Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy

Michal Růžička
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
mruzicka@mail.muni.cz

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
martin.liski@mail.muni.cz

ABSTRACT

This paper describes and summarizes experience of Masaryk University Math Information Retrieval team (MIRMU) with the mathematical search developed and performed for the NTCIR-11 Math-2 Task. Our approach is the similarity search based on canonicalized MathML and second generation of scalable full text search engine Math Indexer and Searcher (MIaS) with attested state-of-the-art information retrieval techniques like query expansion. The capability of MIaS system in terms of math query notation, normalization and combining math with textual query tokens was deployed by submitting multiple runs with four query notations provided, and with results merged from multiple queries. The analysis of the evaluation results shows that the system performs best using TEX queries that are translated and canonicalized to Content MathML, where MIaS ranked as #1 for all metrics returning very relevant results.

Team Name

MIRMU (Math Information Retrieval at Masaryk University)

Subtasks

Math-2 Main Task (English) and optional Math-2 Wikipedia Subtask (English)

Keywords

math, search, similarity search, math information retrieval, MIR, MIaS, evaluation, math representation and indexing, math canonicalization, MathML

> "Study the past if you would define the future."
> Confucius

1. HISTORICAL REMARKS AND MOTIVATION

In the beginning, there was a dream of a global Digital Mathematical Library (DML). Computer Science and Engineering advances showed the possibilities of having all previous knowledge at fingertips of mathematicians at the beginning of millennium. Search became ubiquitous, and has been used as a gate to the digitally stored knowledge in the digital libraries and on the web.

We started to design solutions for building DMLs in 2005 when the Czech DML (DML-CZ) was conceived. [15] We began to develop *math-aware* workflows to handle digital content in the form of full texts of hundreds of thousands of scientific papers. We have realized that math content, and that holds for the whole STEM domain, is specific by the presence of mathematical formulae.

Structured mathematical notation is irreplaceable part of mathematical vernacular, and it should be supported when optically recognizing, storing, representing, indexing, querying, filtering, mining, and linking exponentially growing mathematical literature. To cover these specifics as topics of research, specific research forum, workshop series *Towards a Digital Mathematics Library* [8], has been set up. In the DML proceedings [16] there are already papers tackling math-aware problems as search [13] and other related issues [22]. It become clear that the whole math-aware workflow [17] has to be researched.

For the European Digital Mathematics Library (EuDML), where we have been responsible for searching component, we designed Math Indexer and Searcher (MIaS) system [25] as probably the first production quality math-aware indexing system indexing hundreds of thousands of documents.

Math notation needs to be specifically supported by the Information Retrieval tools: establishment of Math Information Retrieval (MIR) research field was necessary. MIaS was one of two systems at Math IR Happening [19], which took place at CICM 2012 conference.

Finally, MIR research attracted several research groups as Pilot Math Task has been set up at NTCIR-10. MIaS used for the first time MathML canonicalization module and Content MathML indexing. [11] And for this year, MIaS has been enhanced with query expansion strategies and better canonicalization. [14, 2]

The paper is structured as follows: In the Section 2 we give an overview of our approach used in our MIaS system. Section 3 describes runs that MIRMU team submitted for NTCIR Math Task, with scripts used for automation of querying. In the Section 4 our indexing statistics are revealed. Results achieved are discussed in the Section 5. We conclude with summary and directions of further research and developments in the Section 6.

> "God is in the details."
> Ludwig Mies van der Rohe

2. OVERVIEW OF MATH INDEXING SOLUTION MIAS

Main design principles of MIaS system were set aligned with the awaited deployment in a large DMLs like EuDML, or even Global DML [5]. We designed *open*, *scalable*, *math formulae-aware* system for *ranked document* information retrieval, that goes together with leading edge information retrieval systems based on textual keywords and collocations.

The main design questions of MIR system like MIaS were:

1. How to collect data and pre-process them into uniform representation?
2. How to make unambiguous canonical representation of the semantically same formulae and entities?
3. How to index the data allowing quick relevance evaluation to the query?
4. How to rank and sort the documents found?
5. What user interface for querying and presenting results should be used?

Preprocessing

Essence of the problem is to cope with *different*, heterogeneous formats and representations of structured mathematical formulae. Even though most math content is primarily written in some flavour of TEX markup, the markup and levels of abstractions are very different. For some documents, only PDFs are available. That means output of different software for math optical character recognition (OCR), tools for conversion of born-digital PDF, or even various TEX macropackages needs to be converted and normalized into a structural tree form of math. We use MathML standard [3] for this, ideally in both Presentation and Content MathML shapes. Tools as Infty Reader [24], MaxTract [4], Tralics [7] and LaTeXML [23] are used for these tasks in our MIR system.

Canonicalization

Having formulae expressed in MathML still allows plethora of ways representing the same formulae. To increase information retrieval metrics and precision and recall it is necessary to find a canonical representant for all semantically equivalent formulae. We have designed, implemented and continually improve a converter[1] for both Presentation and Content MathML for this task. [6]

The achievement of the full disambiguation of all elements in a formulae is a big future work task for which knowledge understanding would be needed, not mentioning flexibility of natural language to express same or slightly similar things.

