$5e^{x+y}$: Searching over mathematical content in digital libraries

Arthur Oviedo
EPFL, Switzerland.
arthur.oviedo@alumini.epfl.ch

Nikos Kasioumis
CERN, Switzerland.
nikos.kasioumis@cern.ch

Karl Aberer
EPFL, Switzerland.
karl.aberer@epfl.ch

ABSTRACT

This paper presents $5e^{x+y}$, a system that is able to extract, index and query mathematical content expressed as mathematical equations, complementing the CERN Document Server (CDS)[3]. We present the most important aspects of its design, our approach to model the relevant features of the mathematical content, and provide a demonstration of its searching capabilities.

1. INTRODUCTION

CDS (CERN Document Server) is the institutional digital repository developed and used at CERN[1]. It contains more than 1,000,000 records and stores more than 400,000 full text documents, organized in different collections among published articles, books and preprints. Invenio[4] is the open-source digital library software platform behind CDS and it is developed in parallel to CDS at CERN. Besides CDS, Invenio supports around thirty scientific institutions worldwide including INSPIRE (a collaboration between Fermilab, CERN, DESY and SLAC) EPFL and ILO. A considerable amount of the full text documents on CDS contain mathematical content, represented in the form of equations. The Preprints collection alone contains more than 700,000 records, harvesting documents from services like ArXiV[2] where most of the documents are in the areas of physics, mathematics and statistics which are very rich in this type of content. Even though Invenio provides a powerful search engine that allows users to query records using different fields like author or keywords, it is not suitable for mathematical content. $5e^{x+y}$ is developed as a first attempt to address this limitation by allowing querying for records based on mathematical equations.

2. SYSTEM OVERVIEW

In this section we provide a general overview of the main design points of $5e^{x+y}$. A more detailed description of the system can be found in [10].

Joint Conference on Digital Libraries 2015 Knoxville, USA

2.1 Features Extraction

Our initial focus in the project was to define a good set of features that we could extract and index from our equations. We divide them into two different categories: notational and structural features. The notational features relate to the actual naming and representation of the different elements of the equation. The selected format for representation and internal processing of the equations was MathML. This language allows for easier processing because of its markup nature and has a stronger support from the W3C among other communities. For the notational features we extract the leaf elements of the equation: variables, numbers and operators represented by the <mi>, <mn> and <mo> tags respectively. In addition, we index 2-grams and 3-grams of symbol-operator and symbol-operator-symbol sequences.

We perform some processing of these tokens such as normalizing and decomposing of complex characters. This allows us to partially match characters with different unicode points but similar or equal visual rendering. For instance LATIN CAPITAL LETTER A WITH RING ABOVE with code 0xc5 and ANGSTROM SIGN with code 0x212b both are rendered as Å and can both be used to refer to the Angstrom measurement unit. Following this example, we index both the original token <mi>Å</mi> and the normalized form <mi>Å</mi> <mi>A</mi>.For operators which are semantically related such as comparison, integrals and arithmetic, we apply another set of rules since their unicode characters do not provide an easy way to match them. For example the INTEGRAL character with code 0x222b generates the tokens <mo>∫</mo> and INT_OP and the CONTOUR INTEGRAL character with code 0x222e generates the tokens <mo>∮ </mo> and INT_OP. In this way, both have a partial matching in the second token.

The second set of features relate to the structure of the equation. Two different properties of mathematical equations impose additional challenges for a Mathematics Information Retrieval (MIR) system. First, variables can be used indistinguishably and any character set can be used to represent the same equation. For example, $f(t) = vt + c$ and $g(x) = ac + b$ can represent the same linear equation, however the context in which each variable is used follows certain conventions. To address this issue, we relied on pattern matching, a very powerful feature that current Computer Algebra Systems (CAS) provide. We integrated Mathematica[7] into our system, to perform pattern matching on the equations and identify occurrences within a predefined set of patterns. This allows us to detect if the equation contains certain types of common algebraic structures. For instance,

the pattern `a_.x^ 2 + b_.x + c_.` is used to identify a quadratic expression. This mechanism is very powerful since _.x represents any type of expression, not only a single term.

An additional challenge mathematical equations pose is the fact that a given equality can be rewritten in multiple ways taking into account properties of the different operators like associativity, distributivity and commutativity among others. Our approach relies once more in the features provided by Mathematica: we perform the `Simplify` operation which applies a set of predefined rules on a given equation with the goal of providing a canonical representation.

