Joint word embeddings & soft cosine measure at ARQMath

Math StackExchange *singing and dancing* ♫

Vít Novotný

witiko@mail.muni.cz

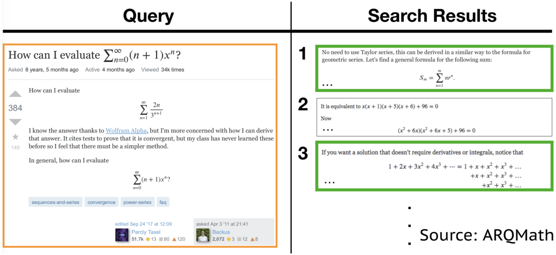
Faculty of Informatics, Masaryk University

April 8, 2020

Introduction

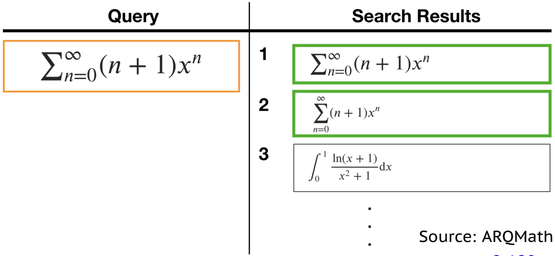
Answer Retrieval for Questions on Math (ARQMath) is a competition aimed at building a math-aware search engine for **Math StackExchange**. ARQMath consists of two tasks:

Task 1: Find Answers Given a posted question as a query, search all answer posts and return relevant answers.



Source: ARQMath

Task 2: Formula Search Given a formula from a question as a query, search all question and answer posts for relevant formulae.



Source: ARQMath

Challenges

Compared to standard question answering and similarity search tasks, task 2 requires a representation of mathematical formulae, and task 1 requires a representation of mathematical documents that contain both text and mathematical formulae.

Source: formula1.com

Our search engines

The **Math Information Retrieval (MIR-MU)** research group has prepared several mathematical search engines that will compete in ARQMath:

CompuBERT (task 1) Search engine with a BERT-based [1] representation of both text and LaTeX mathematical formulae (preprocessed with the WordPiece tokenizer [2]). English BERT model is fine-tuned by learning to rank Math StackExchange answers by number of votes.

Joint Word Embeddings & Soft Cosine Measure (SCM) (tasks 1 and 2) Search engine with a tf-idf soft vector space model (VSM) [3, 4] representation of both text and mathematical formulae. Various representations of mathematics are used.

Formula2Vec (tasks 1 and 2) Same as above, but with Doc2Vec [5] representation.

Math Indexer and Searcher (MIaS) (tasks 1 and 2) Search engine with a tf-idf VSM representation of both text and MathML mathematical formulae. Deployed in the **European Digital Mathematical Library (EuDML)** since 2013. Entered three competitions during 2013–2016, earning medal-winning results. [6]

Representations of mathematics

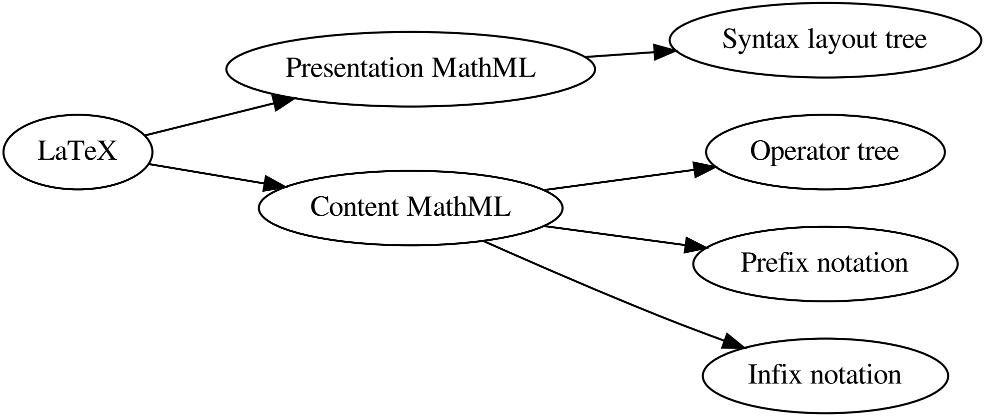


Figure: Preprocessing of input LaTeX mathematical formulae into output representations

LaTeX

Math StackExchange questions, answers, and comments are provided in the same format in which the users type them, in LaTeX:

LaTeX representation of formula $x!! - y^2 = 0$

```
$x!!-y^2=0$
```

LaTeX representation is easy to type, but it is purely syntactic. Since LaTeX is Turing-complete, static tokenization is difficult with complex LaTeX commands.