Representation of Math for Indexing

Concepts of *similarity* and *distributional representations* are central in the design of MIaS[2]. Every formulae is represented in the index as a *set of weighted tokens (subformulae, features)* that grab both structure and content of indexed mathematical formulae. The weighting is computed via small set of rules reflecting similarity distance of indexed tokens to the original formulae: the more similar is token to the original (in size, variable naming, constants used, ...), the higher weighting score is stored in the index for a token. On average, currently the formulae representation is distributed over about 30 indexed weighted tokens. [21]

Document Ranking

Main performance metric used at NTCIR is *precision* (P@1, P@5, ...). It is hard to use global document rankings as PageRank in DMLs—thus importance of good 'local' document ranking, e.g. computation of relevance of query to matched documents, increases. Given the weighting is done at the indexing time, at the query time only summation of similar hits is done, which leads to the good responsiveness of the system and superb scalability.

User Interface

Acceptance of MIR system by conservative mathematicians depends also on the user interface. We offer WebMIaS[3] user interface [10], which allows input in both TEX and MathML and on-the-fly rendering for feedback and conversion. [12]

For a more detailed description of all parts of the system the reader is referred to [20, 21, 18, 9] and web page of our group `https://mir.fi.muni.cz/`.

> "Reason and free inquiry are the only effectual agents against error."
> Thomas Jefferson

3. AUTOMATIC QUERYING SCRIPTS

To query MIaS engine there is WebMIaS user interface [12], which offers also web services interface for query automation. We used it to automate query processing in a novel way this year.

Availability of the task inputs in the XML format supplemented by MIaS web service interface allowed us to fully automate the task data processing. The batch querying script read topic specifications from the particular XML file and constructed four different XML queries for the MIaS web service interface for each of the topics:

PMath query The query contained Presentation MathML representation of the query formulae together with text keywords.

CMath query This query was constructed in the very same way as the PMath query but using Content MathML representation of the formula instead of the Presentation MathML part.

PCMath query This query combines both Presentation and Content MathML, i.e. the query is constructed as a concatenation of the Presentation MathML from the PMath query and Content MathML from the CMath query plus the text keywords.

TEX The last query is similar to the previous ones but the TEX representation was used instead of the MathML. There was no modification of the TEX statement except for removing linebreaks (as well as removing any comments and per cent sings (%) protecting these linebreaks) in case of multiline TEX code. The `\qvar{...}` macros were transformed as described in the next paragraph. Finally, a single dollar sign (`$`) was added on both sides of the original statement to properly indicate TEX encoded part of the query to the MIaS system.

Handling Query Variables

MIaS system is designed not to depend on hints on variables from the users in the queries. In fact, these query variable hints are not supported by the system in the queries. Due to this fact we had to transform `\qvar{...}` markup to regular identifiers.

[1] `https://mir.fi.muni.cz/mathml-normalization/`

[2] `https://mir.fi.muni.cz/mias/`

[3] `https://mir.fi.muni.cz/webmias/`

Use of `\qvar` elements varied in different queries. Query number 1 contained named entities in the `\qvar` markup (`\qvar{square}`, `\qvar{phi}`), query number 2 contained plus sign in the `\qvar` markup (`\qvar{+}`), other queries were using single letter `\qvar` identifiers (such as `\qvar{L}`, `\qvar{k}` etc.).

For use of MIaS, `\qvar` markup was transformed to regular single letter identifiers simply by keeping only the first letter from the original `\qvar` content (i.e. `squere` → `s`, `+` → `+`, `L` → `L`) and the `\qvar` macro itself was removed.

The system keeps track of mapping of the original `\qvar` name to the single-letter substitute not to use the same single-letter substitute for two different `\qvar` names in a single formula. The next letter in alphabetical order is used as the substitute if the original one has already been used to represent a different `\qvar` name (i.e. `left` → `l`, `lowbound` ~~→ `l`~~ → `m`). However, this mechanism was not used in practise as NTCIR-11 Math-2 Task does not contain `\qvar` names colliding on the first letter within a single formula.

In the TEX queries, the single letter identifiers were inserted to the source code to the place of the original `\qvar` macro surrounded with single space. In the MathML queries, a new `mi` and `ci` element replaced the `\qvar` element in Presentation and Content markup respectively, i.e.

```
<mws:qvar xmlns:mws="http://search.mathweb.org/ns"
          name="square"/>

   ↓

<mi>s</mi>
```

However, the investigation of our results (see Section 5) indicates that our system could benefit from some kind of unification of the queried and indexed formulae. On the other hand, the unification should be performed by the system itself with no need of users' hints in the query. Proposal of a possible simple indexing-time unification is presented in Section 6.

Query Expansion

Combination of multiple formulae and multiple text keywords in one query used in NTCIR-11 Math Task seems to be more consistent with the real situation of a human using math-aware search engine: formulae are simply a different expression of keywords used to filter relevant documents from the whole database. They are complement instrument of the query specification to the keywords, not the opposite of them. The queries work best with formulae and keywords together.

