Building Corpora of Technical Texts
Approaches and Tools

Petr Sojka, Martin Líška, and Michal Růžička

Faculty of Informatics, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
sojka@fi.muni.cz, martin.liski@mail.muni.cz, mruzicka@mail.muni.cz

Abstract. Building corpora of technical texts in Science, Technology, Engineering, and Mathematics (STEM) domain has its specific needs, especially the handling of mathematical formulae. In particular, there is no widely accepted format to represent and handle math.
We present an approach based on multiple representations of mathematical formulae that has been used for math retrieval, similarity and clustering of mathematical corpus. We provide an overview of our toolset, summarize our experiments to date and propose further research directions and approaches.

Key words: mathematical corpora; information retrieval of mathematics; representation of mathematical formulae; math search and indexing normalization of MathML; canonicalization

1 Introduction

Leading research in empirical linguistics builds on the large (e.g. web-scale) corpora such as those created by Google (Google Books Corpus, Google Scholar) or by the Sketch Engine (TenTen Corpora). Such corpora allow for natural language processing (NLP) of a new quality level to solve such tasks as more relevant information retrieval, document clustering, classification and similarity, thesauri and ontology building, better word sense disambiguation, machine translation and many others. However, in these research mainstream activities, minority languages or domain specifics are neglected. Such a neglected 'language' is the language of mathematics – typical in Science, Technology, Engineering, and Mathematics (STEM) documents.

Mainstream NLP workflow for building corpora starts with tokenization, which is usually not aware of mathematical formulae or equations. Math is usually supported neither by optical character recognition (OCR) tools, nor by applications that generate PDF or (X)HTML. The use and representation of math on the web is far from settled. As a consequence, no mainstream tools support this niche market of 'the Queen of sciences'.

In previous projects that involved building Digital Mathematics Libraries (DML) such as DML-CZ [1] and EuDML [2], we had to deal with the fact that NLP corpora tools were unable to handle corpora of math texts, let alone build

Aleš Horák, Pavel Rychlý (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2011, pp. 69–79, 2011.

them. We therefore devised some tools for adequate support of mathematical formulae in NLP and information retrieval (IR) tasks. Proper semantic and math-aware representation is a necessary prerequisite for efficient and effective NLP processing of STEM corpora.

This task involved as the first step the design of math formulae representation (Section 2). Then, to build mathematical corpora, we had to preprocess and normalize heterogenious inputs (Section 3) into this new representation. It was also necessary to design ways of math retrieval (index and search are crucial, cf. Section 4). Our aim is to support math-aware document clustering, similarity and disambiguation (Section 5). We summarize our findings in the Section 6.

2 Math Representations

Mathematicians and other authors of STEM documents encode quantities and relations using formulae and equations in compact, often two-dimensional, notation. These objects have to be represented in unique way in the global STEM document handling system.

There are numerous ways of notating the same mathematical object, that has evolved in some geographical location or language. This is an example of different notations for a *binomial coefficient*:

$$\binom{n}{r} = \frac{n!}{r!(n-r)!} = {}_nC_r = {}^nC_r = C(n,r)$$

When searching STEM documents, it should be possible to find the same objects within a corpus, and assign them the same representation even though authors have used different notation. As is the case with text handling, where words with the same meaning are treated and indexed the same, formulae require the same treatment.

The matter is complicated by the fact that there are different formats for handling mathematics: TEX, MathML, OpenMath, etc.

2.1 TEX

Authors prefer the compact and logical notation of **TEX**. The American Mathematical Society (AMS) extended standard plain TEX and **LATEX** notation with AMS packages (so called **AMSLATEX**) for commutative diagrams, aligned equations, etc. The namespace of AMSLATEX macros is nowadays the de facto standard for the typesetting of mathematical documents, and this namespace is also supported in ithe metadata of DML-CZ (only this namespace is allowed and supported there, e.g. by conversion to MathML).

