LaSer : A search engine for mathematical formulae

AGNIVO SAHA, ANURAG ANAND, BISWAJIT PARIA,
HARITABH SINGH, PRITHWISH MUKHERJEE, SANDESH C and
MAYANK SINGH, Department of Computer Science, IIT Kharagpur

We present a search engine for mathematical formulae. The MathWebSearch system harvests the web for content representations (currently MathML) of formulae and indexes them using unigrams, bigrams and trigrams. The query for now is limited to Latex format only and can and will be extended to free form queries. Our Project can be found at https://github.com/biswajitsc/LaSer

Additional Key Words and Phrases: Mathematical Search Engine

ACM Reference Format:
Agnivo Saha, Anurag Anand, Biswajit Paria, Haritabh Singh, Prithwish Mukherjee, Sandesh C and Mayank Singh, 2015. LaSer : A search engine for mathematical formulae. *ACM* V, N, Article A (January YYYY), 3 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

As the world of information technology grows, being able to quickly search data of interest becomes one of the most important tasks in any kind of environment, be it academic or not. This paper addresses the problem of searching mathematical formulae from a semantic point of view, i.e. to search for mathematical formulae not via their presentation but their structure and meaning.

2. RELATED WORK

We have seen that early work has been done in this fields([Oviedo Urazmetov et al. 2014], [Kohlhase and Sucan 2006]), allowing users to search for mathematical equations. There are multiple sites available such as http://latexsearch.com/, http://uniquation.com/en/, http://www.searchonmath.com/ that provide equation searching features. Most of the work done involves converting the equations available into xmls, sustitution trees and indexing via various open source software. On other hand we are extracting various features other than simple xmls of the equations that help in increasing the efficiency of the system, as described in the following sections.

3. OUTLINE OF WORKFLOW

3.1. Convert Latex Formula to xml

The conversion from Latex formulae to xml has been performed using MathML
e.g : $\lambda = 1$ is coverted to the following xml
<?xml version="1.0" encoding="UTF-8"?>
<m:math xmlns:m="http://www.w3.org/1998/Math/MathML" display="block">
<m:mrow>

Author's addresses: Agnivo Saha, Anurag Anand, Biswajit Paria, Haritabh Singh, Prithwish Mukherjee, Sandesh C and Mayank Singh, Computer Science Department, Indian Institute of Technology, Kharagpur.

<m:mi> λ </m:mi>
<m:mo>=</m:mo>
<m:mn>1</m:mn>
</m:mrow>
</m:math>

3.2. Simplifying using equations and generating additional xmls.

The simpification has been done using python library sympy. The MathML's are converted into expressions and sympy's simplify routine is called and the corresponding MathML is generated. We work on both the original MathML and the simplified MathML. e.g : ((a + b) + c) is simplified to a + b + c and then MathML is generated as above. If the operands are constants, then the equation is simplified by evaluating it. e.g : 2 + 3.5 is replaced by 5.5 .

3.3. Number Normalization and Unicode Normalization.

3.3.1. Number Normalization. Firstly, we are taking the MathML and taking the decimal numbers obtained inside $< mn >< /mn >$ tags. We are adding all the numbers obtained by reducing precision. For eg. - Original MathML :-
$< mathxmlns = "http : //www.w3.org/1998/Math/MathML" display = "block" >$
$< mrow >$
$< mi > x < /mi >$
$< mo > + < /mo >$
$< mn > 2.90646 < /mn >$
$< /mrow >$
$< /math >$

Converted MathML :-
$< mathxmlns = "http : //www.w3.org/1998/Math/MathML" display = "block" >$
$< mrow >$
$< mi > x < /mi >$
$< mo > + < /mo >$
$< mn > 2.90646 < /mn >< mn > 3.0 < /mn >< mn > 2.9 < /mn >< mn > 2.91 < /mn >< mn > 2.906 < /mn >< mn > 2.9065 < /mn >$
$< /mrow >$
$< /math >$

3.3.2. Unicode Normalization. Here we are normalizing the unicode characters to increase the match. We use NFKD: Normalization Form Compatibility Decomposition of the unicodes. Each unicode is converted into its NFKD form using python library unicodedata. So, if the base form of two unicode characters are same, they can still be matched.

3.4. Unigram + Bigram + Trigram Feature Extraction

Here the xml is treated simply as string and the unigram, bigram and trigram features are extracted. The corresponding postings list (inverted indexing of the equations) are also computed.
Examples (The format is {feature : [(eqn_id, term_frequency)]} :-

3.5. TF-IDF weights calculation

Each equation is converted into a vector representation using the extracted unigram, bigram and trigram features
The $tf_{weight}(eqn, feature) = 1 + log_{10}tf$, $idf_{weight}(feature) = 1 + log_{10}(N/df)$, here $tf_{weight}(eqn, feature)$ stands for the term frequency or the number of times the "feature" occurs in "eqn" and $idf_{weight}(feature)$ stands for how rarely/frquently the term occurs in the equations.

3.6. Query Normalization

Latex query given as input is also normalised as mentioned above to reduce unicode variants. The resulting equation is also simplified. Finally the query is converted into a high dimensional vector space using the extracted features mentioned above.

3.7. Cosine Similarity

Currently cosine similarity is being used as the metric to determine the similarity between query vector and the indexed vectors.

3.8. Backend and frontend

The Backend is running on a python server and webpage is used as the frontend

4. CONCLUSIONS AND FUTURE WORK

— Support free form queries as input.
— Retrieve on basis of structural similarity, for example: $x + y = 1$ and $a + b$ and $(2 * X - 1) + (2 * Y)$ are all structurally similar.
— Retrieve on simplification for example $x * x = 1$ and $x^2 = 1$ are same after simplification.
— Support search by names of popular equations.

ACKNOWLEDGMENTS

REFERENCES

Michael Kohlhase and Ioan Sucan. 2006. A Search Engine for Mathematical Formulae. In *Proceedings of the 8th International Conference on Artificial Intelligence and Symbolic Computation (AISC'06)*. Springer-Verlag, Berlin, Heidelberg, 241–253. DOI:http://dx.doi.org/10.1007/11856290_21

Arthur Alejandro Oviedo Urazmetov, Nikolaos Kasioumis, and Karl Aberer. 2014. *$5e^{x+y}$: A Math Aware Search Engine for CDS*. Ph.D. Dissertation. Ecole Polytechnique, Lausanne. http://cds.cern.ch/record/1670010 Presented 09 Apr 2014.