The MIaS system supports this kind of queries natively. All the parameters are posted to the system in one text field—formulae are written in MathML or TEX notation with a dollar sign ($) added on both sides of the TEX formulae. Keywords, sometimes consisting of more than one word, were surrounded with single quotation mark (") to handle multi-word keywords as a single entity. Formulae and text keywords were separated by a single space.

Internally, we are expanding the original query further to increase recall on very specific queries with no or just minimal number of results found. More specifically, the original query consisting of k keywords and f formulae is used to generate a set of 'subqueries'. At first, the original query is used. Then subqueries are generated one by one removing the keywords from the query until the query consists of f formulae only. The rest of subqueries are generated with all the keywords but formulae are removed from the query one by one until the query consisting of k keywords only is reached.

An example of the complete 'subqueries' generation sequence for a query consisting of two formulae (denoted f_x) and three keywords (denoted k_y) is shown in Example 1.

subquery 1 (the original query):	f_1	f_2	k_1	k_2	k_3
subquery 2:	f_1	f_2	k_1	k_2	
subquery 3:	f_1	f_2	k_1		
subquery 4:	f_1	f_2			
subquery 5:	f_1		k_1	k_2	k_3
subquery 6:			k_1	k_2	k_3

Example 1: Complete sequence of subqueries derived from the original user's query

All the queries are one by one used to ask the system and the results lists of the subqueries are merged (see the next Section) to the final result list that is presented to the user.

Statistics of the relative number of results found using each of the subqueries in the MIaS most successful CMath run are shown in Figure 1. Every subquery was limited to at most 1,000 results as requested in the NTCIR-11 Math Task. The graph shows that the use of the original unmodified query usually resulted in much less than requested 1,000 results. The use of the results of multiple subqueries thus provides significantly more results that are (at least partially) relevant to the users' query.

Please note that the last subquery does not contain any formulae, i.e. subquery 6 in Example 1, is standard full text search keyword query with no involvement of mathematical elements whatsoever.

Please also note that this algorithm does not cover all the possible combinations of keywords and formulae as well as 'unreasonably' handle different formulae differently—in Example 1 formula f_1 is used in five subqueries in contrast to four uses of f_2. The simplification was used to keep the number of subqueries small enough to reach an acceptable response time even for interactive human users of the search system as the total number of subqueries would increase rapidly with the number of formulae and keywords in the query if all their possible combinations should be used.

Length of MIaS searches for the most successful CMath run are shown in Table 1. Cumulative total MIaS search time for all 50 queries in CMath run was 10.81 seconds. Cumulative totals for other three runs are comparable: PMath 12.01 s, PCMath 14.70 s and for TeX 19.83 s.

This kind of query expansion provide users with results on more general queries than the user originally posted. We consider this behavior useful especially for 'research' search as this shows the user wider context of the query that could possibly reveal new and unexpected connections and paths to follow in the research.

Merging of Results

Every subquery results in an ordered list of items with score[4] assigned to each of the results. However, these scores are only comparable within the context of the result list. That means that a result r_1 with score 0.25 from the subquery 1 is not necessarily more relevant to the subquery 1 than a result r_2 with score 0.15 from the subquery 2 even though $0.25 > 0.15$

[4] Measure of relevance to the query.

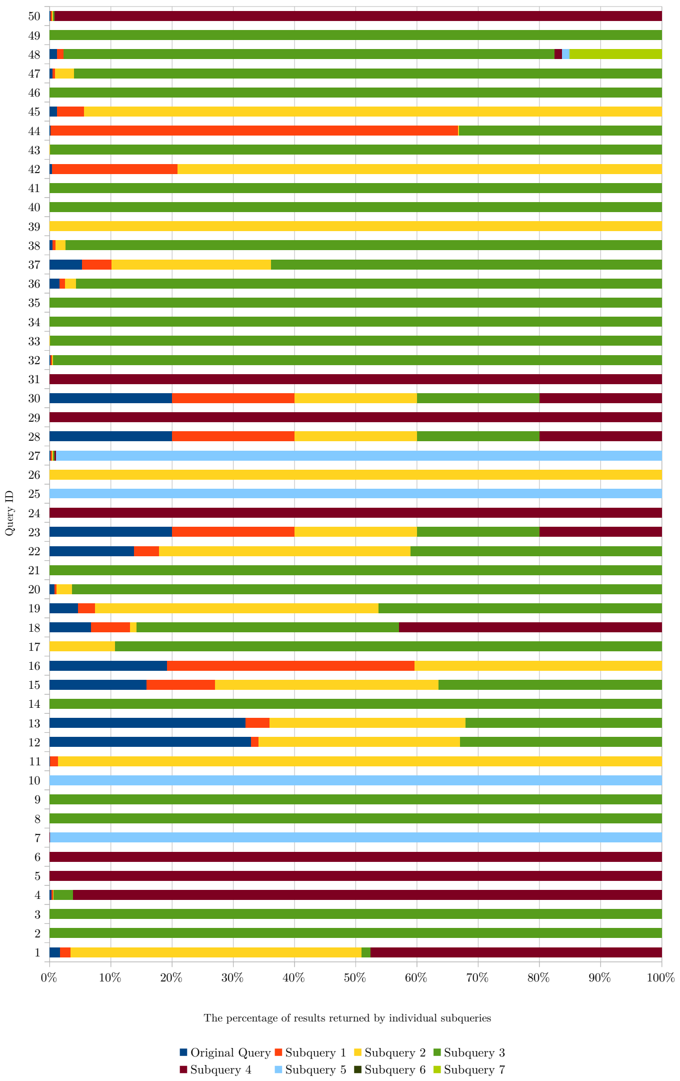


Figure 1: Relative number of results found using different subqueries for every query in CMath run

