TUW-IMP at the NTCIR-11 Math-2

Aldo Lipani, Linda Andersson, Florina Piroi, Mihai Lupu, Allan Hanbury
Vienna University of Technology, Austria
surname@ifs.tuwien.ac.at

ABSTRACT

The TUW-IMP team participated in the NTCIR-11 Math-2 task for retrieving mathematical formulae in scientific documents. This report describes our approach to solving the given math retrieval problem.

Team Name

TUW-IMP

Keywords

Math Information Retrieval

1. INTRODUCTION

The problem of retrieving mathematical content from a collection of documents is a typical domain specific information retrieval situation where out-of-the-box retrieval solutions and general purpose search engines are going to surely fail in IR experiments. The NTCIR11-Math tasks [3, 4] give researchers the possibility to test their IR systems on a test collection with mathematical content and with very specific information needs.

In this report we describe the IR system developed by us to be used on a collection of scientific documents. The implementation of the retrieval system has been done using Lucene 4.6 [1].

The description of the system follows the IR workflow, starting with the description of the document and query preprocessing components, continuing with the indexing and scoring components, result re-ranking components, and concluding with the result merging solution.

2. DOCUMENT PREPROCESSING

The collection provided by NTCIR for the Math challenge consists of paragraphs of documents containing text and, in most of the cases, mathematical formulae. To take up on this multi-modal retrieval challenge we aim to generate two sets of indices, one for text retrieval and one for formula retrieval.

2.1 Text Preprocessing

The first observation that can be made from the analysis of the given topics, is that most of the keywords in the topics refer to either well-known proper names of mathematicians ('Riemann', 'Lebesgue'), or to mathematical entities names ('Diophantine equations', 'Quantum Field Theory', 'hyperelliptic surface'). This pointed us to using a non-aggressive stemming algorithm in processing the text in the NTCIR-11 Math collection. We implemented, thus, an English Minimal Analysed using Lucene's `StandardTokenizer`, `LowerCaseFilter`, `StopAnalyser`, `EnglishMinimalStemFilter`. The English Minimal Stem Filter is a gentle stemmer that is applied to plural forms and third tense verbs only.

2.2 Formula Preprocessing

The NTCIR-11 Math collection is stored as XHTML files. Although, XHTML is part of the family of the XML markup languages, therefore parsable by an XML parser, the errors given by the parser made us decide to use regular expressions in extracting formal content out of the documents (Listing 1),

Listing 1: Regex to extract formulae from text

```
<math(?:(?!</math>).)*</math>
```

After extracting the formulae, for each MathML formula we extracted its semantic section using the regular expression shown in listing 2.

Listing 2: Extract semantic content from formulae

```
("<semantics(?:(?!<annotation-xml).)*" +
"<annotation-xml").r
  .findFirstMatchIn(formula).get.toString
  .replace("annotation-xml", "/semantics>")
```

The mathematical formulae in the collection are stored as trees encoded as MathML expressions. To parse and tokenise them we use their tree-form encoding. We extract two sets of tokens from the formulas, using two different tokenisers: one that extracts all literals (we called this the `LiteralTokenizer`), and one that starting from the tree structure of the formulas, extracts and linearises them. The latter tokenizer slices a formula tree by levels and collapses, then, in a top-down manner, the nodes that have a common parent (we call it the `L1Tokenizer`). We can extend this tokenizer with the tokens extracted from the previous tokenizer including the children of each node (`L2Tokenizer`).

For example, the formula in topic NTCIR-Math2-5

$$S_{EH} = \frac{1}{G_3} \circ d^3x \sqrt{-g^{(3)}}$$

is stored as shown in listing 3. The output of the three tokenisers is shown in tables 1, 2, and 3.

Listing 3: NTCIR-Math2-5 topic formula

```
<apply>
  <eq>
  <apply>
    <csymbol>subscript</csymbol>
    <ci>S</ci>
    <apply>
      <times/>
      <ci>E</ci>
      <ci>H</ci>
    </apply>
  </apply>
  <apply>
    <times/>
    <apply>
      <divide/>
      <cn>1</cn>
      <apply>
        <csymbol>subscript</csymbol>
        <ci>G</ci>
        <cn>3</cn>
      </apply>
    </apply>
    <m:qvar/>
    <apply>
      <csymbol>superscript</csymbol>
      <ci>d</ci>
      <cn>3</cn>
    </apply>
    <ci>x</ci>
    <apply>
      <root/>
      <apply>
        <minus/>
        <apply>
          <csymbol >superscript</csymbol>
          <ci>g</ci>
          <cn>3</cn>
        </apply>
      </apply>
    </apply>
  </apply>
</apply>
```

token	token
–	1
S	G
E	3
H	∧
x	d
g	

Table 1: Output of the Literal Tokenizer

3. QUERY PREPROCESSING

To the keywords in the topics we added new terms by using the additional terms obtained by a hyponymy extraction process, as described in [5]. These terms are added to the query as follows:

$$qt_1 \vee ... \vee qt_n \vee$$
$$qp_{(1,1)} \barwedge qp_{(l_1,1)} \vee ... \vee qp_{(1,m)} \barwedge qp_{(l_m,m)} \vee$$
$$sst_1 \vee ... \vee sst_2 \vee$$
$$ssp_{(1,1)} \vee ssp_{(l_1,1)} \vee ... \vee ssp_{(1,m)} \vee ssp_{(l_m,m)}$$

