Math Indexer and Searcher under the Hood: Fine-tuning Query Expansion and Unification Strategies

Michal Růžička
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
ORCID 0000-0001-5547-8720
mruzicka@mail.muni.cz

Petr Sojka
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
ORCID 0000-0002-5768-4007
sojka@fi.muni.cz

Martin Líška
Faculty of Informatics
Masaryk University
Botanická 68a, 602 00 Brno
Czech Republic
ORCID 0000-0002-0521-0117
martin.liski@mail.muni.cz

ABSTRACT

This paper summarizes the experience of Math Information Retrieval team of Masaryk University (MIRMU) with the NTCIR-12 MathIR arXiv Main Task and its subtasks.

We based our approach on the MIaS system. Based on NTCIR-11 Math-2 Task relevance judgements, we developed an evaluation platform. Using this platform we rigorously evaluated combinations of new features and picked the most promising ones for the NTCIR-12 evaluation.

The new features tested are mostly aimed at further canonicalizing MathML input, structurally unifying formulae for syntactic-based similarity search and query expansion when combining text and math query terms.

Team Name

MIRMU (Math Information Retrieval at Masaryk University)

Subtasks

MathIR arXiv Main Task (English),
optional MathIR arXiv Formula Similarity Task (English),
optional MathIR Wikipedia Task (English),
optional MathIR Wikipedia Formula Browsing Task (English)

Keywords

similarity search, math information retrieval, MIaS, evaluation, MathML, query expansion

1. INTRODUCTION

Structured mathematical notation is irreplaceable part of mathematical vernacular. Searching it should be supported in the query language of full-text-search systems to help researchers handle the exponentially growing amount of mathematical literature. To cover specifics of math-aware full text search we have designed our Math Indexer and Searcher (MIaS) system [7] as probably the first production quality math-aware search system [9] indexing hundreds of thousands of documents.

MIR research attracted several research groups as Pilot Math Task has been set up at NTCIR-10 where MIaS used for the first time MathML canonicalization module and Content MathML indexing. [4] For NTCIR-11 [1], MIaS was enhanced with query expansion strategies and better canonicalization leading to superb results. [5]

Building on our experience from NTCIR-11 we exploited NTCIR-11 Math-2 Task relevance judgements to develop an evaluation platform for rigorous evaluation of several combinations of new features [3] and consequently picked the most promising ones for our participation in both Main and Wiki MathIR Task of NTCIR-12 [10].

To address the need for structural substitutions in query formulae to match indexed formulae we extended MIaS system with MathML structural unification component. This component is described in Section 2.1. New abilities of MIaS on operator unification are presented in Section 2.2. System of combining these features to prepare our results for NTCIR-12 MathIR is described in Section 3. Since NTCIR-11 we also further enhanced and evaluated several new querying strategies, as summarized in Section 4. In Section 5 we describe our strategy to select the most promising setup to generate our final NTCIR-12 results. We conclude with brief discussion of our results in Sections 6 and 7.

2. UNIFICATION

2.1 Structural Unification

One of the important features we were missing in our system on NTCIR-11 [5] was ability to substitute complex structures in query formulae for different structures in similar formulae in the index, i.e. to allow match of query formula $a^2 + \frac{\sqrt{b}}{c}$ on indexed formula $a^2 + \frac{x}{y}$ where numerator $\sqrt{b}$ differs *structurally* from nominator x of the indexed formula.

To allow our system to do this *structural unification* we implemented open-source tool MathML Unificator[1], that is publicly available in GitHub repository as well as in Maven Public Repositories, that is usable as standalone command-line utility or Java library embeddable in other systems.

Using this tools, we are able to generate series of structurally unified versions of the input formulae. The unification is done according to MathML tree 'layers' as shown in Figure 1. Unification of query formula $a^2 + \frac{\sqrt{b}}{c}$ results in series of derived formulae:

1. $a^2 + \frac{\sqrt{\square}}{c}$,
2. $\square^{\square} + \frac{\square}{\square}$,
3. $\square + \square$