Table 1: CMath run querying statistics

Query ID	Number of Subqueries	Subquery Resp. Time [ms] Avg.	Min.	Max.	Query R. Time [ms]
01	5	33.20	28	38	166
02	4	31.75	21	46	127
03	4	33.25	26	49	133
04	5	48.80	30	82	244
05	5	43.20	33	57	216
06	5	48.40	32	73	242
07	6	41.67	31	59	250
08	4	37.50	31	51	150
09	4	34.50	13	54	138
10	6	42.33	29	55	254
11	3	34.67	29	44	104
12	4	59.75	38	76	239
13	4	98.50	52	165	394
14	4	33.00	28	44	132
15	4	40.75	27	54	163
16	3	43.67	39	46	131
17	4	31.00	13	45	124
18	5	34.40	25	51	172
19	4	37.25	32	52	149
20	4	31.00	29	33	124
21	4	34.50	28	42	138
22	4	43.00	30	59	172
23	5	58.40	45	78	292
24	5	43.60	27	63	218
25	6	51.83	29	119	311
26	3	32.67	6	54	98
27	6	40.83	28	65	245
28	5	316.40	117	428	1582
29	5	39.20	31	53	196
30	5	149.20	126	179	746
31	5	37.40	31	50	187
32	4	31.00	25	45	124
33	4	26.50	14	31	106
34	4	34.50	29	46	138
35	4	35.00	29	49	140
36	4	45.50	31	59	182
37	4	49.25	29	82	197
38	4	38.00	28	48	152
39	3	23.00	10	31	69
40	4	40.25	32	51	161
41	4	40.00	31	52	160
42	3	35.67	29	48	107
43	4	35.25	17	50	141
44	4	26.25	8	33	105
45	3	29.00	27	30	87
46	4	45.75	36	56	183
47	4	34.50	17	55	138
48	8	52.75	32	138	422
49	4	38.50	18	54	154
50	5	41.00	32	62	205
<all>	218	49.58	6	428	10808

as absolute scores are incomparable across subquery 1 and 2 results lists. Thus, it is not possible to generate final results list as a simple combination of results from all the subqueries ordered by the score.

Another reason to use a more complicated results merging procedure is a necessity of preference of results on the original user's query to the results found for subqueries. On the other hand, it is well possible that the first result of a subquery could be more relevant for the user than the 10th result on the original query.

To produce the final results list from the subqueries according to this hypothesis we used a method we refer to as '*strip-merging*' of the results. The main idea is interleaving of 'strips' of results from all the ordered results lists from the subqueries. The less modified subquery to the original query the 'wider' strip of results on more relevant position is used in the final result list.

Let us have x subqueries (the original one and $x-1$ derived subqueries). The top x most relevant results in the final result list are the first x most relevant results from the result list to the original query, then $x-1$ most relevant results from the first derived subquery are added, then $x-2$ results from the second subquery and so on until the first most relevant result from the last derived subquery is added. This procedure is then repeated with the next x results from the result list to the original query, $x-1$ results from the first subquery etc. until desired amount of results is reached. If all the results from some subquery are used and no more left we continue without changing the width of strips for other subqueries.

In the NTCIR-11 Math Task exactly 1,000 results were demanded for each of the queries. Provided all the subqueries together did not provide us with this number of results then a random selection from the database of indexed documents was used to fill in the gap. Score for all these artificial results in the subquery results lists was the same: 0.00000000001 This constant was selected as sufficiently small not to outnumber any real result score.

Strip-merging on three subqueries (the original one and two derived subqueries) is demonstrated in Example 2.

Results of the original query:
1: $r1_{\text{original}}$
2: $r2_{\text{original}}$
3: $r3_{\text{original}}$
4: $r4_{\text{original}}$
5: $r5_{\text{original}}$
6: $r6_{\text{original}}$
7: $r7_{\text{original}}$
8: $r8_{\text{original}}$
9: $r9_{\text{original}}$
10: $r10_{\text{original}}$
11: $r11_{\text{original}}$

Results of the subquery 1:
1: $r1_{\text{subquery 1}}$
2: $r2_{\text{subquery 1}}$
3: $r3_{\text{subquery 1}}$
4: $r4_{\text{subquery 1}}$
5: $r5_{\text{subquery 1}}$

Results of the subquery 2:
1: $r1_{\text{subquery 2}}$
2: $r2_{\text{subquery 2}}$
3: $r3_{\text{subquery 2}}$
4: $r4_{\text{subquery 2}}$
5: $r5_{\text{subquery 2}}$

