MathWebSearch 0.5: Scaling an Open Formula Search Engine

Michael Kohlhase, Bogdan A. Matican, and Corneliu-Claudiu Prodescu

Computer Science, Jacobs University Bremen

http://kwarc.info

Abstract. MathWebSearch is an open-source, open-format, content-oriented search engine for mathematical formulae. It is a complete system capable of crawling, indexing, and querying expressions based on their functional structure (operator tree) rather than their presentation.

In version 0.5, we concentrate on scalability issues in MathWebSearch to take advantage of corpora in the giga-formula range. We re-implemented the index to make it distributable and made all the APIs web standards conformant. Our experiments show that this architecture results in a scalable application.

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. Here we tackle the problem of finding information that is given in the form of (mathematical) formulae. Standard search engines like Google cannot deal with formulae at all, severely limiting the reach and utilization of technical, scientific, and engineering documents.

In this paper we present new work in the context of the MathWebSearch system; a search engine that addresses the problem of searching mathematical formulae from a semantic point of view, it finds formulae by their structure and meaning not via their presentation.

In [KS06] we have presented the motivation, query language, and web front end of MathWebSearch 0.1. In [KK07] we have re-examined the value proposition of semantic search for mathematical knowledge homing in on the benefits and sacrifices induced by the various search approaches [You06b, MM06, LM06], from a user’s perspective. The result of this analysis is MathWebSearch 0.5, which we describe in this paper. The new version features significant efficiency gains (space efficiency increased by a factor of five), new management features, advanced searching capabilities, and a new user interface. The MathWebSearch system (see [MWS] for details) is released under the Gnu General Public License.

The motivation for the work reported in this paper is the availability of large corpora, such as the arXMLiv corpus [SK08] with almost three quarters of a million scientific articles and an estimated giga-formula. This has not only re-kindled

1 We deem a corpus as large if it has more than 20 million expressions
interest in formula search, but also severely taxes the scalability of systems. Scalability issues for presentation-based search engines have been addressed in [SL11]. Such engines map formulae to “special words” which can then be indexed by conventional bag-of-word information retrieval engines, which have become extremely scalable over the last years. The case for MathWebSearch is completely different, since the content-based unification queries it offers require an index data structure that reflects the inner structure of formulae (rather than just pointers to words). Even with the space efficiency gains in MathWebSearch 0.5, the indices will surpass the main memory of most machines. Therefore, we have laid the foundations for distributing the MathWebSearch in this version.

Before we present MathWebSearch 0.5 from a technical perspective in Section 3, we will recap unification-based querying. We evaluate the system on a large corpus in Section 4 and see that we need to distribute MathWebSearch to cope with linear RAM usage. Section 5 presents the necessary extensions of the indexing. Section 6 concludes the paper and discusses future work.

2 Querying Mathematics by Unification

Retrieval of mathematical knowledge and information via unification-based queries for content-encoded mathematical formulae is very natural. In [KS06] we have already discussed instantiation queries, which can be used to retrieve partially remembered formulae, e.g. the query for the formula for energy of a given signal \( s(t) \) in Figure 1. Note that instantiation queries are more expressive as a query language than e.g. regular expressions supported by some text-based search engines, since we can use variable co-occurrences to query for co-occurring subterms.

<table>
<thead>
<tr>
<th>Query (query variables marked as named boxes)</th>
<th>Result (Parseval’s Theorem)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ \int_{\min}^{\max} f(x)^2 dx ]</td>
<td>[ \frac{1}{T} \int_0^T s^2(t)dt = \sum_{k=-\infty}^{\infty}</td>
</tr>
</tbody>
</table>

Fig. 1. An Instantiation Query

To see the full power of unification-based querying consider a student who encounters \( \int_{\mathbb{R}^2} |\sin(t) \cos(t)| dt \) and wishes to know if there are any mathematical statements (like theorems, identities, inequalities) that can be applied to it. Indeed, there are many such statements (for example Hölder’s inequality) and they can be found using generalization queries. The idea behind answering generalization queries is that the index marks universal variables in subterms.

\[ \text{The next NTCIR-10 Challenge in spring 2013 will have a “math track”. NTCIR evaluates information access technologies in a series of competition events in Japan.} \]

\[ \text{We consider an identifier as universal if it can be instantiated without changing the truth value of the containing expression. In formal representations like first-order logic, such variable occurrences can be effectively computed, but in semi-formal settings like mathematical textbooks, they have to be approximated by heuristic methods; see the discussion in the conclusion for details.} \]