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

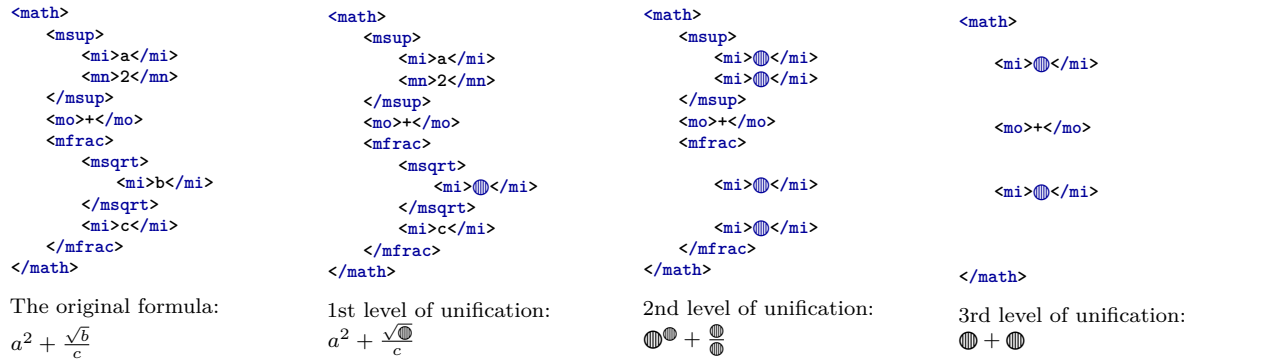

Figure 1: Process of MathML structural unification driven by 'layers' of the formula MathML tree

Similarly, document formula $a^2 + \frac{x}{y}$ is at indexing time indexed in its original version but also structurally unified derivatives are index with appropriately lowered weights:

1. $\square^{\square} + \frac{\square}{\square}$,
2. $\square + \square$

This consequently allows match of the query formula $a^2 + \frac{\sqrt{b}}{c}$ with indexed formula $a^2 + \frac{x}{y}$ on common structurally unified derivatives $\square^{\square} + \frac{\square}{\square}$ and $\square + \square$.

Unification is done on indexing time that allows faster response of the search system for interactive users. On the other hand that implies restrictions on the number of derivatives generated out of the input formulae not to overfill the index with unified derivatives. To address these issues the structural unification is driven by 'layers' of the formula MathML tree (see Figure 1). Thus, not all possible combination of structure differences are addressed. For example, $a^{\square} + \frac{x}{y}$ is never derived from the index formula from our example above as the upper index lies at the same level of the MathML tree as a, x and y. On the other hand, this significantly limits the number of derived formulae to be put into the index and structurally similar formulae can match on more general structural-unification derivatives.

The MathML Unificator tool is now integrated with our (Web)MIaS system and structural unification is done at indexing time for formulae of the indexed documents as well as at query time to structurally unify formulae from the users' queries. One issue still to be solved is proper weighting of structurally unified derivatives of the input formulae in relation to the original non-modified formulae and tokenized and unified subformulae (see MIaS processing description in [6]).

2.2 Operator Unification

Structural unification described in Section 2.1 is not the only form of unification our system does. MIaS itself unifies math expressions in several ways. Namely, the processing consists of formula tree ordering, unification of variables, unification of number constants and MathML attributes handling. Each processing step creates a new and more unified version of the original expression. This process is described in detail in [6].

We analyzed NTCIR-11 results and identified errors in retrieval caused by different additive operators in index term and query term [5, p. 132]. Difference in an additive operator in math expressions may be in some cases an insignificant difference to the semantics of the formula.

With this motivation, we experimented with operators unification feature for NTCIR-12. First, we remove all unary additive operators from expressions. This allows MIaS to match, for example, $-tr(x\,ln\,x)$ with $tr(x\,ln\,x)$. Secondly, we substitute all additive operators with a universal symbol that represents this group of 'similar' operators. It will be possible to define other groups of 'similar' operators for the substitution in the future. These expressions have naturally lost some of their original information, therefore they are retrieved with a lower score based on an operators unification factor.

3. COMBINING SYSTEM FEATURES

We experimented with the features explained in the previous sections and other utilities integrated with our system. We set up combinations of these features in different arrangements with other MIaS features as canonicalization and its various configurations.

The functionality we put into different combinations are as follows[2]:

Canonicalization (*canon*) The canonicalization process, in detail described in [2], is based on our publicly available open-source tool MathML Canonicalizer[3] and aims to normalize different possible serializations (different notations in MathML encoding) of the same math formulae to one canonical version of such a formula that will be used at indexing time as well as at querying time resulting in match of the same formulae with distinct serializations to MathML on this canonical version.

The normalization is optimized for similarity search. Thus, the normalization steps does not necessarily preserve full semantic information of the original formulae (see the next point) but possibly removes 'negligible' differences in behalf of similarity matches.