The final result list:
1: $r1_{\text{original}}$
2: $r2_{\text{original}}$
3: $r3_{\text{original}}$
4: $r1_{\text{subquery 1}}$
5: $r2_{\text{subquery 1}}$
6: $r1_{\text{subquery 2}}$
7: $r4_{\text{original}}$
8: $r5_{\text{original}}$
9: $r6_{\text{original}}$
10: $r3_{\text{subquery 1}}$
11: $r4_{\text{subquery 1}}$
12: $r2_{\text{subquery 2}}$
13: $r7_{\text{original}}$
14: $r8_{\text{original}}$
15: $r9_{\text{original}}$
16: $r5_{\text{subquery 1}}$
No more results from subquery 1.
17: $r3_{\text{subquery 2}}$
18: $r10_{\text{original}}$
19: $r11_{\text{original}}$
No more results from the original query.
20: $r4_{\text{subquery 2}}$
21: $r5_{\text{subquery 2}}$
No more results from subquery 2.
22: $r1_{\text{random}}$
23: $r2_{\text{random}}$
...
1000: $r979_{\text{random}}$

Example 2: Strip-merging of results from three subqueries

Relevance scores from the subqueries are mutually incomparable for two results from different subqueries. Moreover, strip merging may not preserve score and rank consistency, i.e. $\text{rank}_i < \text{rank}_j$ iff $\text{score}_i < \text{score}_j$. Thus, scores in the final result list are not directly usable and have to be recomputed.

The score of the result in the final result list is computed as

$$\begin{aligned}\text{score} \quad = \quad & \text{maximumScore} \cdot (\text{targetNumberOfResults} - \\ & - (\text{resultNumber} - 1)) + \text{scoreInSubquery}\end{aligned}$$

where

- *maximumScore* is ceiling on the highest score of a result across all the subquery results lists,
- *targetNumberOfResults* is the desired number of results in the final results list,[5]
- *resultNumber* is rank of the result in the final result list, and
- *scoreInSubquery* is the original score of the result in the subquery result lits.

Scores computed in this way certainly retains the property of consistency of rank and score mentioned above.

> "There are three types of lies — lies, damn lies, and statistics."
> Benjamin Disraeli

4. INDEXING STATISTICS

NTCIR-11 Math Task dataset consists of around 100,000 documents divided into single paragraphs for better evaluation purposes. We think this is a good way of keeping the search and evaluation units at a reasonable size as well as the number of documents in the collection. As opposed to single-formula documents from NTCIR-10, the NTCIR-11 single-paragraph documents enable searchers to combine formula queries with text queries as well.

For the dataset we created an index with a little more than 3 billion indexed subformulae. Indexing process comprises of reading the source documents from a hard drive, parsing the XML, separating math content from textual content, analyzing and indexing textual content, canonicalization of mathematical notation, normalization of formulae and extraction of subformulae, and finally, putting all to the index.

The complete index statistics can be found in Tables 2 and 3.

Table 2: Index statistics

Indexing times [min]		Index
Wall Clock	**CPU**	**size [GiB]**
1,940.0	3,413.55	68

Table 3: Formulae count statistics

	Formulae	
Documents	**Original**	**Indexed**
8,301,545	59,647,566	3,021,865,236

In our previous research we have already observed that the indexing statistics of MIaS such as indexing time and index size are more dependant on the number of math formulae in the collection than on the number of documents. This is confirmed by the above numbers with comparison to NTCIR-10 MIaS data [11]. Nearly twice as many formulae in the dataset means almost exactly twice as many number of indexed documents as well as doubled CPU indexing time and final index size. Wall clock time is incomparable to the previous results due to the different indexing job management.

These performance results we find satisfying taking the extra overhead caused by the extended canonicalization process as well as the precision of the results into account.

> "There is only one thing that makes a dream impossible
> to achieve: the fear of failure."
> Paulo Coelho

5. DISCUSSION OF ACHIEVED RESULTS

The result of all runs submitted by MIRMU team can be found in Tables 4 and 5. The highest scores of our submitted runs of all teams are highlighted in bold — our system got the best results of all teams in 4 out of 6 evaluated categories but more importantly, we has achieved the best score of all teams in the evaluation of results with Relevance Level ≥ 3.

PMath run based solely on Presentation MathML reached the lowest precision from our runs. Nevertheless, the average ratio between PMath run and CMath run raised from 0.64 in NTCIR-10 to 0.90 in the current evaluation. This is thanks to the constant development of our open-source

[5] 1,000 for NTCIR-11 Math Task.

Table 4: Results of submitted runs with Relevance Level ≥ 3 (Relevant). Main task team rank is in [ ] for our best runs (in bold).

	PMath	CMath	PCMath	TeX
MAP avg	0.3073	**0.3630** [1]	0.3594	0.3357
P@10 avg	0.3040	**0.3520** [1]	0.3480	0.3380
P@5 avg	0.5120	**0.5680** [1]	0.5560	0.5400

Table 5: Results of submitted runs with Relevance Level ≥ 1 (Partially Relevant). Number in [ ] is team rank of all runs.