MathML

Presentation MathML (PMML)

Using LaTeXML [7] and MathML Canonicalizer [8], mathematical formulae in LaTeX are converted to the MathML 3.0 [9] XML language:

PMML representation of formula $x!! - y^2 = 0$

```
<mrow>
  <mi>x</mi><mo>!!</mo>
  <msup>
    <mi>y</mi><mn>2</mn>
  </msup>
  <mo>=</mo><mn>0</mn>
</mrow>
```

Presentation MathML is still purely syntactic, but it solves tokenization of LaTeX.

MathML

Content MathML (CMML)

CMML representation of formula $x!! - y^2 = 0$

```
<apply><eq/><apply><minus/>
  <apply><csymbol cd="latexml">double-factorial</csymbol>
    <ci>x</ci></apply>
  <apply><csymbol cd="ambiguous">superscript</csymbol>
    <ci>y</ci><cn type="integer">2</cn></apply>
</apply><cn type="integer">0</cn></apply>
```

Content MathML is no longer purely syntactic. It is independent on the presentation aspects of the formula and encodes semantically equivalent formulae the same.

Syntax layout tree (SLT)

From PMML, we extract a syntax layout tree (SLT) typed representation [10]:

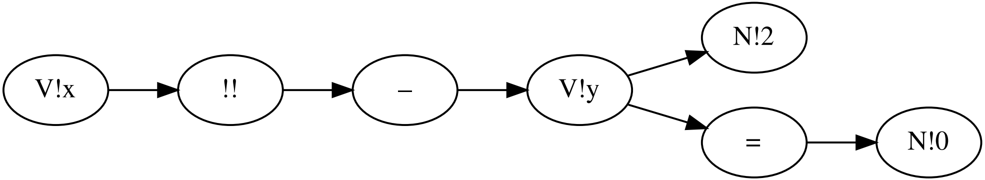


Figure: SLT representation of formula $x!! - y^2 = 0$

In our system, we tokenize the SLT into paths in depth-first-search order:

Tokenized SLT representation of formula $x!! - y^2 = 0$

```
V!x !! n -, V!x - nn -, !! - n n, !! V!y nn n, - V!y n nn,
- N!2 na nn, - = nn nn, V!y N!2 a nnn, N!2 0! n nnna,
V!y = n nnn, V!y N!0 nn nnn, = N!0 n nnnn, N!0 0! n nnnnn
```

Operator tree (OPT)

From CMML, we extract an operator tree (OPT) typed representation [10]:

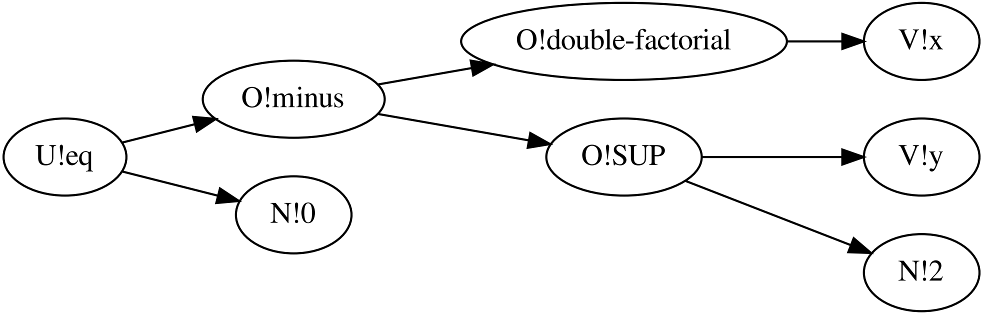


Figure: OPT representation of formula $x!! - y^2 = 0$

Operator tree (OPT)

Tokenization into paths

In our system, we tokenize the OPT into paths in depth-first-search order:

Tokenized OPT representation of formula $x!! - y^2 = 0$

```
U!eq O!minus 0 -, U!eq O!SUP 01 -,
U!eq O!double-factorial 00 -, O!minus O!double-factorial 0 0,
O!minus V!x 00 0, !double-factorial V!x 0 00, V!x 0! 0 000,
O!minus O!SUP 1 0, O!minus N!2 11 0, O!minus V!y 10 0,
O!SUP V!y 0 01, V!y 0! 0 010, O!SUP N!2 1 01, N!2 0! 0 011,
U!eq N!0 0 -, N!0 0! 0 0
```

Operator tree (OPT)

Tokenization into prefix (Polish) and infix notation

In our system, we also tokenize the OPT into visited nodes in depth-first-search order (also known as topological sort, prefix notation, or Polish notation):

Tokenized OPT representation of formula $x!! - y^2 = 0$

```
U!eq O!minus O!double-factorial V!x O!SUP V!y N!2 N!0
```

By parenthesizing and recognizing infix operators, we also tokenize into infix notation:

Tokenized OPT representation of formula $x!! - y^2 = 0$

```
((O!double-factorial(V!x) O!minus O!SUP(V!y, N!2)) U!eq N!0)
```

The infix notation is the closest to the LaTeX representation. When used in a search engine with a tf-idf vector space model (VSM) representation of mathematical formulae, prefix and infix notations are equivalent.