The MathML Canonicalizer tool is configurable to allow users to easily select normalization operations that are suitable for their needs.

[2]Short names in brackets are for reference with Table 1.
[3]https://mir.fi.muni.cz/mathml-normalization/#mathml-canonicalizer

Canonicalization operators removal (*canon-rm-oper*) One of the canonicalization tool configuration options (see the previous point) we experimented with for NTCIR-12 was the list of math operators to be removed from canonicalized formulae. When use of this function is indicated in Table 1, operators

- U+2062 INVISIBLE TIMES
- U+22C5 DOT OPERATOR
- U+002A ASTERISK
- U+2063 INVISIBLE SEPARATOR
- U+2064 INVISIBLE PLUS

were completely removed from the formula. When use of the feature is not indicated in Table 1 operators were left intact in the formulae.

Unary operators removal (*unary-rm-oper*) Removal of unary operators is a completely new feature of the MathML Canonicalizer described in detail in Section 2.2.

Operators unification (*oper-unif*) Unification of groups of 'similar' operators is a new feature of the MIaS system unification procedure described in Section 2.2.

Structural unification (*struct-unif*) The completely new concept of indexing structurally unified derivatives of the original formulae fist used by MIaS system at NTCIR-12 is presented in Section 2.1.

We prepared several dozens of standalone indices for the Main and Wiki Math Task of NTCIR-12 representing various combinations of the features described above and other implementation details of our system such as particular weighting of the products of various unification operations. Then we evaluated them using different query expansion techniques (see Section 4). The indices we picked for the final cursory manual evaluation on their results on the NTCIR-12 Math Tasks topics are summed up in Table 1 with indication of used combination of the features.

Please note the index names corresponds to the precise revision of MIaS tools as tagged in their GitHub repositories.[4] The Main Task indices follows naming convention ntcir-12-1x and the Wiki Task indices ntcir-12-3x where x indicates equal set of (un)used features.

4. NEW QUERYING STRATEGIES

Based on our experience on NTCIR-11 and ground truth of judged data resulting from NTCIR-11 we developed an evaluation framework that allowed us to rigorously evaluate several new querying strategies. The summary of our results is available in [3].

The query relaxation strategy we used at NTCIR-11 we call *Leave Rightmost Out* (LRO). For its description please see [5, pp. 129–131]. Based on this concept we evaluated also other querying strategies:

[4]https://github.com/michal-ruzicka/MathMLCan, https://github.com/michal-ruzicka/MathMLUnificator, https://github.com/martinliska/MIaSMath, https://github.com/martinliska/MIaS, https://github.com/martinliska/WebMIaS

Table 1: Summary of features of our search system indices ntcir-12-10/30, ntcir-12-15/35, ntcir-12-18/38, ntcir-12-19/39 used for NTCIR-12 MathIR

	Index ntcir-12-x			
Feature	***10/30***	***15/35***	***18/38***	***19/39***
canon	•	–	•	•
canon-rm-oper	•	–	•	•
unary-rm-oper	–	–	•	•
oper-unif	–	–	•	•
struct-unif	–	–	•	• (progressive weighting)

***Original Query Only* (OQO)** The basic reference querying strategy is to use the original query without any modifications or derived subqueries. Results found for the original query is the final list of results returned to the user.

***Math Terms Only* (MTO)** *Math Terms Only* querying strategy is simple modification of the *Original Query Only* strategy: The query consists of formulae from the original query, all the text keywords are removed from the query.

***Text Terms Only* (TTO)** In *Text Terms Only* strategy the query consists of only text keywords from the original query.

***All Possible Subqueries* (APS)** The opposite extreme to using the original query only is to use all the possible subqueries derivable from the original query. Provided the original query consists of x formulae and y text keywords, all the possible combinations of formulae $f_1, \ldots, f_x$ and text keywords $k_1, \ldots, k_y$ provide us with $2^{x+y} - 1$ non-empty subqueries (including the original query itself). Weight of interleaving 'strips' of results from subqueries depends on the degree of modification of the query comparing to the original query.

***Leave One Out* (LOO)** The *Leave One Out* querying strategy is similar to the *All Possible Subqueries* strategy with the following differences:

- We work with a restricted set of the subqueries—only the original query and derived subqueries with exactly one component (one formula or one text keyword) excluded are used.
- Weight of interleaving 'strips' of results from subqueries is 2 if taking results from the original query results list, and 1 otherwise.

