The MCAT Math Retrieval System for NTCIR-10 Math Track

Goran Topić  Giovanni Yoko Kristianto  Minh-Quoc Nghiem
Akiko Aizawa

National Institute of Informatics

2013
Introduction
Unnecessary
Objective

Extract descriptions of math expressions from the text
Objective

Extract **descriptions** of math expressions from the text

Assumption

Descriptions are **noun phrases in the same sentence** as the math expression
**Objective**

Extract *descriptions* of math expressions from the text

**Assumption**

Descriptions are *noun phrases* in the same *sentence* as the math expression

**Method**

Identify noun phrases using *Stanford parser*

Train a linear-kernel *SVM classifier* for (math, description) pairs
<table>
<thead>
<tr>
<th>Feature</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apposition?</td>
<td>set $\mathbb{N}$</td>
</tr>
<tr>
<td>Colon?</td>
<td>set: $\mathbb{N}$</td>
</tr>
<tr>
<td>Comma?</td>
<td>set, $\mathbb{N}$</td>
</tr>
<tr>
<td>Intervening expression?</td>
<td>set $\ldots A \ldots \mathbb{N}$</td>
</tr>
<tr>
<td>Parenthetical?</td>
<td>$\mathbb{N}$ (set)</td>
</tr>
<tr>
<td>Word distance</td>
<td>set (4 words) $\mathbb{N}$</td>
</tr>
<tr>
<td>After description?</td>
<td>set $\mathbb{N}$</td>
</tr>
<tr>
<td>2-word description context</td>
<td>in/IN the/DT set/NN $\mathbb{N}$/MATH of/IN</td>
</tr>
<tr>
<td>3-word expression context</td>
<td>in/IN the/DT set/NN $\mathbb{N}$/MATH of/IN natural/JJ numbers/NNS</td>
</tr>
<tr>
<td>First word of description</td>
<td>set/NN</td>
</tr>
<tr>
<td>Last word of description</td>
<td>set/NN</td>
</tr>
<tr>
<td>Unigrams</td>
<td>in/IN, the/DT, set/NN, $\mathbb{N}$/MATH, of/IN, natural/JJ, numbers/NNS</td>
</tr>
<tr>
<td>Bigrams</td>
<td>in/IN the/DT, the/DT set/NN, set/NN $\mathbb{N}$/MATH, $\mathbb{N}$/MATH of/IN</td>
</tr>
<tr>
<td>Trigrams</td>
<td>in/IN the/DT set/NN, set/NN $\mathbb{N}$/MATH of/IN</td>
</tr>
<tr>
<td>First intervening verb</td>
<td>set $\ldots$ shows $\ldots$ $\mathbb{N}$</td>
</tr>
</tbody>
</table>
Baseline

Noun phrase in **apposition** to the mathematical expression
## Experiment

### Baseline

Noun phrase in *apposition* to the mathematical expression

### Runs

- Full descriptions, *all* features
- Full descriptions, *without apposition* feature
- Short descriptions, *all* features
- Short descriptions, *without apposition* feature
## Results