TEX math notation is in such demand that even for **Word** there is plug-in by Design Science that allows entering the formulae in TEX notation. This is much quicker, and more convenient than choosing the symbols from numerous menus and symbol tables. Nevertheless, using TEX notation for indexing

purposes is a disaster, as an example of **LaTeXSearch** application by Springer (`http://latexsearch.com`) shows. Authors are so creative in macroexpansion use or TEX language formatting that different notations cannot be coped with by simple string similarity. A formulae structure and other types of similarity has to be used in formulae representation for similarity computation. For this, the tree structure of XML (MathML) is better, as it is understood by the majority of math-aware software developers.

2.2 MathML

In the world of applications and software interfaces, MathML usually wins, as it is supported by W3C and AMS. TEX's macro namespace extensibility is a nightmare to support by software without the full TEX macroexpansion complex engine, and here MathML clearly wins. MathML DTD allows easy formulae validation and processing with XML tools. There are even recently developed portable tools like **MathJax**, a JavaScript library that displays mathematics in web browsers, supporting both LATEX and MathML markup as it attempts to convert LATEX on-the-fly into appropriate markup language–HTML or MathML.

2.3 Set of M-terms

A mathematical document contains mathematical formulae, which are integral to the content of the document. As mentioned in the previous sections, these formulae are usually represented in TEX if authored by humans, or in MathML (presentation or mixed content-presentation) if produced by machines.

To be able to search for such structural information using a fulltext indexing approach as in the Math Indexer and Searcher (**MIaS**) system [3,4], a convenient representation needs to be selected. This representation needs to be a trade-off between the TEX powered authors' part of the world and machine-friendly, preserve-as-most-information-as-possible, structural and semantic notation such as Content MathML. In MIaS system we opted for Presentation MathML as it stands, we find, exactly halfway. It is relatively easy to obtain by converting the author's TEX markup and it still holds the necessary structural information for machine processing. It is still easily extensible by Content MathML trees capturing the formulae semantics.

Such mathematical markup needs to be preprocessed before the indexing. This is mainly to accommodate the best user search experience as possible. For each formula in the text, the system produces several representations which are stored in the index and are searchable in the same way as regular textual terms. These are called *M-terms*.

M-terms are translated from XML to a linear string form. In this form they are stored by the indexing core. This representation omits any XML markup that would be redundant in such a form, such as start and end tags, and replaces it with brackets to prevent ambiguity. Also most of the attributes setting the visual behaviour of the expression can be left out, since it does not hold any

information pertaining to the meaning of the formula. This representation can be further compacted by substituting tag names for single characters to decrease storage space requirements.

For example, simple expression $a^2 + b$ in its XML form

```
<math>
  <mrow>
    <msup><mi>a</mi><mn>2</mn></msup>
    <mo>+</mo>
    <mi>b</mi>
  </mrow>
</math>
```

is translated to the linear form `mrow(msup(mi(a)mn(2))mo(+)mi(b))` and based on a custom tag name dictionary, where `mrow = R`; `msup = J`; `mi = I`; `mn = N` and `mo = O`. This is further compacted to `R(J(I(a)N(2))O(+)I(b))`. A set of sub-M-terms is generated for each input formula. It consists of subformula-weight pairs. For this particular expression, it is:

```
{
(mi(a),0.08166666),
(mn(2),0.08166666),
(msup(mi(a)mn(2)),0.11666667),
(mo(+),0.11666667),
(mi(b),0.11666667),
(mrow(mi(b)mo(+)msup(mi(a)mn(2))),0.16666667),
(msup(mi(1)mn(2)),0.093333334),
(mrow(mi(1)mo(+)msup(mi(2)mn(2))),0.13333334),
(msup(mi(a)mn(¶)),0.058333334),
(mrow(mi(b)mo(+)msup(mi(a)mn(¶))),0.083333336),
(msup(mi(1)mn(¶)),0.046666667),
(mrow(mi(1)mo(+)msup(mi(2)mn(¶))),0.06666667)
}
```

These formulae are derived from the original one and their level of similarity is expressed by the weight factor.