***Leave One or Two Out* (LOoTO)** The *Leave One or Two Out* querying strategy is further extension of the similar *Leave One Out* strategy:

- The set of the subqueries consists of the original query and derived subqueries with exactly one or two components excluded.
- The strip-weight is 3 if taking results from the original query results list, 2 if taking results from a derived query with exactly one component excluded, and 1 otherwise.

4.1 Phrase and Full Phrase Expansion

In addition to various query expansion schemas described in Section 4 we experimented with modification of the text component of the original queries *before* applying particular querying strategy.

These modifications are only applicable on multi-word text keywords of the original query. For example, having query

Formula 1: $\aleph_0$
Keyword 1: categorical simple theory

we have two methods of transformation — *phrase* and *full phrase expansion*:

Phrase exansion For multi-word keywords individual words are used instead of the original multi-word keywords.

Thus, our query is transformed to query:

Formula 1: $\aleph_0$
Keyword 1: categorical
Keyword 2: simple
Keyword 3: theory

Full phrase exansion Individual words from the multi-word keywords are added one by one at the end of the keywords list (removing duplicates across multiple multi-keywords, if any).

Thus, our query is transformed to query:

Formula 1: $\aleph_0$
Keyword 1: categorical simple theory
Keyword 2: categorical
Keyword 3: simple
Keyword 4: theory

This modified version is then used as the 'original' query inputting to the querying strategy (see Section 4) providing the system more flexibility for query relaxation and boolean operations on the query components.

5. RUNS SELECTION

Based on NTCIR-11 ground truth data we developed an evaluation framework that allowed us to rigorously evaluate every system setup we put together and let us compare it with all previous setups. [3]

We have built several dozen indices for the Main and Wiki Math Task with different features and configurations enabled for the indexing and processing of math data. See Section 3. Out of them we selected 10 indices for the Main and 10 indices for the Wiki Task with the same configurations differing only in the set of indexed input documents.

We queried each index with the full set of querying strategies described in Section 4. We queried each index with full 50 topics from NTCIR-11 with 11 different querying strategies.

For each querying strategy we also tried both original queries and two types of text keywords phrases expansions described in Section 4.1.

For each combination *index–querying strategy–phrase expansion strategy* we also tested math query input types with, from our experience, the best results: Content MathML and mixed Presentation & Content MathML.

To summarize, using the evaluation framework we evaluated number of runs which can be defined as a Cartesian product of 10 indices, 11 querying strategies, 3 phrase expansion strategies and 4 input types. This gave us 660 results, each of which gave us the performance of each particular combination. We used MAP and Bpref metrics for evaluation against NTCIR-11 ground truth.

We put all 660 runs into two tables. The first table contained MAP measures for each run; the second table contained Bpref measures for each run. Sample view of this table is shown in Figure 2. We highlighted the top performing combinations which became the main candidates for the use in NTCIR-12.

For NTCIR-12 submission we handpicked the most promising runs in terms of precision × recall and consequently briefly manually evaluated their results on NTCIR-12 topics. We also picked some of the runs which we wanted to have evaluated in NTCIR-12 even though they were not among the best, for comparison and to get better insight into system parameters.

6. NOTES ON TASKS AND STATISTICS

6.1 Handling Query Variables and Similarity Regions

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.

For use of MIaS, `<qvar>` elements were transformed to regular Presentation/Content MathML identifiers, i.e. `<mi>`/`<ci>` elements. As the value of the new operator element the value of the `name` attribute of the particular `qvar` element was used literally. To show an example, the task formula element

```
<m:mrow xml:id="m1.1.4.pmml" xref="m1.1.4">
  <m:mi xml:id="m1.1.1.pmml" xref="m1.1.1">x</m:mi>
  <m:mo xml:id="m1.1.2.pmml" xref="m1.1.2">+</m:mo>
  <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="y"/>
</m:mrow>
```

was transformed to:

```
<m:mrow id="m1.1.4.pmml" xref="m1.1.4">
  <m:mi id="m1.1.1.pmml" xref="m1.1.1">x</m:mi>
  <m:mo id="m1.1.2.pmml" xref="m1.1.2">+</m:mo>
  <m:mi>y</m:mi>
</m:mrow>
```

Wiki Task topics specifications used a different `<qvar>` markup:

```
<qvar>*1*</qvar>
```

To keep our workflow identical for both Main and Wiki Task we preprocessed the Wiki Task topics transforming `<qvar>` elements to the Main Task format:

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

Similarly, MIaS system has no means of letting the user define similarity regions in the formula query as required in the Simto Subtask of the NTCIR-12 Math Main Task.

