The MCAT Math Retrieval System for NTCIR-10 Math Track

NII MCAT team (`mathcat@nii.ac.jp`): Goran Topić, Giovanni Yoko Kristianto, Minh-Quoc Nghiem, Akiko Aizawa

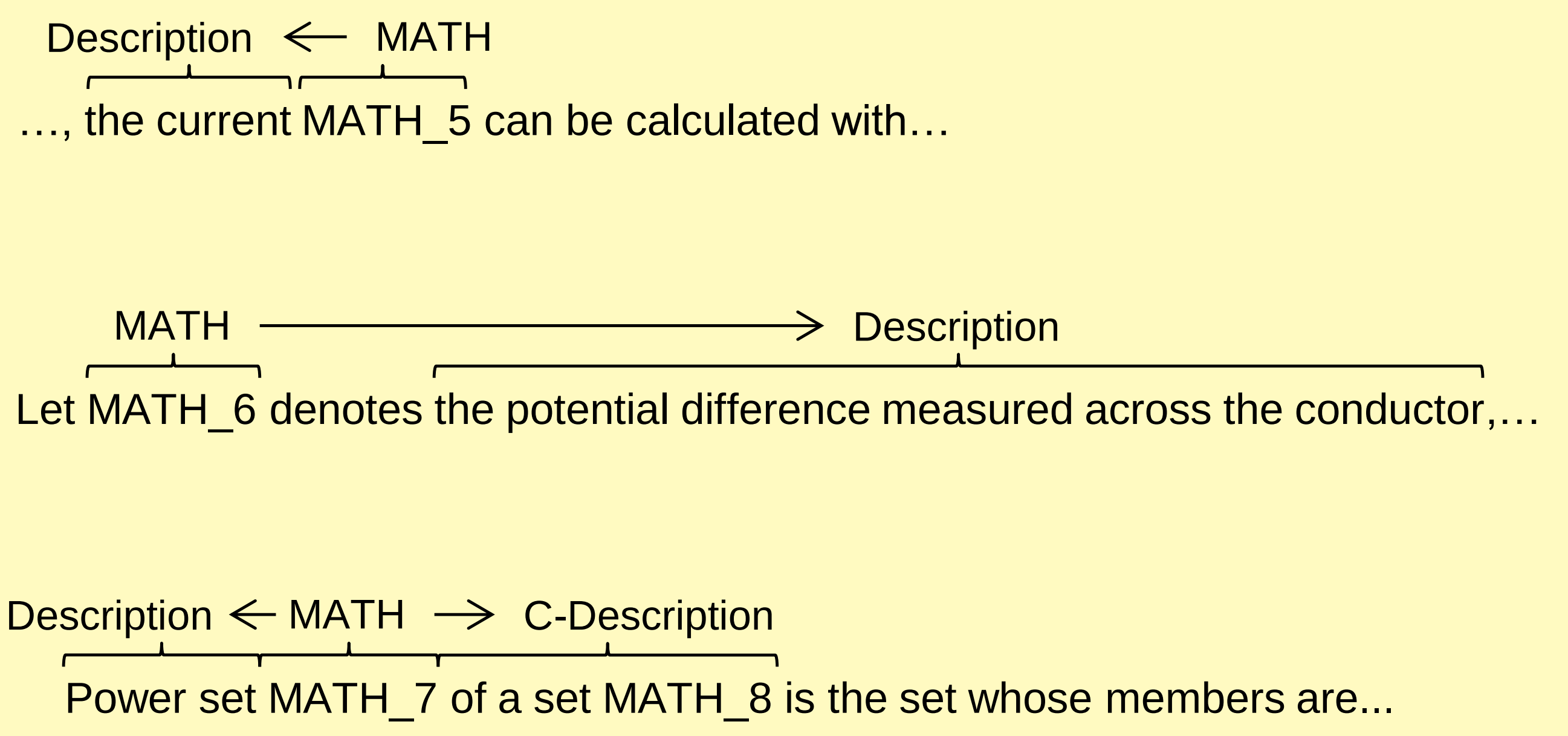
Summary

NTCIR Math Track targets mathematical content access based on both natural language text and mathematical formulae. This research describes the participation of MCAT group in the NTCIR math retrieval subtask and math understanding subtask. We introduce our mathematical search system that is capable of formula search, and full-text search. We also introduce our mathematical description extraction system which was based on a support vector machine model. Experimental results show that our general-purpose search engine can work reasonably well with math queries.

Math Understanding Subtask

Extracting textual descriptions of mathematical formulas in documents.

Annotation



Description ← MATH

..., the current MATH_5 can be calculated with...

MATH → Description

Let MATH_6 denotes the potential difference measured across the conductor,...

Description ← MATH → C-Description

Power set MATH_7 of a set MATH_8 is the set whose members are...

Description Extraction Method

Apposition?	set ℕ
Colon?	set: ℕ
Comma?	set, ℕ
Intervening expression?	set . . . A . . . ℕ
Parenthetical?	ℕ (set)
Word distance	set *(4 words)* ℕ
After description?	set ℕ
2-word description context	in/IN the/DT **set/NN** ℕ/MATH of/IN
3-word expression context	in/IN the/DT set/NN **ℕ/MATH** of/IN natural/JJ numbers/NNS
First word of description	set/NN
Last word of description	set/NN
Unigrams	in/IN, the/DT, set/NN, ℕ/MATH, of/IN, natural/JJ, numbers/NNS
Bigrams	in/IN the/DT, the/DT set/NN, set/NN ℕ/MATH, ℕ/MATH of/IN
Trigrams	in/IN the/DT set/NN, set/NN ℕ/MATH of/IN
First intervening verb	set . . . shows . . . ℕ

- *Nearest noun method (baseline)* defines a description as a combination of adjectives and nouns in text preceding target expressions.
- *Machine learning method*: SVM using features shown in the Table.
- *Four runs*: short and long desc, using apposition feature or not

Results

- Inclusion of the apposition feature slightly reduces the performance of full desc extraction. The opposite happens for short descs.
- Full desc contains longer text regions than short ones, thus chance of partial (soft) hit in full desc is higher than in short desc.
- In strict match, there are 64% of full and 75% of short desc that are NP. In soft match, it is 89% of full and 87% of short.

