

The MCAT Math Retrieval System for NTCIR-11 Math Track

NII MCAT team (mathcat@nii.ac.jp): Giovanni Yoko Kristianto, Goran Topić, Florence Ho, Akiko Aizawa

Summary

We introduce an encoding technique to capture the structure and content of mathematical expressions. We associate each mathematical expression with two types of automatically extracted textual information, namely words in context window and descriptions. We also introduce dependency graph and post-retrieval reranking methods to improve the performances of our mathematical search system.

Extracting Textual Information for Math Formulae

Context window for a mathematical expression consists of ten words preceding and following the expression.

Example of context:

MATH

Context

The notation T refers to a set of topics, V to the word vocabulary and $g_i(\alpha_i)$ to the Dirichlet distribution associated with topic t_i .

Descriptions for each mathematical expression is extracted using binary classification method (SVM) by taking all noun phrases as description candidates and using features as follows.

Matched sentence patterns?	\mathbb{N} is set
Apposition?	set \mathbb{N}
Colon?	set: \mathbb{N}
Comma?	set, \mathbb{N}
Intervening expression?	set ... A ... \mathbb{N}
Parenthetical?	\mathbb{N} (set)
Word distance	set (4 words) \mathbb{N}
After description?	set \mathbb{N}
2-word description context	in/IN the/DT set/NN \mathbb{N} /MATH of/IN
3-word expression context	in/IN the/DT set/NN \mathbb{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, \mathbb{N} /MATH, of/IN, natural/JJ, numbers/NNS
Bigrams	in/IN the/DT, the/DT set/NN, set/NN \mathbb{N} /MATH, \mathbb{N} /MATH of/IN
Trigrams	in/IN the/DT set/NN, set/NN \mathbb{N} /MATH of/IN
First intervening verb	set ... shows ... \mathbb{N}
Distance in predicate-arg.	set (4 hops) \mathbb{N}
<i>Several other features based on predicate argument structure</i>	

Example of description:

MATH

Description

The notation T refers to a set of topics, V to the word vocabulary and $g_i(\alpha_i)$ to the Dirichlet distribution associated with topic t_i .

Performance of description extraction

- Exact match (Precision, Recall, F-1) : 73.72%, 45.88%, 41.40%
- Partial match (Precision, Recall, F-1): 80.80%, 72.77%, 76.58%

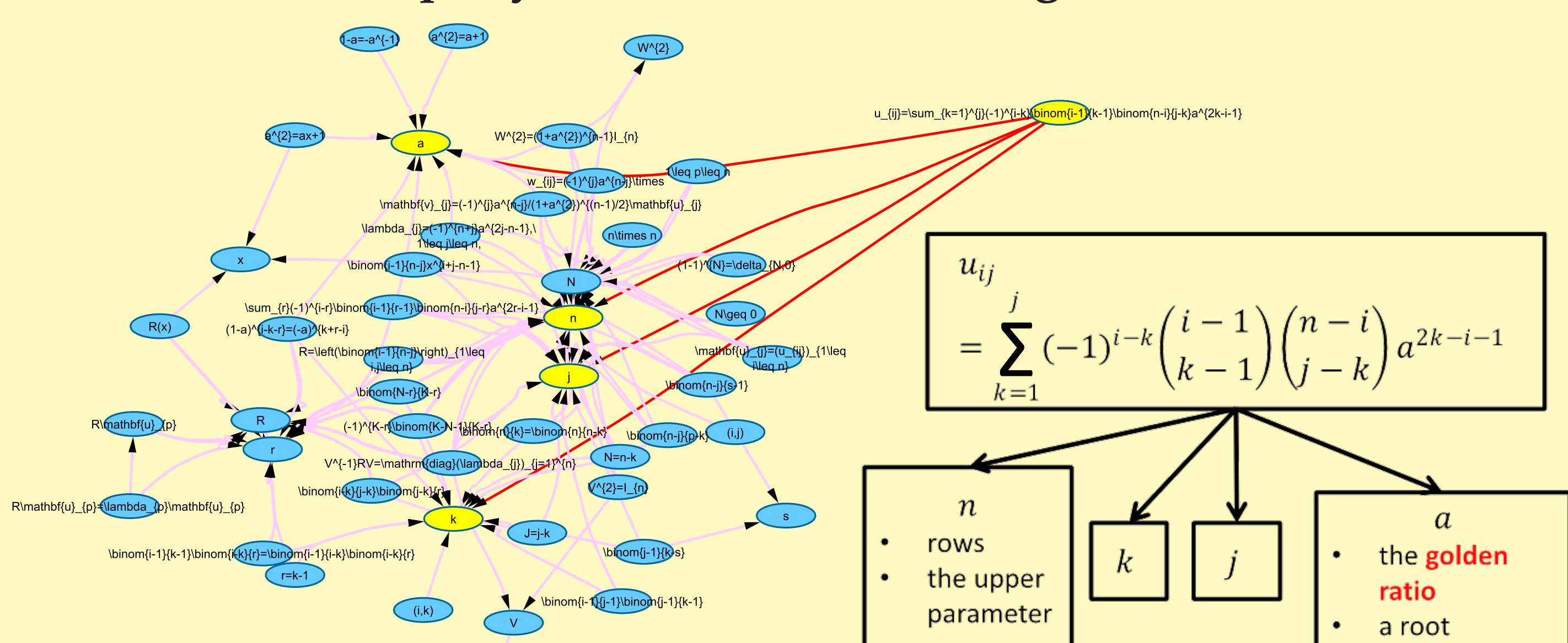
Dependency Graph

A dependency graph is defined as follows.