Strategy of our team was to transform every formula with at least one `<simto>` area to two formulae:

- 'Unified' version of the formula with whole `<simto>` region substituted for single `<mi>`/`<ci>` element with text content derived from the `name` attribute of the `<simto>` element, i.e. transformation very similar to `<qvar>` transformation.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	BPREF	PCM-T	CM-APSWMWWS	CM-LOOTO	CM-LOO	CM-N1MRIR	CM-N1MRMSFR	CM-N1MRMSF	CM-N1MR	CM-N1M	CM-OQFPO	CM-OQKPO	CM-OQO
4	ntcir-12-10-pfe	0,0492	0,1029	0,1011	0,0987	0,1047	0,1047	0,1057	0,1027	0,1055	0,0220	0,0619	0,0343
5	ntcir-12-11	0,0557	0,0897	0,0873	0,0898	0,0972	0,0977	0,0973	0,0956	0,0977	0,0216	0,0514	0,0348
6	ntcir-12-11-fe	0,0529	0,0964	0,0991	0,1040	0,1105	0,1104	0,1107	0,1079	0,1110	0,0216	0,0673	0,0341
7	ntcir-12-11-pfe	0,0509	0,1037	0,1015	0,0987	0,1045	0,1038	0,1055	0,1019	0,1051	0,0216	0,0616	0,0348
8	ntcir-12-12	0,0552	0,0893	0,0866	0,0891	0,0970	0,0971	0,0969	0,0951	0,0969	0,0226	0,0511	0,0367
9	ntcir-12-12-fe	0,0527	0,0949	0,0969	0,1023	0,1094	0,1090	0,1097	0,1073	0,1094	0,0226	0,0670	0,0351
10	ntcir-12-12-pfe	0,0502	0,1009	0,0989	0,0976	0,1040	0,1033	0,1045	0,1015	0,1040	0,0226	0,0614	0,0357
11	ntcir-12-13	0,0561	0,0882	0,0860	0,0889	0,0960	0,0967	0,0962	0,0945	0,0970	0,0224	0,0513	0,0373
12	ntcir-12-13-fe	0,0529	0,0963	0,0981	0,1032	0,1089	0,1096	0,1094	0,1067	0,1099	0,0224	0,0671	0,0363
13	ntcir-12-13-pfe	0,0517	0,1014	0,1012	0,0986	0,1035	0,1034	0,1043	0,1013	0,1045	0,0224	0,0616	0,0368
14	ntcir-12-14	0,0548	0,0871	0,0858	0,0872	0,0944	0,0945	0,0947	0,0927	0,0947	0,0221	0,0511	0,0356
15	ntcir-12-14-fe	0,0517	0,0935	0,0954	0,1008	0,1066	0,1071	0,1078	0,1046	0,1075	0,0221	0,0668	0,0340
16	ntcir-12-14-pfe	0,0503	0,0995	0,0988	0,0965	0,1021	0,1014	0,1029	0,0996	0,1024	0,0221	0,0613	0,0350
17	ntcir-12-15	0,0965	0,0892	0,0870	0,0895	0,0984	0,0986	0,0984	0,0966	0,0987	0,0220	0,0515	0,0340
18	ntcir-12-15-fe	0,1067	0,0942	0,0980	0,1033	0,1096	0,1102	0,1102	0,1081	0,1106	0,0220	0,0671	0,0332
19	ntcir-12-15-pfe	0,1021	0,1013	0,1001	0,0979	0,1044	0,1040	0,1052	0,1020	0,1048	0,0220	0,0614	0,0338
20	ntcir-12-16	0,0955	0,0884	0,0870	0,0895	0,0974	0,0976	0,0980	0,0961	0,0974	0,0225	0,0512	0,0347
21	ntcir-12-16-fe	0,1058	0,0961	0,0988	0,1038	0,1097	0,1100	0,1104	0,1079	0,1104	0,0225	0,0668	0,0340
22	ntcir-12-16-pfe	0,1017	0,1032	0,1013	0,0989	0,1042	0,1040	0,1051	0,1019	0,1046	0,0225	0,0614	0,0345
23	ntcir-12-17	0,0665	0,0662	0,0682	0,0677	0,0653	0,0639	0,0671	0,0642	0,0659	0,0085	0,0512	0,0411
24	ntcir-12-17-fe	0,0579	0,0602	0,0646	0,0661	0,0616	0,0594	0,0597	0,0578	0,0590	0,0085	0,0670	0,0388
25	ntcir-12-17-pfe	0,0570	0,0685	0,0708	0,0706	0,0653	0,0654	0,0662	0,0640	0,0674	0,0085	0,0615	0,0440
26	ntcir-12-18	0,0467	0,0438	0,0443	0,0446	0,0433	0,0419	0,0434	0,0420	0,0427	0,0017	0,0512	0,0236
27	ntcir-12-18-fe	0,0495	0,0484	0,0509	0,0529	0,0509	0,0484	0,0479	0,0463	0,0481	0,0017	0,0668	0,0285
28	ntcir-12-18-pfe	0,0496	0,0496	0,0540	0,0540	0,0477	0,0459	0,0470	0,0458	0,0466	0,0017	0,0613	0,0273
29	ntcir-12-19	0,0000	0,0697	0,0699	0,0681	0,0688	0,0681	0,0702	0,0679	0,0696	0,0089	0,0514	0,0361
30	ntcir-12-19-fe	0,0000	0,0763	0,0781	0,0794	0,0800	0,0772	0,0771	0,0748	0,0772	0,0089	0,0673	0,0432
31	ntcir-12-19-pfe	0,0000	0,0770	0,0809	0,0794	0,0757	0,0734	0,0745	0,0729	0,0745	0,0089	0,0617	0,0417