Future Work

- Extend our method using the current system as a baseline.
- 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.
- Reranking 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.

Math Retrieval Subtask

Indexing the mathematical formulae

Our indexing is similar to pq-gram method, but unlike in pq-grams, the structure of the tree is encoded in several Lucene fields:

- `opaths`: all vertical paths in the tree, specifying for each node its position among sisters
- `upaths`: all vertical paths, without the position information
- `sisters`: all non-trivial collections of sisters
- This is repeated for all non-trivial subtrees
- Each path is a space-delimited term; each subtree is one value in a multi-valued Lucene field

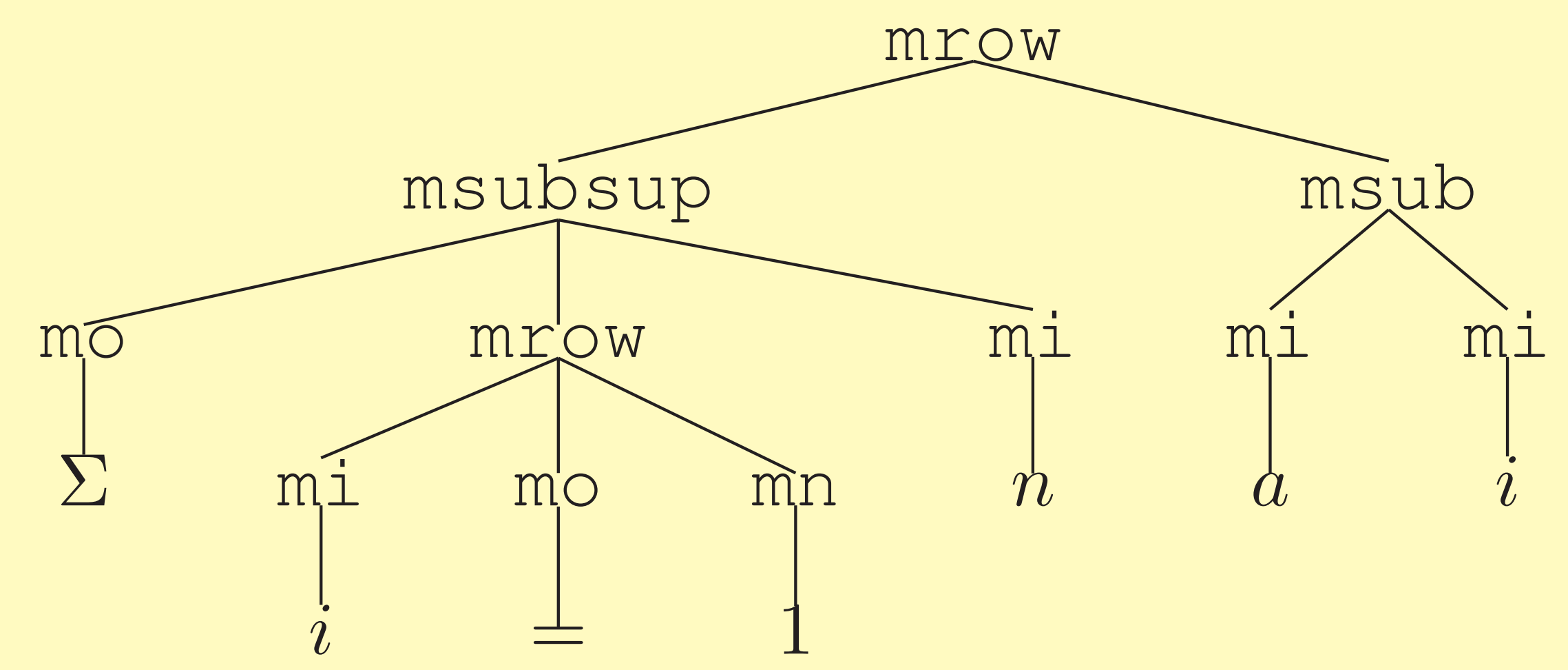
Indexing the natural language descriptions

- In addition, there are full-text fields for expression descriptions, processed according to the language (word segmentation, stemming...) **Note:** The NTCIR-Math results were produced using 10-word context of the expressions instead the extracted descriptions.

At query time

- Extract `opaths`, `upaths` and `sisters` from the query expression
- Perform a Lucene query, scoring by matching terms

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



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`

description_en: `the polynomial` (indexed as: `polynomi`)

Conclusion

In this research, we have presented MCAT's submissions to the NTCIR Math Task. For the math retrieval subtask, we have introduced `opaths`, `upaths` for indexing and a modified TF/IDF score for ranking. For the math understanding subtask, we have proposed an SVM classification to detect descriptions of mathematical expressions. Although our work is still at a preliminary stage, the results showed that a general-purpose search engine can work reasonably well with math queries.