This representation not only grabs the structural similarity of mathematical formulae, it also copes with different variable names, and with mathematical properties of operators (commutativity). As such, representation of formulae by an M-term set with weights is directly useable for indexing or for document similarity computations.

To provide these and other uses of this representation, we have set up a RESTful web service, where for each input formula one can get a set of M-terms as they would be indexed in the MIaS system. An example of use can be found here:

`http://aura.fi.muni.cz:8085/mias4gensim/mathprocess?mterm=<math><mrow><mi>a</mi><mo>+</mo><mi>b</mi></mrow></math>`

3 Mathematical Corpora

3.1 Normalization

When building mathematical corpora using MathML as a language for mathematical formulae preservation, it emerges that it is very useful to process and normalize MathML that is being stored. It is necessary as one mathematical formula can be encoded in MathML in different forms – using different sequences of characters in the source code – but its meaning is the same.

For example, the formula $x^2 + y^2$ can be encoded in MathML in the form:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
    <msup>
        <mi>x</mi>
        <mn>2</mn>
    </msup>
    <mo>+</mo>
    <msup>
        <mi>y</mi>
        <mn>2</mn>
    </msup>
</math>
```

But some other author can use this form:

```
<math xmlns="http://www.w3.org/1998/Math/MathML">
  <mrow>
    <msup>
        <mi>x</mi><mn>2</mn></msup>
    <mo>+</mo>
    <msup>
        <mi>y</mi><mn>2</mn></msup>
  </mrow>
</math>
```

To be able to find documents that contains our formula in any of these codings we need one normalized form that will be stored in the index. Subsequently, any query for this formula in any coding has to be transformed to the normalized form at the beginning.

Moreover, examples of documents from the real world (PubMed Central digital library workflow) show that validation of MathML source codes is not enough. Elbow et al. [5] demonstrate a well-known fact that current authors' main target is print output – consequently one can find MathML fragment `<mml:mn>7</mml:mn><mml:mn>5</mml:mn>` as source code of the number '75' for example. These anomalies have to be sorted out before publishing and indexing in a repository.

For the semantically same formalae there exist infinitely many ways of representing them in MathML. For NLP handling it would be convenient to have one *canonical* representation of a formulae.

3.2 Canonicalization

Proper MathML normalization (canonicalization) is not easy given that MathML is a very complex markup language. Some existing tools we have tested fail when run over a set of MathML test documents [6] that were designed to cover a wide range of MathML features.

Our approach to MathML normalization has so far involved a trial use of **UMCL** (Universal Maths Conversion Library; `http://inova.ufr-info-p6.jussieu.fr/maths/umcl`). [7,8] The main purpose of the UMCL tool set is to enable transcription of the MathML formulae to Braille national codes. Related to this task is also the need for MathML formulae unification. UMCL transformation of the MathML to Canonical MathML is carried out using a set of XSL stylesheets [9].

With minor modifications, the UMCL MathML transformation was used in the WebMIaS interface [10] (see Section 4) that can be used to search over our MREC corpus (see Section 3.3). This showed benefits of formulae normalization in practice – search form $x^2 + y^2$ formula using the first form of MathML code from the previous section found no results. However, for the second form of MathML – the form that is the result of UMCL XSL transformation from the first form – there were 36,817 hits in MREC corpus version 2011.4.

Unfortunately, the MathML canonicalization module of the UMCL tool set is not as powerful as we thought at the beginning. Using the W3C MathML Test Suite mentioned in the previous section, some weak points in UMCL normalization process have been identified. Among other things, there are problems with MathML tags like 'mphantom', 'mfenced', 'mglyphe', 'mmultiscripts', 'mover' and 'mstyle' that are not properly converted. Furthermore, attributes of MathML elements are not reported in the UMCL canonicalized MathML.