Figure 2: View of Bpref evaluation results from all runs base on NTCIR-11 ground truth

- 'Full' version of the formula with `<simto>`/`<exact>` markup striped off[5], i.e. the formula was handled by the system as if no similarity region was defined for it at all.

Having these two variants together in the query we expected structural unification integrated in our system (see Section 2.1) to be able to find suitable similarity-match candidates.

To show a simple example, to prepare the 'unified' version for the formula from the task formula element

```
<m:mrow xml:id="m1.1.10.1.pmml" xref="m1.1.10.1">
  <mws:simto name="a">
    <m:mrow>
      <m:mi xml:id="m1.1.1.pmml" xref="m1.1.1">d</m:mi>
      <m:mo xml:id="m1.1.10.1.1.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
      <m:mi xml:id="m1.1.2.pmml" xref="m1.1.2">c</m:mi>
      <m:mo xml:id="m1.1.10.1.1a.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
      <m:mi xml:id="m1.1.3.pmml" xref="m1.1.3">l</m:mi>
    </m:mrow>
  </mws:simto>
  <m:mo xml:id="m1.1.10.1.1b.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
  <m:mrow xml:id="m1.1.10.1.2.pmml">
    <m:mo xml:id="m1.1.4.pmml">(</m:mo>
    <mws:qvar xmlns:mws="http://search.mathweb.org/ns" name="a"/>
    <m:mo xml:id="m1.1.6.pmml">)</m:mo>
  </m:mrow>
</m:mrow>
```

we substituted the `<simto>` region for `<mi>` element:

```
<m:mrow id="m1.1.10.1.pmml" xref="m1.1.10.1">
  <m:mi>simtovar:a</m:mi>
  <m:mo id="m1.1.10.1.1b.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
  <m:mrow id="m1.1.10.1.2.pmml">
    <m:mo id="m1.1.4.pmml">(</m:mo>
    <m:mi>a</m:mi>
    <m:mo id="m1.1.6.pmml">)</m:mo>
  </m:mrow>
</m:mrow>
```

In Content MathML markup, substitution for `<ci>` was done respectively.

The 'full' version of the formula from the example above is as follows:

```
<m:mrow id="m1.1.10.1.pmml" xref="m1.1.10.1">
  <m:mrow>
    <m:mi id="m1.1.1.pmml" xref="m1.1.1">d</m:mi>
    <m:mo id="m1.1.10.1.1.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
    <m:mi id="m1.1.2.pmml" xref="m1.1.2">c</m:mi>
    <m:mo id="m1.1.10.1.1a.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
    <m:mi id="m1.1.3.pmml" xref="m1.1.3">l</m:mi>
  </m:mrow>
  <m:mo id="m1.1.10.1.1b.pmml" xref="m1.1.10.1.1">&#x2062;</m:mo>
  <m:mrow id="m1.1.10.1.2.pmml">
    <m:mo id="m1.1.4.pmml">(</m:mo>
    <m:mi>a</m:mi>
    <m:mo id="m1.1.6.pmml">)</m:mo>
  </m:mrow>
</m:mrow>
```

Please note that the text content of the `<mi>`/`<ci>` element produced by `<simto>` substitution differs from the `<mi>`/`<ci>` elements via `<qvar>` substitution—text prefix `simtovar:` is added to the `name` attribute of the `<simto>` element to distinguish these two distinct variants.