<table>
<thead>
<tr>
<th>Run</th>
<th>P</th>
<th>R</th>
<th>F-1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Strict Matching Evaluation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full, baseline</td>
<td>43.10</td>
<td>23.85</td>
<td>30.96</td>
</tr>
<tr>
<td>MCAT: full, all features</td>
<td>61.94</td>
<td>37.03</td>
<td>46.35</td>
</tr>
<tr>
<td>MCAT: full, no apposition</td>
<td>61.92</td>
<td>37.33</td>
<td>46.58</td>
</tr>
<tr>
<td>short, baseline</td>
<td>55.35</td>
<td>29.94</td>
<td>38.86</td>
</tr>
<tr>
<td>MCAT: short, all features</td>
<td>68.24</td>
<td>40.42</td>
<td>50.77</td>
</tr>
<tr>
<td>MCAT: short, no apposition</td>
<td>67.67</td>
<td>40.22</td>
<td>50.45</td>
</tr>
<tr>
<td><strong>Soft Matching Evaluation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>full, baseline</td>
<td>64.21</td>
<td>34.73</td>
<td>45.08</td>
</tr>
<tr>
<td>MCAT: full, all features</td>
<td>85.48</td>
<td>47.41</td>
<td>61.24</td>
</tr>
<tr>
<td>MCAT: full, no apposition</td>
<td>87.25</td>
<td>48.30</td>
<td>62.18</td>
</tr>
<tr>
<td>short, baseline</td>
<td>64.21</td>
<td>34.73</td>
<td>45.08</td>
</tr>
<tr>
<td>MCAT: short, all features</td>
<td>81.68</td>
<td>42.81</td>
<td>56.18</td>
</tr>
<tr>
<td>MCAT: short, no apposition</td>
<td>81.24</td>
<td>42.61</td>
<td>55.90</td>
</tr>
</tbody>
</table>
Apposition feature slightly harms full description extraction (because of discontinuous descriptions), but helps short description extraction (since they mostly are appositions). Unsurprisingly, soft matching works better on full descriptions than on short descriptions (larger text region: more chances of overlap). In strict matching runs, 64% of full and 75% of short descriptions appear as noun phrases. In soft matching runs, 89% of full and 87% of short descriptions appear as noun phrases.
Objective

Provide a search system for mathematical expressions
Objective
Provide a search system for mathematical expressions

Requirements
Flexibility: focus on recall (sacrifice precision)
Encode structure as well as tokens
Allow full-text search on descriptions
Objective

Provide a search system for mathematical expressions

Requirements

Flexibility: focus on recall (sacrifice precision)
Encode structure as well as tokens
Allow full-text search on descriptions

Engine

Starting with full-text search requirement, we selected Apache Solr (Lucene) for our database.
Papers are placed in a separate index, with full-text, language-dependent fields for titles and abstracts, and general fields for authors.
Indexing

Papers
Papers are placed in a separate index, with full-text, language-dependent fields for titles and abstracts, and general fields for authors.

Expressions
Expression descriptions are indexed as full-text, language-dependent fields. Expression MathML structure is encoded into three whitespace-separated, multivalued fields: ordered paths, unordered paths and sisters.
Indexing

Papers

Papers are placed in a separate index, with full-text, language-dependent fields for titles and abstracts, and general fields for authors.

Expressions

Expression descriptions are indexed as full-text, language-dependent fields. Expression MathML structure is encoded into three whitespace-separated, multivalued fields: ordered paths, unordered paths and sisters.

A Note on Descriptions

Due to time constraints, we did not use the extracted descriptions, but used a fixed 10-word context instead.
Indexing

Papers

Papers are placed in a separate index, with full-text, language-dependent fields for titles and abstracts, and general fields for authors.

Expressions

Expression descriptions are indexed as full-text, language-dependent fields. Expression MathML structure is encoded into three whitespace-separated, multivalued fields: ordered paths, unordered paths and sisters.

A Note on Descriptions

Due to time constraints, we did not use the extracted descriptions, but used a fixed 10-word context instead.

Key Fields

(Additionally, there are various primary and foreign keys.)
Ordered Paths

Encode all *vertical paths* to the leaves, including the left-to-right position of each node
(Equivalently, encode all vertical paths from root to leaves, repeat recursively for each non-trivial subtree)
Order Paths

Encode all **vertical paths** to the leaves, including the left-to-right position of each node
(Equivalently, encode all vertical paths from root to leaves, repeat recursively for each non-trivial subtree)

Unordered Paths

Same, but without the position information
Expression Indexing

Ordered Paths
Encode all vertical paths to the leaves, including the left-to-right position of each node
(Equivalently, encode all vertical paths from root to leaves, repeat recursively for each non-trivial subtree)

Unordered Paths
Same, but without the position information

Sisters
Collection of sister nodes in the same subtree
Example Encoding

Expression

the polynomial $\sum_{i=1}^{n} a_i x^i$
Example Encoding

Expression

the polynomial $\sum_{i=1}^{n} a_i x^i$

MathML Tree

$$\begin{array}{c}
\text{mrow} \\
\text{msubsup} \\
\text{mo} \\
\text{mi} \\
\Sigma \\
i \\
= \\
1 \\
\text{mn} \\
n \\
\text{mi} \\
a \\
\text{mi} \\
i
\end{array}$$
The polynomial \( \sum_{i=1}^{n} a_i x^i \)
Example Encoding

