.fi.muni.cz/webmias/

[11] https://github.com/MIR-MU/WebMIaS

[12] https://mir.fi.muni.cz/webmias-demo/

[13] https://tomcat.apache.org/

Figure 3: The user interface of WebMIaS. Users can input their query in a combination of text, and math with native support for LaTeX provided by Tralics, and MathJax. Matches are conveniently highlighted in the search results.

Table 1: Speed evaluation results on the MREC dataset using 448G of RAM, and eight Intel Xeon™ X7560 2.26 GHz CPUs.

	Mathematical (sub)formulae		Indexing time (min)	
Docs	Input	Indexed	Real	CPU
10,000	3,406,068	64,008,762	35.75	35.05
50,000	18,037,842	333,716,261	189.71	181.19
100,000	36,328,126	670,335,243	384.44	366.54
200,000	72,030,095	1,326,514,082	769.06	733.44
300,000	108,786,856	2,005,488,153	1,197.75	1,116.64
350,000	125,974,221	2,318,482,748	1,386.66	1,298.10
439,423	158,106,118	2,910,314,146	1,747.16	1,623.22

Table 2: Speed evaluation results on the NTCIR-11 Math-2 dataset using the same computer as above.

	Mathematical (sub)formulae		Indexing time (min)	
Docs	Input	Indexed	Real	CPU
8,301,545	59,647,566	3,021,865,236	1940.07	3,413.55

built from a subset of the arXMLiv dataset [20] made available to the NTCIR-12 conference participants and will serve as the basis for our live demonstration at the conference.

3 EVALUATION

We performed a speed evaluation of MIaS on the MREC dataset of 439,423 documents [13] (see Table 1), a quality and speed evaluation on the NTCIR-10 Math [1, 12] dataset of 100,000 documents, and a quality and speed evaluation on the NTCIR-11 Math-2 [2, 16] (see Tables 2, and 3), and NTCIR-12 MathIR [22, 15] dataset of 105,120 documents that were split into 8,301,578 paragraphs. Speed evaluation shows that the indexing time of our system is linear in the number of indexed documents and that the average query time is 469 ms. With respect to quality evaluation, MIaS has notably won the NTCIR-11 Math-2 task.

4 CONCLUSION AND FUTURE WORK

With the growing importance of DMLs, there is a growing demand for effective MIR systems. The evaluation shows that our open-source MIaS system is both efficient, and effective while building

Table 3: Quality evaluation results on the NTCIR-11 Math-2 dataset. The mean average precision (MAP), and precisions at ten (P@10), and five (P@5) are reported for queries formulated using Presentation (PMath), and Content MathML (CMath), a combination of both (PCMath), and LaTeX. Two different relevance judgement levels of ≥ 1 (partially relevant), and ≥ 3 (relevant) were used to compute the measures. Number between slashes (/·/) is our rank among all teams.

Measure	Level	PMath	CMath	PCMath	LaTeX
MAP	3	0.3073	**0.3630 /1/**	0.3594	0.3357
P@10	3	0.3040	**0.3520 /1/**	0.3480	0.3380
P@5	3	0.5120	**0.5680 /1/**	0.5560	0.5400
MAP	1	0.2557	**0.2807 /2/**	0.2799	0.2747
P@10	1	0.5020	0.5440	**0.5520 /1/**	0.5400
P@5	1	0.8440	**0.8720 /2/**	0.8640	0.8480

on industrial-strength full-text search engine Apache Lucene. The system allows low-latency responses even on the big math corpora as proved by its deployment in EuDML.

The speed of indexing and response latency of MIR will be further increased by the migration of MIaS from Apache Lucene to the distributed full-text search engine ElasticSearch[14]. The idea of indexing structures rather than terms can be generalized from mathematical formulae to semi-structured text. Reordering the operands of associative operators is only a simple transformation. For example, to convert $\sqrt[n]{a}$, and $a^{1/n}$ to a single canonical representation, a general computer algebra system (CAS) can be used. We experiment [17] with improving the vector space representations of document passages, aiming to add support for mathematics in the future. Embeddings can also be computed for equations [9] now, which presents new possibilities of using language modeling for the semantic segmentation of STEM articles, and weighting the segments [17]. Grasping the meaning of mathematical formulae is crucial: content is king.