If any `<exact>` subregion was defined in the `simto` area, the contents of the `<exact>` region was completely hidden in the 'unified' version of the formula as the whole `simto` area (including the `<exact>` region inside) was replaced by the `<mi>`/`<ci>` element. In contrast, in the 'full' version of the formula was this content preserved as well as any other content of the formula but the `<exact>` markup was removed together with the `simto` markup.

[5]Only the region indication markup itself was removed, the math contents of the region was left intact.

6.2 Main Tasks

There were 8,301,545 documents in the Main Task document collection. The collection contained 119,306,300 formulae. Statistics for the indices used for this task are summed up in the Table 2. We can see that adding structural unification to the indexing process doubles the number of indexed formulae, makes the index five times heavier and it takes about five times longer to index.

Table 2: Main Task indices statistics

Index	Indexing times [min]		Index size [GiB]	Indexed formulae
	Wall Clock	CPU		
10	1,054	1,453	47	2,354,850,850
15	769	1,155	59	2,704,770,446
18	5,913	5,604	288	5,591,527,950
19	5,181	5,486	288	5,592,489,129

6.3 Wiki Tasks

The original version of the Wiki data set was provided as HTML5 data. Moreover, same of these documents contained several syntactical and other errors. Our system works internally with XML representation of input data (XHTML with MathML for math) and depends at least on well formed XML on the input. To mitigate these issues we used Tidy (version 5.1.25) tool[6] to transforme Wiki data set of HTML documents to XHTML5 using the GNU Parallel [8] tool:

```
$ parallel --nice 19 -j500% --timeout 30 --null --keep-order \
>   echo '"### Processing {} ###"'\; \
>   tidy -q --output-xhtml yes --char-encoding utf8 \
>   --clean yes --indent yes --doctype auto --write-back yes \
>   -w 0 --numeric-entities yes --force-output yes {} \
> :::: <(find NTCIR12_MathIR_WikiCorpus_v2.1/ -type f -print0)
```

Consequently were addressed problems with left unescaped ampersand signs (`&`) with a Perl one-liner:

```
$ parallel --nice 19 -j500% -vv --bar --null \
>   perl -pi -e "'s/\&(?!(([a-z]{1,30})|(#\d+));)/&amp;/aig'" {} \
> :::: <(find NTCIR12_MathIR_WikiCorpus_v2.1/ -type f -print0)
```

This way we got almost completely (there were just 45 documents with possible errors) XML well formed dataset. The Tidy processing is heuristic so semantic errors are possible but we believe math content was untouched and plain text extracted from the modified data was also the same.

Based on this conversion a new version of the Wiki data set was released and used by all Wiki Task participants.

In addition to the dataset conversion small modifications were made on XML with specification of Wiki Task topics. In MathML of formulae the order of MathML flavours inside the `<semantics>` elements was reversed from

```
<math>
  <semantics>
    Presentation MathML
    <annotation-xml encoding="MathML-Content">
        Content MathML
    </annotation-xml>
    <annotation encoding="application/x-tex">
        TeX
    </annotation>
  </semantics>
</math>
```

to

```
<math>
  <semantics>
    Content MathML
    <annotation-xml encoding="MathML-Presentation">
        Presentation MathML
    </annotation-xml>
    <annotation encoding="application/x-tex">
        TeX
    </annotation>
  </semantics>
</math>
```

to be consistent with the format of the Main Task topics. This way we were able to apply the very same processing workflow for both Main and Wiki Task topics.

There were 319,763 documents in the Wiki Task document collection. The collection contained 1,184,528 formulae. Statistics for the indices used for this task are summed up in the Table 3.

Table 3: Wiki Task indexing statistics

Index #	Indexing times [min]		Index size [GiB]	Indexed formulae
	Wall Clock	CPU		
30	30	46	3.4	25,242,248
35	20	37	4.7	28,842,062
38	81	98	13	67,231,738
39	82	99	13	67,308,089

7. CONCLUSIONS AND FUTURE WORKS

The MIaS system is universal. We were able to participate in both Main and Wiki MathIR Tasks with no need of any reconfiguration or modification of our workflow except for indexing different set of input documents.