	PMath	CMath	PCMath	TeX
MAP avg	0.2557	**0.2807** [2]	0.2799	0.2747
P@10 avg	0.5020	0.5440	**0.5520** [1]	0.5400
P@5 avg	0.8440	**0.8720** [2]	0.8640	0.8480

standalone MathML Canonicalizer tool (see Section 2) which is an important preprocessing step in the indexing as well as searching phase of search. [6]

During the investigation of our results we have found several ‘classes’ of problems that are root of search engine failure.

Realistic User Query Formulation Problem

In the particular case the user was looking for formulae containing operator Im. However, this was not properly specified in the user’s query.

Index: `\operatorname{Im}P^{+}_{\Gamma}=C_{\mu}^{+}(\Gamma)`

$$\operatorname{Im} P^{+}_{\Gamma} = C_{\mu}^{+}(\Gamma)$$

Query: `ImP^{+}_{\gamma}=C^{+}_{\mu}(\gamma)`

$$ImP^{+}_{\gamma} = C^{+}_{\mu}(\gamma)$$

Index

```
   ...
  <mrow>
    <mo>Im</mo>
    <!-- U+2061 FUNCTION APPLICATION -->
    <mo>&#x2061;</mo>
    <msup>
   ...
```

Query

```
   ...
  <mrow>
    <mi>I</mi>
    <mi>m</mi>
    <msup>
   ...
```

This caused misinterpretation of the letters ‘Im’ to be multiplication of two variables I and m. With this query no matching formula was found even though suitable formula with `\operatorname{Im}` was in the index of the system.

It is unclear whether similar situations can be properly and easily handled by the system. Some kind of unification could be helpful: see proposal of possible simple unification for MIaS system in Section 6.

Substructure Difference Problem

The unification could possibly be helpful also in other situations where the query differs from the indexed formulae only in some of the subtrees.

Example 1

The following examples illustrate particular cases where the current version of MIaS missed the match even though the query and indexed formulae share large portions of code.

```
...
  <mrow>
    QUERY-FORMULA-SUBPART-1
    [[ INDEX
        <mrow>
          <mo>\int</mo>
          <mrow>
    || QUERY
        <mi>o</mi>
    ]]
    QUERY-FORMULA-SUBPART-2
    INDEX-FORMULA-SUPPLEMENT
        </mrow>
      </mrow>
    </mrow>
  </mrow>
...
```

Example 2

In the following example MIaS matches on complex formula with commutativity applied. However, more simple formula is not found due to difference in the complexity of the nominator and denominator in the query and indexed formulae:

Original task query:

```
\qvar{x}\frac{\qvar{y}}{\qvar{z}}
                      -\qvar{u}\frac{\qvar{v}}{\qvar{w}}
```

$$x\frac{y}{z} - u\frac{v}{w}$$

Matches:

$$\begin{aligned} \{q_s, q_r\} &= \int dx \int dy \, \{A(x,\mu)^p, A(y,\nu)^q\} \Big|_{\mu^{s+1}\nu^{r+1}} \\ &= pq\,\mu\nu \int dx\, A(\mu)^p (A(\nu)^q)' \\ &\quad \left[\left\{\frac{s}{p}\nu - \frac{r}{q}\mu\right\}\frac{1}{\mu - \nu} + \frac{1}{h}\frac{rs}{pq}\right] \Big|_{\mu^{s+1}\nu^{r+1}} \end{aligned}$$

Does not match:

$$\zeta \sim c_1\frac{\delta\rho_\sigma}{\rho_\sigma} - c_2\frac{\delta H_{\rm osc}}{H_{\rm osc}}$$

Query Variables Problem

Original task query:

```
\qvar{S}=-\qvar{T}_{\qvar{p}}\int\qvar{d}^{\qvar{p}+1}
                               \qvar{x}\sqrt{\qvar{g}}
```

Index: `S=-T_{p}\int d^{p+1}x\sqrt{-g}`

$$S = -T_p \int d^{p+1}x\sqrt{-g}$$

Query: `S=-T_{p}\int d^{p+1}x\sqrt{g}`

$$S = -T_p \int d^{p+1}x\sqrt{g}$$

MIaS missed the hit only due to the tiny difference between `\sqrt{g}` and `\sqrt{-g}`.

> "The future belongs to those who believe
> in the beauty of their dreams."
> Eleanor Roosevelt

6. CONCLUSIONS AND FUTURE WORK

We have described the story of MIaS leading to the performance we are getting now for keyword based textual and formulae search. We indexed canonicalized version of MathML, both Presentation and Content MathML. Best 'winning' results have been achieved with Content Math representation of data. Our explanation is that with Content MathML there is smaller degree of ambiguity than with Presentation MathML. We have described and evaluated our query expansion and merging strategy, which definitely helped to reach the best results.

The achieved performance allows the system to be used in real digital libraries as EuDML to the benefit of math-aware information seekers. Still, there is a long route to math-aware question answering and we still see large possibilities for improvements and optimizations of efficiency.

Further investigation of the best strategies of subqueries derivation from the original users' query is needed as well as proper evaluation of different strategies of merging subqueries results to the final result list.

In this context and in the long term we plan to experiment with *Strict Content MathML*, W3C subset of Content MathML. Supporting it in our canonicalization process may further decrease ambiguity in formulae indexing and retrieval.