Acknowledgements We gratefully acknowledge the support by the European Union under the FP7-CIP program, project 250,503 (EuDML), and by the ASCR under the Information Society R&D program, project 1ET200190513 (DML-CZ). We also sincerely thank three anonymous reviewers for their insightful comments.

REFERENCES

[1] Akiko Aizawa, Michael Kohlhase, and Iadh Ounis. 2013. NTCIR-10 Math Pilot Task Overview. In *Proc. of the 10th NTCIR Conference*. NII, Tokyo, Japan, 654–661.

[2] Akiko Aizawa, Michael Kohlhase, Iadh Ounis, and Moritz Schubotz. 2014. NTCIR-11 Math-2 Task Overview. In *Proc. of the 11th NTCIR Conference on Evaluation of Information Access Technologies*. Noriko Kando and Kazuaki Kishida, (Eds.) NII, Tokyo, Japan, 88–98.

[3] Dominique Archambault and Victor Moço. 2006. Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations. In *Computers Helping People with Special Needs*. Lecture Notes in Computer Science. Vol. 4061. Klaus Miesenberger, Joachim Klaus, Wolfgang Zagler, and Arthur Karshmer, (Eds.) Springer Berlin / Heidelberg, 1191–1198. DOI: 10.1007/11788713_172.

[4] Josef B. Baker, Alan P. Sexton, and Volker Sorge. 2012. MaxTract: Converting PDF to LaTeX, MathML and Text. In *AISC/DML/MKM/Calculemus* (Lecture Notes in Computer Science). Johan Jeuring et al., (Eds.) Vol. 7362. Springer, 422–426. ISBN: 978-3-642-31373-8. DOI: 10.1007/978-3-642-31374-5_29.

[5] Andrzej Białecki, Robert Muir, and Grant Ingersoll. 2012. Apache Lucene 4. In *SIGIR 2012 Workshop on Open Source Information Retrieval*, 17.

[6] Davide Cervone. 2012. MathJax: a platform for mathematics on the Web. *Notices of the AMS*, 59, 2, 312–316.

[7] David Formánek, Martin Líška, Michal Růžička, and Petr Sojka. 2012. Normalization of Digital Mathematics Library Content. In *Joint Proc. of the 24th OpenMath Workshop, the 7th Workshop on Mathematical User Interfaces (MathUI), and the Work in Progress Section of the Conference on Intelligent Computer Mathematics* (CEUR Workshop Proceedings) number 921. (Bremen, Germany, July 9–13, 2012). James Davenport, Johan Jeuring, Christoph Lange, and Paul Libbrecht, (Eds.) http://ceur-ws.org/Vol-921/wip-05.pdf. Aachen, 91–103.

[8] Michael Kohlhase et al. 2008. MathWebSearch 0.4, a semantic search engine for mathematics. *Manuscript at http://mathweb. org/projects/mws/pubs/mkm08.pdf*.

[9] Kriste Krstovski and David M. Blei. 2018. Equation Embeddings. *ArXiv e-prints*, (Mar. 2018). arXiv: 1803.09123 [stat.ML].

[10] Martin Líška, Petr Sojka, and Michal Růžička. 2015. Combining Text and Formula Queries in Math Information Retrieval: Evaluation of Query Results Merging Strategies. In *Proceedings of the First International Workshop on Novel Web Search Interfaces and Systems* (NWSearch '15). ACM. ACM, Melbourne, Australia, 7–9. ISBN: 978-1-4503-3789-2. DOI: 10.1145/2810355.2810359. http://doi.acm.org/10.1145/2810355.2810359.

[11] Martin Líška, Petr Sojka, and Michal Růžička. 2014. Math Indexer and Searcher Web Interface: Towards Fulfillment of Mathematicians' Information Needs. In *Intelligent Computer Mathematics CICM 2014. Proceedings of Calculemus, DML, MKM, and Systems and Projects*. Stephen M. Watt et al., (Eds.) Springer International Publishing Switzerland, Zurich, 444–448. ISBN: 978-3-319-08434-3. DOI: 10.1007/978-3-319-08434-3_36.