2.2 System architecture

$5e^{x+y}$ is implemented on top of the Lucene/Solr [6] framework. We expanded the default schema with several fields, the most important ones being: the `math_notational_field` which stores all the described tokens and n-grams, the `math_structural_field` which stores the occurrences of predefined algebraic structures and the `filename` which is used to store the file/record where the particular equation was found.

In order to extend Lucene with the new use case, our work extends some of the base classes. `MultiplePatternTokenizer` is in charge of producing a stream of tokens from a single equation. These tokens are identified by applying multiple regular expressions. `UnicodeNormalizingFilter` applies the unicode normalizing transformation described above. `SynonymExpandingFilter` expands operators, using a manually computed table based on [8], with the category they belong to. Finally `StructuralPatternsTokenizer` provides the pattern matching functionality by establishing communication with Mathematica's `Kernel Link` API.

To integrate $5e^{x+y}$ with Invenio, we used the `solrpy` library which communicates with an instance of Solr running $5e^{x+y}$ as a plugin. Since Solr provides different integrating mechanisms with external systems, $5e^{x+y}$ can be easily be incorporated into different digital libraries.

3. DEMONSTRATION

As part of the demonstration of $5e^{x+y}$, we present a demo instance of the Invenio software with the mathematical equations search functionality. The user can access the mathematical search functionality through a link in the main page. Once there, the user is presented with a classic search functionality where they can input an equation. The equation can be expressed in either LaTeXor MathML. To ease the writing of LaTeXcode, we include an online editor[5] that renders the input as the user types it. LaTeXcode is usually more familiar to users, however, the internal representation is in MathML as previously explained. The translation from LaTeXto MathML is done through the third party library SnuggleTex[9] that has some shortcomings, so we allow users to input the query in MathML as well. When the query is ready, the user can click the submit button and the system then performs the search. Once the system has performed the lookup and produced the results, they are presented to the user. The system displays the 20 results with the highest score. Each result shows the matched equation snippet, the title of the record containing the given equation and the link to the record.

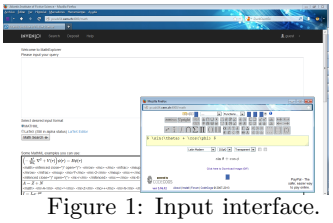

Figure 1: Input interface.

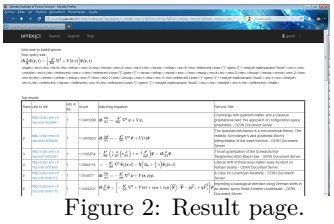

Figure 2: Result page.

4. CONCLUSIONS

In this demonstration, we presented an important limitation of CDS, and how our system, $5e^{x+y}$, provides efficient and precise searching capabilities over mathematical to our system. We presented our approach to model mathematical equations in terms of two different types of features. How we extended the Lucene/Solr Framework to handle our specific type of content and how through the Mathematica CAS, we were able to include powerful functionalities such as pattern matching and equation simplificiation to our system, and use them to improve the quality of the results. Finally we presented our integration mechanism with the Invenio software through the solarpy library.

5. AVAILABILITY

Both the Invenio software and the $5e^{x+y}$ module are open-source software available through GPL Licence. Invenio can be downloaded from `http://invenio-software.org/`. $5e^{x+y}$ can be downloaded from `https://github.com/arthoviedo/cern_math_explorer`.

6. REFERENCES

[1] About cern, 2014. `http://home.web.cern.ch/about`.

[2] arxiv.org e-print archive, 2014. `http://arxiv.org`.

[3] Cern document server, 2014. `http://cds.cern.ch/`.

[4] Invenio, 2014. `http://invenio-software.org/`.

[5] Online latex editor - create, integrate and download, 2014. `http://www.codecogs.com/latex/eqneditor.php`.

[6] Solr is the popular, blazing-fast, open source enterprise search platform built on apache lucene™., 2014. `http://lucene.apache.org/solr//`.

[7] Wolfram mathematica: The world's definitive system for modern technical computing, 2014. `http://www.wolfram.com/mathematica//`.

[8] X. Lee. Unicode: Math symbols, 2010. `http://xahlee.info/comp/unicode_math_operators.html`.

[9] T. U. of Edinburgh School of Physics and Astronomy. Snuggletex (1.2.2). `http://www2.ph.ed.ac.uk/snuggletex/documentation/overview-and-features.html`.

[10] A. Oviedo, N. Kaisoumis, and K. Aberer. $5e^{x+y}$: A math-aware search engine for cds. CERN Document Sever.