Where the *qt*s are query terms taken from the NTCIR-Math2 topics (the keywords), the *qp*s are query noun phrases extracted from the NTCIR-Math2 topics, the *sst*s and *ssp*s are the set of terms for the query terms, and the set of noun phrases for the query noun phrase.

The symbol $\barwedge$ represents a parameter called *StrictText* (or *Strict Multi Word Entities*) that affects the multiword query entities. When it is set to `true` its interpretation is $\wedge$ (all terms have to be present in the document). If set to `false` its interpretation is $\vee$ (at least one term should be present in the document).

To match the formulas given in the topics with documents and formulas in the collection we generate new tokens replacing each `qvar` ($\square$) with all the following semantic MathML symbols: *cn*, `apply` ($\circ$), `ci`, `csymbol` ($\bullet$), `cerror` ($\times$). Then, these tokens are combined with the formula tokens obtained from the topic formula, obtained as described in the previous section. The combination of tokens is controlled by two parameters, *StrictLiteral* and *StrictNonLiteral*. The *StrictLiteral* parameter controls the combination of formula tokens given by the literal tokenizer, the *StrictNonLiteral* parameter controls the combination of the formula tokens given by the L1 and L2 tokenizers.

token	token
$\circ$	$\circ \cdot \square \cdot \circ \cdot x \cdot \circ$
$\circ = \circ$	$1/\circ$
$ci \bullet \circ$	$\circ \cdot \square \cdot \circ \cdot x \cdot \circ$
$ci \cdot ci$	G_3
$\circ \cdot \square \cdot \circ \cdot ci \cdot \circ$	d_3
$cn/\circ$	g^3
$ci \circ cn$	$E \cdot H$
$\sqrt{\circ}$	$-S_\circ$
$-\circ$	

Table 2: Output of the L1 Tokenizer

token
$ci \bullet \circ = \circ \cdot \square \cdot \circ \cdot ci \cdot \circ$
$ci \bullet (ci \cdot ci)$
$(cn/\circ) \cdot \square \cdot (ci \bullet cn) \cdot ci \cdot (\sqrt{\circ})$
$cn/(ci \bullet cn)$
$\sqrt{-\circ}$
$-(ci \bullet cn)$
$S_\circ = \circ \cdot \square \cdot \circ \cdot x \cdot \circ$
$S_{E \cdot H}$
$(1/\circ) \cdot \square \cdot (d^3)x \cdot (\sqrt{\circ})$
$1/G_3$
$\sqrt{-\circ}$
$-g^3$

Table 3: Output of the L2 Tokenizer

4. INDEXING AND SCORING

Using the document preprocessing described in section 2 we created four indices, one for text and three for formulas in the documents. The three formula indices use, each, one of the tokenizers described: Literal, L1, and L2.

The retrieval model we used is BM25 with the default parameters of Lucene.

The query generated from the Math2 topics are ran against the four indeces. The result set given using the text index is

a list of document ids (call it the *text run*), while the result sets given by the formula indeces are lists of formulae in documents (the *formula runs*). To merge these runs we extract all the formulas found in the documents of the *text run* and compare them with the formulas in the *formula runs*. Each formula extracted from the *text run* receives a score equal to 10% of the score of the retrieved document.

4.1 Result Set Re-ranking

Each index has a result reranker that can be enabled using the *Normalized* parameter. The effect of it is, first, to normalize the score of the retrieved formula by the distance between its size and the size of the query formula (in terms of number of tokens), and, second, to dampen the result scores, using a log function:

$$\frac{k}{\log(|\sum_{i=1}^{n} dtf_i - \sum_{j=1}^{m} qtf_j| + 1) + k}$$

where we set $k = 3$.

The result re-ranking that uses the formula above is applied to the *text runs* and all *formula runs*.

4.2 Merging and Re-ranking

All the result sets (formulas and text) are summed linearly, with a second re-ranking step that groups all formulas occuring in the same document, which in the final, submitted run, is assigned a score equal to the sum of the formula scores.