[12] Martin Líška, Petr Sojka, and Michal Růžička. 2013. Similarity Search for Mathematics: Masaryk University team at the NTCIR-10 Math Task. In *Proc. of the 10th NTCIR Conference on Evaluation of Information Access Technologies*. Noriko Kando and Kazuaki Kishida, (Eds.) NII, Tokyo, Japan, Tokyo, 686–691. ISBN: 978-4-86049-062-1.

[13] Martin Líška, Petr Sojka, Michal Růžička, and Petr Mravec. 2011. Web Interface and Collection for Mathematical Retrieval: WebMIaS and MREC. In *Towards a Digital Mathematics Library. Bertinoro, Italy, July 20–21st, 2011*. Petr Sojka and Thierry Bouche, (Eds.) http://hdl.handle.net/10338.dmlcz/702604. Masaryk University, Bertinoro, Italy, (July 2011), 77–84. ISBN: 978-80-210-5542-1.

[14] Michal Růžička. 2017. *Math Information Retrieval for Digital Libraries*. Dissertation. Masaryk University, Faculty of Informatics, Brno, CZ. https://is.muni.cz/th/pxz4q/?lang=en.

[15] Michal Růžička, Petr Sojka, and Martin Líška. 2016. Math Indexer and Searcher under the Hood: Fine-tuning Query Expansion and Unification Strategies. In *Proc. of the 12th NTCIR Conference on Evaluation of Information Access Technologies*. Noriko Kando, Tetsuya Sakai, and Mark Sanderson, (Eds.) NII Tokyo, 331–337. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings12/pdf/ntcir/MathIR/05-NTCIR12-MathIR-RuzickaM.pdf.

[16] Michal Růžička, Petr Sojka, and Martin Líška. 2014. Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy. In *Proc. of the 11th NTCIR Conference on Evaluation of Information Access Technologies*. Hideo Joho and Kazuaki Kishida, (Eds.) https://is.muni.cz/auth/publication/1201956/en. NII, Tokyo, Japan, (Dec. 2014), 127–134.

[17] Jan Rygl, Petr Sojka, Michal Růžička, and Radim Řehůřek. 2016. ScaleText: The Design of a Scalable, Adaptable and User-Friendly Document System for Similarity Searches: Digging for Nuggets of Wisdom in Text. eng. In *Proc. of the 10th Workshop on Recent Advances in Slavonic NLP, RASLAN 2016*. Aleš Horák, Pavel Rychlý, and Adam Rambousek, (Eds.) Tribun EU, Brno, 79–87. ISBN: 978-80-263-1095-2. https://nlp.fi.muni.cz/raslan/2016/paper08-Rygl_Sojka_etal.pdf.

[18] Petr Sojka, (Ed.) *Towards a Digital Mathematics Library*. Birmingham, UK, (July 2008). Masaryk University. ISBN: 978-80-210-4658-0. http://dml.cz/dmlcz/702564.

[19] Petr Sojka and Martin Líška. 2011. Indexing and Searching Mathematics in Digital Libraries – Architecture, Design and Scalability Issues. In *Intelligent Computer Mathematics. Proceedings of 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011* (Lecture Notes in Artificial Intelligence, LNAI). James H. Davenport, William M. Farmer, Josef Urban, and Florian Rabe, (Eds.) Vol. 6824. Springer-Verlag, Bertinoro, Italy, (July 2011), 228–243. DOI: 10.1007/978-3-642-22673-1_16.

[20] Heinrich Stamerjohanns et al. 2010. Transforming Large Collections of Scientific Publications to XML. *Mathematics in Computer Science*, 3, 299–307, 3. ISSN: 1661-8270. DOI: 10.1007/s11786-010-0024-7.

[21] Masakazu Suzuki et al. 2003. INFTY — An Integrated OCR System for Mathematical Documents. In *Proc. of ACM Symposium on Document Engineering 2003*. C. Vanoirbeek, C. Roisin, and E. Munson, (Eds.) ACM, Grenoble, France, 95–104.

[22] Richard Zanibbi et al. 2016. NTCIR-12 MathIR task overview. In *Proc. of the 12th NTCIR Conference on Evaluation of Information Access Technologies*. NII Tokyo, 299–308.

[14] https://elastic.co