Another area of future research is what we call '*Math Entailment*'. Textual entailment is directional relation between two fragments, text t and hypotheses h. t entails h if human would from t infer that h probably holds. Math entailment is entailment adapted to MIR by the usage of math formulae, connotations and named entities for t or h to model semantic relatedness or weighting similarity during indexing. We plan to measure its impact on MIR qualities on an available NTCIR evaluation database or on developed reference document and query corpus for MIR evaluation.

It would also be useful to implement simplified version of unification in our system. With query-time complexity in mind we would like to implement and do simple unification at indexing time as follows:

- The main idea is to implement a special identifier, let denote it $\boxed{\mathbb{U}}$, working as a single universal unifying element through the whole index.

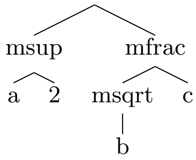
- The symbol $\boxed{\mathbb{U}}$ would be used to derive a set of unified versions of formulae from the original formula.

  The derived versions are generated 'layer-by-layer' according to the MathML tree structure by substituting subtrees for $\boxed{\mathbb{U}}$. For example:

  - Original formula:

    ```
    a^2+\frac{\sqrt{b}}{c}
    ```

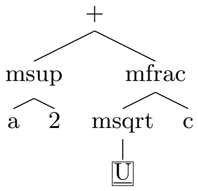
    $$a^2 + \frac{\sqrt{b}}{c}$$

  - Sequence of the unified formulae derivation:

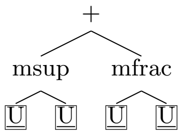
    ```
    a^2+\frac{\sqrt{\unificator}}{c}
    ```

    $$a^2 + \frac{\sqrt{\boxed{\mathbb{U}}}}{c}$$

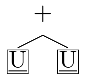
    ```
    \unificator^\unificator
            +\frac{\unificator}{\unificator}
    ```

    $$\boxed{\mathbb{U}}^{\boxed{\mathbb{U}}} + \frac{\boxed{\mathbb{U}}}{\boxed{\mathbb{U}}}$$

    ```
    \unificator+\unificator
    ```

    $$\boxed{\mathbb{U}} + \boxed{\mathbb{U}}$$

- This expansion of the original formula would be applied to every formula during the indexing of the database of documents. The same method would also be used to expand every formula in the user's query to extend the original query. Hits on unified formulae would be rated with gradually decreased score according to the unification level of the formula.

This approach does not cover all the possible substitutions on the formula tree. However, even this approach significantly increases probability of the query match on structurally similar formulae with reasonable amount of additional processing during indexing and just minimal addition of complexity during querying. The increase in index size would also be acceptable.

Acknowledgement. We would like to thank David Formánek for the first idea of the simple unification for MIaS.

We acknowledge the support (Short and Exchange Visit Grants 6965 and 6967) received from ESF, European Science Foundation, for the activity entitled ELIAS—Evaluating Information Access Systems.

7. REFERENCES

[1] *NTCIR Workshop 11 Meeting*, Tokyo, Japan, 2014.

[2] A. Aizawa, M. Kohlhase, I. Ounis, and M. Schubotz. NTCIR-11 Math-2 Task Overview. In *Proceedings of NTCIR-11 Math-2 task Workshop Meeting* [1].

[3] R. Ausbrooks, S. Buswell, D. Carlisle, G. Chavchanidze, S. Dalmas, S. Devitt, A. Diaz, S. Dooley, R. Hunter, P. Ion, M. Kohlhase, A. Lazrek, P. Libbrecht, B. Miller, R. Miner, C. Rowley, M. Sargent, B. Smith, N. Soiffer, R. Sutor, and S. Watt. Mathematical Markup Language (MathML) Version 3.0, 2010. W3C Recommendation 21 October 2010, `http://www.w3.org/TR/2010/REC-MathML3-20101021/`.

[4] J. B. Baker, A. P. Sexton, and V. Sorge. MaxTract: Converting PDF to LaTeX, MathML and Text. In J. Jeuring, J. A. Campbell, J. Carette, G. D. Reis, P. Sojka, M. Wenzel, and V. Sorge, editors, *AISC/DML/MKM/Calculemus*, volume 7362 of *Lecture Notes in Computer Science*, pages 422–426. Springer, 2012.

[5] T. W. Cole, I. Daubechies, K. M. Carley, J. L. Klavans, Y. LeCun, M. Lesk, C. A. Lynch, P. Olver, J. Pitman, and Z. J. Xia. *Developing a 21st Century Global Library for Mathematics Research.* National Research Council, Washington, D.C.: The National Academies Press, Mar. 2014.

[6] D. Formánek, M. Líška, M. Růžička, and P. Sojka. Normalization of digital mathematics library content. In J. Davenport, J. Jeuring, C. Lange, and P. Libbrecht, editors, *24th OpenMath Workshop, 7th Workshop on Mathematical User Interfaces (MathUI), and Intelligent Computer Mathematics Work in Progress*, number 921 in CEUR Workshop Proceedings, pages 91–103, Aachen, 2012. http://ceur-ws.org/Vol-921/wip-05.pdf.