5. RUNS

The four runs we submitted to the NTCIR-Math2 challenge are obtained from our IR system using different parameter value combinations. For our submission, out of the many possible combinations of parameter values, we choose only the ones shown in table 4. Note that, for all runs, we required that the literals occuring in the query formula should be found in the retrieved formula as well.

The FLA run is obtained by using the weakest of the constraints set chosen by us, where retrieved formulas should contain all the literals in the formula part of the query, but no restriction is put on the textual part of the query nor on the non-literal part of the topic formula (i.e. operators). From here, we vary each of the other three parameters, and obtain the next tree submitted run as follows: Our second run, FLASM, is similar to the FLA run with the additional constraint that all formula symbols occuring in the query formula should occur in the retrieved formulas. The FLASL run is similar to the FLA run, with the additional constraint that in the textual part of the query we require that all query terms occur in the retrieved documents.

Finally, the FLAN run a is similar to the FLA run to which the normalization step described in section 4.1 is applied.

When the list of retrieved results after constraint application is shorter than 1000 documents, the difference up to 1000 is filled with result retrieved with no constraints applied.

5.1 Retrieval Performance

The original experiments submitted by our team to the NTCIR11-Math-2 task did not obtain scores to please us for our work. Upon closer examination, we found that our formula indeces actually miss almost 28% of the documents in the collection. Due to this error, almost 40% of the relevant documents (relevance level greater than 0) in the relevance judgement distributed by the task organizers could not be found by our system, since they were not in the indeces created by us.

	FLA	FLASM	FLASL	FLAN
SMWE	No	No	Yes	No
SL	Yes	Yes	Yes	Yes
SNL	No	Yes	No	No
N	No	No	No	Yes

Table 4: Constraint sets for the submitted runs. Strict Multi Words Entities (SMWE) Strict Literals (SL); Strict Non Literals (SNL); Normalized (N)

Because by the time of writing this report, reindexing the collection was not finished, we decided to run a second round of evaluation. We first filtered out the documents missing from our index from the relevance judgements (qrels) and from all participants' runs. Table 5 shows the percentage of removed retrieval results from the task runs. We then ran `trec_eval` on the filtered runs and qrels using the same evaluation parameters as the ones used by the task organizers.

Table 5: Removed retrieval results from participant experiments, in percentages. There was no content removed from the TUW experiments because of the incomplete index files.

Run name	% removed
FSE_{LATEX}	1.38
ICST_{PKU}	8.78
$\text{IFISB}_{QUALIBETA}$	17.12
$\text{KWARC}_{default}$	23.00
MCAT_{all}	17.35
$\text{MCAT}_{depdesc}$	17.43
$\text{MCAT}_{deprerank}$	15.95
$\text{MCAT}_{nodepctxt}$	17.58
MIRMU_{Cmath}	21.71
MIRMU_{PCMath}	21.96
MIRMU_{Pmath}	21.89
MIRMU_{TeX}	21.98
RIT_{mf}	28.63
RIT_{mo}	28.62
RIT_{mte}	27.65
RIT_{nd}	28.60
TUW-IMP_{FLA}	0.00
TUW-IMP_{FLAN}	0.00
TUW-IMP_{FLASL}	0.00
TUW-IMP_{FLASM}	0.00

Since the percentage of removed content is as high as 28% for some of the submitted experiments, we report here only scores for precision at 5 and 10. Table 6 shows official and our scores for the Precision at 5 and 10, partially relevant.

As seen in table 6, ranking did change in our favour for the partially relevant evaluations, but the experiment ranking for the relevant evaluation stayed the same (table 7).

6. CONCLUSION

The IR system we employed for our first participation to the pilot Math retrieval task is rather minimal, the method of formula token extraction being based on our intuition on how IR systems should view formal content. It remains to examine closer the impact of extending the query terms with hyponyms on the retrieval results. After examining the

Table 6: Partially relevant evaluation scores, ordered by the recomputed P at 10

	Re-evaluation scores		Official scores	
Run	P 10	P 5	P 10	P 5
MIRMU_{PCMath}	0.426	0.700	0.552	0.864
MIRMU_{Cmath}	0.420	0.704	0.544	0.872
MIRMU_{TeX}	0.414	0.680	0.540	0.848
MIRMU_{Pmath}	0.382	0.668	0.502	0.844
RIT_{mte}	0.382	0.664	0.546	0.924
$\text{KWARC}_{default}$	0.354	0.568	0.278	0.792
RIT_{nd}	0.330	0.540	0.456	0.648
TUW-IMP_{FLASL}	0.238	0.376	0.238	0.376
RIT_{mf}	0.220	0.392	0.292	0.484
MCAT_{all}	0.220	0.368	0.280	0.448
TUW-IMP_{FLA}	0.220	0.336	0.220	0.336
$\text{MCAT}_{deprerank}$	0.216	0.368	0.280	0.464
TUW-IMP_{FLASM}	0.216	0.348	0.216	0.348
TUW-IMP_{FLAN}	0.216	0.332	0.216	0.332
RIT_{mo}	0.214	0.388	0.266	0.464
$\text{MCAT}_{depdescr}$	0.206	0.336	0.260	0.416
$\text{MCAT}_{nodepctxt}$	0.186	0.316	0.240	0.404
$\text{IFISB}_{QUALIBETA}$	0.184	0.352	0.244	0.460
ICST_{PKU}	0.072	0.144	0.076	0.152
FSE_{LATEX}	0.008	0.016	0.008	0.016