Tf-idf vector space model (VSM)

In our system, we use a tf-idf soft vector space model (VSM) for the representation of both text and mathematical formulae. We use the following representations of mathematical formulae:

- SLT tokenized into paths in depth-first-search order,
- OPT tokenized into paths in depth-first-search order, and
- OPT tokenized into nodes in prefix notation.

Tokens from several representations of mathematical formulae can be combined, and previous work shows the usefulness of this approach [10]. The best combination of representations will be selected by parameter optimization.

Soft cosine measure (SCM)

Whereas the VSM and the cosine similarity measure only take matching tokens into account when retrieving documents, we use the soft VSM with the soft cosine measure (SCM) [11, 3, 4, 12, 13, 14, 15], which also take the similarity of tokens into account, solving the issue of synonymy. We use the following sources of similarity between textual and mathematical tokens:

- a FastText model [16] trained jointly on text and mathematical formulae, and
- a linear combination of token similarities produced by two FastText models, one trained on text and the other trained on mathematical formulae.

The best source of token similarity and FastText parameters will be selected by parameter optimization.

Conclusion

The **MIR-MU** research group will compete in the **ARQMath** competition using four math-aware search engines. In this talk, we discussed system design and the representations of mathematical formulae used by the search engines. Hopefully, we will make **Math StackExchange** sing and dance. Fingers crossed!

Source: formula1.com

Bibliography I

[1] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. URL: `https://arxiv.org/abs/1810.04805` (visited on 04/09/2020).

[2] Mike Schuster and Kaisuke Nakajima. "Japanese and korean voice search". In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2012, pp. 5149–5152. URL: `https://research.google/pubs/pub37842.pdf` (visited on 04/09/2019).

[3] Delphine Charlet and Geraldine Damnati. "Simbow at SemEval-2017 Task 3. Soft-Cosine Semantic Similarity Between Questions for Community Question Answering". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. 2017, pp. 315–319.

Bibliography II

[4] Vít Novotný. “Implementation notes for the soft cosine measure”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 1639–1642. DOI: 10.1145/3269206.3269317.

[5] Quoc Le and Tomas Mikolov. “Distributed representations of sentences and documents”. In: *International conference on machine learning*. 2014, pp. 1188–1196.

[6] Petr Sojka, Michal Růžička, and Vít Novotný. “MIaS: Math-Aware Retrieval in Digital Mathematical Libraries”. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 2018, pp. 1923–1926. DOI: 10.1145/3269206.3269233.

[7] Heinrich Stamerjohanns et al. “Transforming large collections of scientific publications to XML”. In: *Mathematics in Computer Science* 3.3 (2010), pp. 299–307.

Bibliography III

[8] David Formánek et al. "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, July 9, 2012–July 13, 2012). Ed. by James Davenport et al. CEUR Workshop Proceedings 921. Aachen, 2012, pp. 91–103. URL: `http://ceur-ws.org/Vol-921/wip-05.pdf` (visited on 04/08/2020).

[9] Ron Ausbrooks et al. *Mathematical markup language (MathML) version 3.0 2nd edition*. Ed. by David Carlisle, Patrick Ion, and Robert Miner. 2014. URL: `https://www.w3.org/TR/MathML3/Overview.html` (visited on 01/16/2020).

Bibliography IV

[10] Behrooz Mansouri et al. "Tangent-CFT: An Embedding Model for Mathematical Formulas". In: *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*. 2019, pp. 11–18.

[11] Grigori Sidorov et al. "Soft similarity and soft cosine measure. Similarity of features in vector space model". In: *Computación y Sistemas* 18.3 (2014), pp. 491–504.

[12] Vít Novotný et al. *Text classification with word embedding regularization and soft similarity measure*. 2020. URL: `https://arxiv.org/abs/2003.05019` (visited on 04/08/2020).

[13] Vít Novotný. *Implement Soft Cosine Measure*. Jan. 6, 2018. URL: `https://github.com/RaRe-Technologies/gensim/pull/1827`.

Bibliography V

[14] Vít Novotný. *Implement Levenshtein term similarity matrix and fast SCM between corpora*. Apr. 5, 2018. URL: https://github.com/RaRe-Technologies/gensim/pull/2016.

[15] Vít Novotný. *Reduce memory use of term similarity matrix constructor and deprecate the positive_definite parameter*. Apr. 4, 2020. URL: https://github.com/RaRe-Technologies/gensim/pull/2783.

[16] Piotr Bojanowski et al. "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 135–146.

MUNI

FACULTY

OF INFORMATICS