- a directed graph
- each vertex represents a distinctive math expression
- an edge from vertex $mathexp_1$ to $mathexp_2$ indicates $mathexp_1$ (parents) contains $mathexp_2$ (child)

Descriptions from children can be useful to represent the parent.

Example: expression $u_{ij} = \sum_{k=1}^j (-1)^{i-k} \binom{i-1}{k-1} \binom{n-i}{j-k} a^{2k-i-1}$ which is relevant to a query $x^2 - x - 1 = 0$ (golden ratio; Fibonacci)



Indexing Math Formulae and Textual Information

Indexing the mathematical formulae

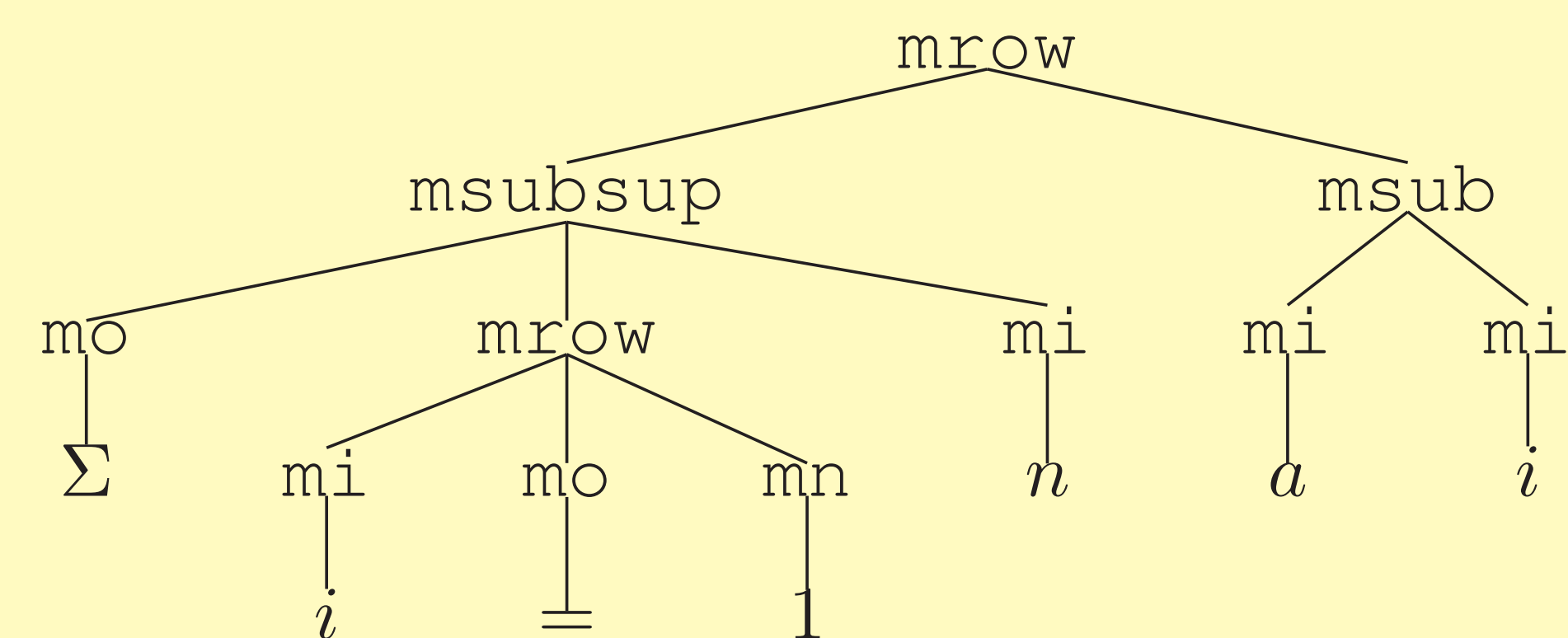
The structure of MathML tree is encoded in several Lucene fields:

- 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

Indexing the natural language descriptions

- There are also full-text fields for expression descriptions, processed according to the language (word segmentation, stemming...)

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

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

sisters: mi#i mo#= mn#1

sisters: mo#Σ mi#n

description_en: the polynomial (indexed as: polynomi)

Post-Retrieval Reranking

Given a query formula f_q and each formula f_x in the retrieved list, we calculate a score $formSim$.

$$formSim(f_x, f_q) = \lambda \cdot constSim(f_x, f_q) + (1 - \lambda) \cdot structSim(f_x, f_q)$$

where

- λ is weighting parameter ($[0 \dots 1]$).
- $constSim$ is content similarity measure (operators, numeric literal, and identifiers) and is computed using Euclidean distance.
- $structSim$ is structure similarity (distinct paths from the MathML representation) computed using Euclidean distance.

The $formSim$ scores are used to weight the search scores and thus rerank the result.

Results

Run	High Relevancy			Partial Relevancy		
	P@5	P@10	MAP	P@5	P@10	MAP
nodep-context	.1640	.0960	.0515	.4040	.2400	.0776
dep-descriptions	.1920	.1160	.0648	.4160	.2600	.0860
all_text	.2120	.1240	.0718	.4480	.2800	.0926
dep-rerank	.2080	.1300	.0742	.4640	.2800	.0933

Description extraction time: 360 hours (approx.)

Indexing time: 48 hours (40h encoding + 8h importing data to Solr)

Solr Index size (on disk): 58Gb

Average query answer time: 55,262 ms.