Table 7: Relevant evaluation scores, ordered by the recomputed P at 10

	Re-evaluation scores		Official scores	
Run	P 10	P 5	P 10	P 5
MIRMU_{Cmath}	0.296	0.496	0.352	0.568
MIRMU_{PCMath}	0.292	0.476	0.348	0.556
MIRMU_{TeX}	0.282	0.464	0.338	0.540
MIRMU_{Pmath}	0.252	0.436	0.304	0.512
$\text{KWARC}_{default}$	0.250	0.424	0.306	0.500
RIT_{mte}	0.220	0.388	0.286	0.504
RIT_{nd}	0.200	0.340	0.268	0.384
RIT_{mf}	0.116	0.204	0.142	0.244
$\text{MCAT}_{deprerank}$	0.112	0.180	0.130	0.208
MCAT_{all}	0.108	0.180	0.124	0.212
$\text{MCAT}_{depdesc}$	0.104	0.172	0.116	0.192
RIT_{mo}	0.096	0.172	0.116	0.200
$\text{MCAT}_{nodepctxt}$	0.086	0.140	0.096	0.164
TUW-IMP_{FLASL}	0.084	0.128	0.084	0.128
TUW-IMP_{FLA}	0.080	0.116	0.080	0.116
TUW-IMP_{FLASM}	0.078	0.120	0.078	0.120
TUW-IMP_{FLAN}	0.074	0.120	0.074	0.120
$\text{IFISB}_{QUALIBETA}$	0.048	0.084	0.062	0.108
ICST_{PKU}	0.018	0.036	0.018	0.036
FSE_{LATEX}	0.006	0.012	0.006	0.012

relevance judgements, it is clear that in the assessors' information need, the topic query words clearly carry a weight at least as important as the topic formulae. In our IR model, keywords were given, implicitly, a lower weight than the tokens extracted from formulas. Further more, we believe that some form of unification between the formulas retrieved from the collection and the topic formulas is necessary. Only one participating team—Kwarc, [6]—actually used unification in the mathematical sense, while the other teams used, quite successfully, matching algorithms [8, 7]. Our intention to apply a unification algorithm was hindered by the difficulty of extracting valid formal expressions, both from the topics and from the formulas in the collection documents, to be passed to a unification algorithm. Time constraints did not permit us to pursue this avenue for this task participation. It is clear to us, though, that semantic errors of content representations, like the 'Imaginary' operator Im being expressed as a product of I and m [7], cannot be completely eliminated from a collection of documents containing mathematical formulas, as these documents are created using different document editors and use different representation standards. Conversion to XHTML and MathML formats is obviously not flawless.

Acknowledgment

This research was partly funded by the Austrian Science Fund (FWF) project number P25905-N23 (ADmIRE).

7. REFERENCES

[1] Apache Lucene. `http://lucene.apache.org/`, last retrieved: September 2014.

[2] ????, editor. *NTCIR Workshop 11 Meeting*, Tokyo, Japan, 2014.

[3] A. Aizawa, M. Kohlhase, and I. Ounis. NTCIR-10 math pilot task overview. In *Proceedings of the 10th NTCIR Conference, Tokyo, Japan*, 2013.

[4] A. Aizawa, M. Kohlhase, and I. Ounis. NTCIR-11 math-2 task overview. In *Proceedings of the 11th NTCIR Conference, Tokyo, Japan*, 2014.

[5] L. Andersson, M. Lupu, J. Palotti, F. Piroi, A. Hanbury, and A. Rauber. Insight to hyponymy lexical relation extraction in the patent genre versus other text genres. In *First International Workshop on Patent Mining and Its Applications, IPaMin 2014*.

[6] M. Kohlhase, R. Hambasan, and C.-C. Prodescu. MathWebSearch at NTCIR-11. In ???? [2].

[7] P. S. Michal Ružička and M. Líška. Math indexer and searcher under the hood: History and development of a winning strategy. In ???? [2].

[8] N. Pattaniyil and R. Zanibbi. Combining tf-idf text retrieval with an inverted index over symbol pairs in math expressions: The tangent math search engine at NTCIR 2014. In ???? [2].