[7] J. Grimm. Producing MathML with Tralics. In P. Sojka, editor, *Proceedings of DML 2010*, pages 105–117, Paris, France, July 2010. Masaryk University. http://dml.cz/dmlcz/702579.

[8] J. Hoffman. Starting small but adding up: a free maths archive. *Nature*, 454:263, 2008. http://www.nature.com/news/2008/080716/full/454263b.html.

[9] M. Líška. Evaluation of Mathematics Retrieval, Jan. 2013. Master Thesis, Masaryk University, Brno, Faculty of Informatics (advisor: Petr Sojka), https://is.muni.cz/th/255768/fi_m/?lang=en.

[10] M. Líška, P. Sojka, M. Růžička, and P. Mravec. Web Interface and Collection for Mathematical Retrieval: WebMIaS and MREC. In P. Sojka and T. Bouche, editors, *Towards a Digital Mathematics Library. Bertinoro, Italy, July 20–21st, 2011*, pages 77–84. Masaryk University, July 2011. http://hdl.handle.net/10338.dmlcz/702604.

[11] M. Líška, P. Sojka, and M. Růžička. Similarity Search for Mathematics: Masaryk University team at the NTCIR-10 Math Task. In N. Kando and K. Kishida, editors, *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*, pages 686–691, Tokyo, 2013. National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/06-NTCIR10-MATH-LiskaM.pdf.

[12] M. Líška, P. Sojka, and M. Růžička. Math indexer and searcher web interface: Towards fulfillment of mathematicians' information needs. In S. M. Watt, J. H. Davenport, A. P. Sexton, P. Sojka, and J. Urban, editors, *Intelligent Computer Mathematics CICM 2014. Proceedings of Calculemus, DML, MKM, and Systems and Projects*, pages 444–448, Zurich, 2014. Springer International Publishing Switzerland. http://dx.doi.org/10.1007/978-3-319-08434-3_36.

[13] J. Mišutka and L. Galamboš. Extending Full Text Search Engine for Mathematical Content. In Sojka [16], pages 55–67. http://dml.cz/dmlcz/702546.

[14] M. Růžička, P. Sojka, and M. Líška. Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy. In *Proceedings of NTCIR-11 Math-2 task Workshop Meeting* [1].

[15] P. Sojka. From Scanned Image to Knowledge Sharing. In K. Tochtermann and H. Maurer, editors, *Proceedings of I-KNOW '05: Fifth International Conference on Knowledge Management*, pages 664–672, Graz, Austria, June 2005. Know-Center in coop. with Graz Uni, Joanneum Research and Springer Pub. Co.

[16] P. Sojka, editor. *Towards a Digital Mathematics Library*, Birmingham, UK, July 2008. Masaryk University. http://dml.cz/dmlcz/702564.

[17] P. Sojka. Digitization Workflow in the Czech Digital Mathematics Library. *Math-for-Industry Lecture Note Series*, 22:272–280, Dec. 2009.

[18] P. Sojka. Exploiting Semantic Annotations in Math Information Retrieval. In J. Kamps, J. Karlgren, P. Mika, and V. Murdock, editors, *Proceedings of ESAIR 2012 c/o CIKM 2012*, pages 15–16, Maui, Hawaii, USA, 2012. Association for Computing Machinery. http://doi.acm.org/10.1145/2390148.2390157.

[19] P. Sojka and M. Kohlhase, editors. *Towards a Digital Mathematics Library: MIR and DML 2012.* Masaryk University, Dec. 2014. to appear.

[20] 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 2011*, volume 6824 of *LNAI*, pages 228–243, Berlin, Germany, July 2011. Springer-Verlag. http://dx.doi.org/10.1007/978-3-642-22673-1_16.

[21] P. Sojka and M. Líška. The Art of Mathematics Retrieval. In *Proceedings of the ACM Conference on Document Engineering, DocEng 2011*, pages 57–60, Mountain View, CA, Sept. 2011. Association of Computing Machinery. http://doi.acm.org/10.1145/2034691.2034703.

[22] P. Sojka and J. Rákosník. From Pixels and Minds to the Mathematical Knowledge in a Digital Library. In Sojka [16], pages 17–27. http://dml.cz/dmlcz/702564.

[23] H. Stamerjohanns, D. Ginev, C. David, D. Misev, V. Zamdzhiev, and M. Kohlhase. MathML-aware Article Conversion from LaTeX. In P. Sojka, editor, *Proceedings of DML 2009*, pages 109–120, Grand Bend, Ontario, CA, July 2009. Masaryk University. http://dml.cz/dmlcz/702561.

[24] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori. INFTY — An integrated OCR system for mathematical documents. In C. Vanoirbeek, C. Roisin, and E. Munson, editors, *Proceedings of ACM Symposium on Document Engineering 2003*, pages 95–104, Grenoble, France, 2003. ACM.

[25] K. Wojciechowski, A. Nowiński, P. Sojka, and M. Líška. The EuDML Search and Browsing Service - Final, Feb. 2013. Deliverable D5.3 of EU CIP-ICT-PSP project 250503 EuDML: The European Digital Mathematics Library, revision 1.2 https://project.eudml.eu/sites/default/files/D5_3_v1.2.pdf.