These problems were consulted with UMCL developers but no fast and clear solution seems to be available. Due to these problems, UMCL in the current version does not seem to be directly applicable to MREC corpus and further research in this area is definitely necessary.

3.3 Corpus MREC

To provide a test platform for mathematical search tools, we are building a corpus of mathematical texts. We call this corpus MREC.

MREC is based on arXMLiv [11] – a project of Michael Kohlhase's group at Jacobs University Bremen. arXMLiv documents came from `arXiv.org` but have been translated to XML by arXMLiv project. These documents cover different STEM areas – Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance and Statistics.

However, MREC is not an exact copy of the arXMLiv content. MREC contains just a subset of the arXMLiv – arXMLiv puts transformed documents into several classes – successful, complete with errors and incomplete, depending on the results of the transformations. MREC contains papers from conversion classes, successful and complete with errors (missing macros) – see Table 1. We

have collected 439,423 documents in well-formed XHTML, containing mathematical formulae in valid MathML.

Table 1. Documents collected from arXMLiv

arXMLiv transformation result class	Quantity
successful (no problem)	65,874
successful (warning)	291,879
complete with errors (missing macros)	81,670
All documents	439,423

Moreover, there were several modifications of the files that from our point of view were necessary in order to make the documents well-formed and valid. These modifications include removing unnecessary attributes, namespace proxies, 'div' elements nested in 'span' elements and so on.

MREC consists of well-formed XHTML documents. MathML is used for representation of mathematical formulae.

Although MREC is under constant development, it is necessary for both archive and comparison purposes to produce a stable release versions. For this reason, there are several version of MREC corpora available at `http://nlp.fi.muni.cz/projekty/eudml/MREC/`.

The first public version of MREC, version 2011.3.324, consists of 324,060 documents. The resulting corpus size was 53 GB uncompressed, 6.7 GB compressed. Documents contained 112,055,559 formulae in total, of which 2,129,261,646 mathematical expressions were indexed. The resulting index size was approximately 45 GB.

The newer version of MREC, version 2011.4.439, consists of 439,423 scientific documents containing 158,106,118 mathematical formulae. 2,910,314,146 expressions were indexed and the resulting size of the index is 63 GB. The sizes of uncompressed and compressed corpora are 124 GB and 15 GB, respectively.

4 Math Retrieval

Searching functionality is nowadays a key form of getting orientated in the vast amount of information "out there" and obtaining the information we seek. Just as websites providing special content such as images and videos enable searching for these tokens, portals providing mathematical content such as EuDML [12] should also be able to search for the formulae.

In our view, the optimal way of doing so is to provide a simple Google-like interface where one can pose mathematical and textual query tokens one alongside the other. Search results returned to a textual query can then be finely constrained by adding a formula to the query and, in fact, vice-versa. We present this approach in the WebMIaS interface [10].

For example, by posting a simple query $x^2 + y^2$ in our web interface, the system returns 36,817 results. Addition of one more keyword *Euclid* reduces the number of results to only 97 – all of them contain this textual term. Conversely, searching only for *Euclid* returns 848 results and by adding $x^2 + y^2$ expression, we get the same 97 matches (MREC 2011.4.439).

To implement math-aware IR system in addition to the web-interface it was necessary to create an index to be consulted during query evaluation. We use our M-term representation for this, as described in detail in [3,4]. We have evaluated the system's speed. As is shown in Table 2, the performance of the MIaS system scales linearly. This gives feasible response times even for our billions of indexed subformulae.

Table 2. Indexing scalability test results (run on 448 GiB RAM, eight 8-core 64bit processors Intel Xeon^{TM} X7560 2.26 GHz driven machine).