**Encoding**

**opaths:** 1#msubsup 1#1#mo#Σ 1#2#1#mi#i 1#2#2#mo#= 1#2#3#mn#1

1#3#mi#n 2#msub 2#1#mi#a 2#2#mi#i

**opaths:** msubsup 1#mo#Σ 2#1#mi#i 2#2#mo#= 2#3#mn#1 3#mi#n

**opaths:** 1#mi#i 2#mo#= 3#mn#1

**opaths:** msub 1#mi#a 2#mi#i

**upaths:** #msubsup ##mo#Σ ###mi#i ###mo#= ###mn#1 ##mi#n msub

##mi#a ##mi#i

**upaths:** msubsup #mo#Σ ##mi#i ##mo#= ##mn#1 #mi#n

**upaths:** #mi#i #mo#= #mn#1

**upaths:** msub #mi#a #mi#i

**sisters:** mi#i mo#= mn#1

**sisters:** mo#Σ mi#n

**sisters:** mi#a mi#i

**sisters:** msubsup msub

**sisters:** msubsup msub

**description_en:** the polynomial (indexed as: polynomi)
Querying

Encode the query MathML and description (whichever is provided) in the same way.
Perform a disjunctive query on Lucene.
Lucene scores matching terms, modified by tf/idf and length normalization.
Querying

Encode the query MathML and description (whichever is provided) in the same way.
Perform a disjunctive query on Lucene.
Lucene scores matching terms, modified by tf/idf and length normalization.

Document Queries

Document queries (with author, title or abstract terms) are done using a Lucene join query.
## Querying

Encode the query MathML and description (whichever is provided) in the same way.

Perform a disjunctive query on Lucene.

Lucene scores matching terms, modified by tf/idf and length normalization.

### Document Queries

Document queries (with author, title or abstract terms) are done using a Lucene join query.

### Jokers

Jokers implemented by simply leaving off the term from the query. Alternately, due to the flexibility of the encoding, a single wrong leaf will only slightly penalize the score, not reject the result.
## Results

<table>
<thead>
<tr>
<th></th>
<th>P-10 avg</th>
<th>P-5 avg</th>
<th>MAP avg</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Formula Search</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevant</td>
<td>0.229</td>
<td>0.219</td>
<td>0.162</td>
<td>0.065</td>
</tr>
<tr>
<td>Partially Relevant</td>
<td>0.500</td>
<td>0.476</td>
<td>0.379</td>
<td>0.220</td>
</tr>
<tr>
<td><strong>Fulltext Search</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevant</td>
<td>0.293</td>
<td>0.320</td>
<td>0.297</td>
<td>0.103</td>
</tr>
<tr>
<td>Partially Relevant</td>
<td>0.660</td>
<td>0.680</td>
<td>0.534</td>
<td>0.309</td>
</tr>
</tbody>
</table>
Conclusions

The current approach achieves satisfactory recall (for the first effort). Precision is low, as there is no clear cutoff where the results stop being relevant. Term-based search somewhat mitigates inconsistencies in representation.
Conclusions

The current approach achieves **satisfactory recall** (for the first effort). **Precision is low**, as there is no clear cutoff where the results stop being relevant. Term-based search somewhat **mitigates inconsistencies** in representation.

Overall Conclusion

The approach, although simple, seems fruitful, and we intend to continue refining it.
Future Work

Normalization of commonly interchangeable MathML elements.
Usage of Content MathML instead of Presentation MathML.
Giving more weight to operators and structure.
Implementation of common subexpression unification rules, which would additionally penalize the results where the instances of the same subexpression are replaced by different subexpressions.
Restriction of the number of disjunct terms, since their number adversely impacts search times.
Usage of actual pq-grams.
Usage of extracted descriptions.
Post-search reordering of top results using a more precise similarity measure.
Extraction of more advanced features for the math understanding subtask, such as information from dependency trees.
Thank you for your attention