The main advances of our approach since NTCIR-11 Math Task were development and use of our evaluation platform based on NTCIR-11 ground truth and introduction of math structural unification component as part of the MIaS processing workflow. We considered the lack of ability of our system to match some types of structural differences of formulae as a weak point we tried to solve via MathML Unificator. However, performance of our system at NTCIR-12 MathIR Tasks did not met our expectations. We consider structural unification as a possible reason — use of structurally unified derivatives increases recall but has negative impact on precision. Setting and tuning of the indexing and preprocessing parameters is necessary for given application task.

We aim to again reuse NTCIR-12 MathIR judged data as ground truth in our evaluation platform to improve performance of our system. Fine tuning of weighting of structural unification products could possibly improve precision/recall ratio of our system.

We see possibilities of further improvement of canonicalization of math formulae. We are experimenting with the use of computer algebra systems (such as Maple or Mathematica) as another canonicalization step in math statements processing now.

Given insight we have got into MIR now, our future MathIR research targets to incorporate machine learning techniques to formulae disambiguation and ranking, and deploying CAS systems to open new possibilities of MIaS developments.

[6] http://www.html-tidy.org/

References

[1] Akiko Aizawa, Michael Kohlhase, Iadh Ounis, and Moritz Schubotz. "NTCIR-11 Math-2 Task Overview". In: *NTCIR Workshop 11 Meeting*. Tokyo, Japan, 2014.

[2] David Formánek, Martin Líška, Michal Růžička, and Petr Sojka. "Normalization of Digital Mathematics Library Content". In: *Joint Proceedings 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*. (Bremen, Germany, 2012-07-09/2012-07-13). Ed. by James Davenport, Johan Jeuring, Christoph Lange, and Paul Libbrecht. CEUR Workshop Proceedings 921. http://ceur-ws.org/Vol-921/wip-05.pdf. Aachen, 2012, pp. 91–103.

[3] Martin Líška, Petr Sojka, and Michal Růžička. "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. Melbourne, Australia: ACM, 2015, pp. 7–9. ISBN: 978-1-4503-3789-2. DOI: 10.1145/2810355.2810359. URL: http://doi.acm.org/10.1145/2810355.2810359.

[4] Martin Líška, Petr Sojka, and Michal Růžička. "Similarity Search for Mathematics: Masaryk University team at the NTCIR-10 Math Task". eng. In: *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*. Ed. by Noriko Kando and Kazuaki Kishida. http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings10/pdf/NTCIR/MATH/06-NTCIR10-MATH-LiskaM.pdf. Tokyo: National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan, 2013, pp. 686–691. ISBN: 978-4-86049-062-1.

[5] Michal Růžička, Petr Sojka, and Martin Líška. "Math Indexer and Searcher under the Hood: History and Development of a Winning Strategy". In: *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*. Ed. by Hideo Joho and Kazuaki Kishida. Tokyo: National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430 Japan, 2014-12. ISBN: 978-4-86049-065-2. URL: http://research.nii.ac.jp/ntcir/workshop/OnlineProceedings11/pdf/NTCIR/Math-2/07-NTCIR11-MATH-RuzickaM.pdf.

[6] Petr Sojka and Martin Líška. "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*. Ed. by James H. Davenport, William M. Farmer, Josef Urban, and Florian Rabe. Vol. 6824. Lecture Notes in Artificial Intelligence, LNAI. http://dx.doi.org/10.1007/978-3-642-22673-1_16. Bertinoro, Italy: Springer-Verlag, 2011-07, pp. 228–243.

[7] Petr Sojka and Martin Líška. "The Art of Mathematics Retrieval". In: *Proceedings of the ACM Conference on Document Engineering, DocEng 2011*. Mountain View, CA: Association of Computing Machinery, 2011-09, pp. 57–60. ISBN: 978-1-4503-0863-2. DOI: 10.1145/2034691.2034703. URL: http://doi.acm.org/10.1145/2034691.2034703.

[8] O. Tange. "GNU Parallel – The Command-Line Power Tool". In: *;login: The USENIX Magazine* 36.1 (2011-02). http://www.gnu.org/s/parallel, pp. 42–47.

[9] Krzysztof Wojciechowski, Aleksander Nowiński, Petr Sojka, and Martin Líška. *The EuDML Search and Browsing Service - Final*. 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. 2013-02.

[10] Richard Zanibbi et al. "NTCIR-12 MathIR Task Overview". In: *NTCIR*. National Institute of Informatics (NII), 2016.