# Docs	Input formulae	Indexed formulae	run-time [ms]	CPU time [ms]
10,000	3,406,068	64,008,762	2,145,063	2,102,770
50,000	18,037,842	333,716,261	11,382,709	10,871,500
100,000	36,328,126	670,335,243	23,066,679	21,992,100
200,000	72,030,095	1,326,514,082	46,143,472	44,006,180
300,000	108,786,856	2,005,488,153	71,865,018	66,998,550
350,000	125,974,221	2,318,482,748	83,199,724	77,886,160
439,423	158,106,118	2,910,314,146	104,829,757	97,393,301

5 Further Research Directions in Math Similarity, Clustering and Disambiguation

In mathematics, Mathematical Subject Classification (MSC) is used by most journals today, being supported and developed by both Mathematical Reviews (MR) and Zentralblatt Math (ZMath). Our research so far [13] has shown that machine-learned classification and similarity tasks are tractable to be supported by DMLs. However, previous research paid very little attention to the representation of mathematics. Either textual tokens alone were used, or the formulae were split into variables, constants and operators, and used in a 'bag of words' for documents. Such representation is insufficient given that it does not convey the structure of formulae, and neither does it pay attention to semantically similar formulae (e.g. written in different variable names, sorted differently as $a + b$ vs. $b + a$, etc.).

We are currently using the **Gensim** [14] system to evaluate the possibility of using M-terms instead of the usual tokenization and comparing the effects this new representation has on similarity and clustering improvements over non math-aware representations. We believe that M-term representation will

significantly improve the quality of document similarity metrics computed by Gensim.

Further improvements could be achieved by employing cutting edge results on semantic disambiguation. Symbol f might play the rôle (have meaning) of a variable, functional, (linear) function, and potentially a dozen other meanings. To have a greater relevance to searching and better document clustering, even mathematical formulae should be disambiguated at this level, as authors are usually reluctant to do so in the (LaTeX) sources or in Content MathML. Our representation method is easily inclusive with respect to these refinements – one just needs to add a notation and weighting for similarity of new terms representing Content MathML (semantics).

There were attempts to bring NLP approaches to math corpora handling recently [15,16]. The most consistent problem remains the high degree of ambiguity in mathematical formulae and nonexistence of tagged disambiguated math data.

There is a promising approach to distinguishing roles of words (math tokens) which depends on the contexts of use in a corpus called *LDA-frames* [17]. It uses statistics to distinguish different roles based on different structural patterns of word usage in corpora. We are considering the possibility of using a fuzzy version of Formal Concept Analysis (FCA) [18] to identify the rôles of math tokens in formulae, and in combination with LDA-frames to disambiguate them.

6 Summary and Conclusions

In this paper, we have identified and described the problems we have faced when building nontrivial corpora of STEM documents MREC. We have suggested M-term representation for math-aware indexing and similarity computations. We have reported current results in imath-aware indexing and isearching. We have discusses future research directions towards fully fledged math-aware corpora processing like math-aware document similarity or disambiguation of math symbols in formulae.

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 (Information and Communications Technologies Policy Support Programme, "Open access to scientific information", Grant Agreement No. 250503).

References

1. Bartošek, M., Lhoták, M., Rákosník, J., Sojka, P., Šárfy, M.: DML-CZ: The Objectives and the First Steps. In: Borwein, J., Rocha, E.M., Rodrigues, J.F., eds.: CMDE 2006: Communicating Mathematics in the Digital Era. A. K. Peters, MA, USA (2008) 69–79.

2. Borbinha, J., Bouche, T., Nowiński, A., Sojka, P.: Project EuDML—A First Year Demonstration. In: Davenport, J.H., Farmer, W.M., Urban, J., Rabe, F., eds.: Intelligent Computer Mathematics. Proceedings of 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011. Volume 6824 of Lecture Notes in Artificial Intelligence, LNAI., Berlin, Germany, Springer-Verlag (2011) 281–284 http://dx.doi.org/10.1007/978-3-642-22673-1_21.
3. Sojka, P., Líška, M.: Indexing and Searching Mathematics in Digital Libraries – Architecture, Design and Scalability Issues. In: Davenport, J.H., Farmer, W.M., Urban, J., Rabe, F., eds.: Intelligent Computer Mathematics. Proceedings of 18th Symposium, Calculemus 2011, and 10th International Conference, MKM 2011. Volume 6824 of Lecture Notes in Artificial Intelligence, LNAI., Berlin, Germany, Springer-Verlag (2011) 228–243 http://dx.doi.org/10.1007/978-3-642-22673-1_16.
4. Sojka, P., Líška, M.: The Art of Mathematics Retrieval. In: Proceedings of the ACM Conference on Document Engineering, DocEng 2011, Mountain View, CA, Association of Computing Machinery (2011) 57–60 http://doi.acm.org/10.1145/2034691.2034703.
5. Elbow, A., Krick, B., Kelly, L.: PMC Tagging Guidelines: A case study in normalization. In: Proceedings of the Journal Article Tag Suite Conference 2011, National Center for Biotechnology Information (2011) http://www.ncbi.nlm.nih.gov/books/NBK62090/#elbow-S8.
6. W3C: MathML Test Suite (2010) http://www.w3.org/Math/testsuite/.
7. Archambault, D., Stöger, B., Batušić, M., Fahrengruber, C., Miesenberger, K.: A software model to support collaborative mathematical work between Braille and sighted users. In: Proceedings of the ASSETS 2007 Conference (9th International ACM SIGACCESS Conference on Computers and Accessibility), ACM (2007) 115–122 http://portal.acm.org/ft_gateway.cfm?id=1296864&type=pdf.
8. Archambault, D., Berger, F., Moço, V.: Overview of the "Universal Maths Conversion Library". In: Pruski, A., Knops, H., eds.: Assistive Technology: From Virtuality to Reality: Proceedings of 8th European Conference for the Advancement of Assistive Technology in Europe AAATE 2005, Lille, France, Amsterdam, The Netherlands, IOS Press (2005) 256–260.
9. Archambault, D., Moço, V.: Canonical MathML to Simplify Conversion of MathML to Braille Mathematical Notations. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A., eds.: Computers Helping People with Special Needs. Volume 4061 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 1191–1198 http://dx.doi.org/10.1007/11788713_172.
10. Líška, M., Sojka, P., Růžička, M., Mravec, P.: Web Interface and Collection for Mathematical Retrieval. In: Sojka, P., Bouche, T., eds.: Proceedings of DML 2011, Bertinoro, Italy, Masaryk University (2011) 77–84 http://www.fi.muni.cz/~sojka/dml-2011-program.html.
11. Stamerjohanns, H., Kohlhase, M., Ginev, D., David, C., Miller, B.: Transforming Large Collections of Scientific Publications to XML. Mathematics in Computer Science **3** (2010) 299–307 http://dx.doi.org/10.1007/s11786-010-0024-7.
12. Sylwestrzak, W., Borbinha, J., Bouche, T., Nowiński, A., Sojka, P.: EuDML—Towards the European Digital Mathematics Library. In: Sojka, P., ed.: Proceedings of DML 2010, Paris, France, Masaryk University (2010) 11–24 http://dml.cz/dmlcz/702569.
13. Řehůřek, R., Sojka, P.: Automated Classification and Categorization of Mathematical Knowledge. In: Autexier, S., Campbell, J., Rubio, J., Sorge, V., Suzuki, M., Wiedijk, F., eds.: Intelligent Computer Mathematics—Proceedings of 7th International Con-

ference on Mathematical Knowledge Management MKM 2008. Volume 5144 of Lecture Notes in Computer Science LNCS/LNAI., Berlin, Heidelberg, Springer-Verlag (2008) 543–557.
14. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks, Valletta, Malta, ELRA (2010) 45–50 http://is.muni.cz/publication/884893/en, software available at http://nlp.fi.muni.cz/projekty/gensim.
15. Anca, Ş.: Natural Language and Mathematics Processing for Applicable Theorem Search. Master's thesis, Jacobs University, Bremen (2009) https://svn.eecs.jacobs-university.de/svn/eecs/archive/msc-2009/aanca.pdf.
16. Grigore, M., Wolska, M., Kohlhase, M.: Towards context-based disambiguation of mathematical expressions. Math-for-Industry Lecture Note Series **22** (2009) 262–271.
17. Materna, J.: LDA-Frames: an Unsupervised Approach to Generating Semantic Frames. In: Proceedings of CICLING 2012, Springer-Verlag (2012) 12 pages, submitted.
18. Bělohlávek, R.: Concept lattices and order in fuzzy logic. Annals of Pure and Applied Logic **128**(1–3) (2004) 277–298.