Units for MathWebSearch[*] EdN:1

Guided Research Proposal

Tom Wiesing
Supervisor: Michael Kohlhase
Jacobs University, Bremen, Germany

January 3, 2015

Abstract

In this proposal we describe an approach to introduce Units to MathWebSearch [2] EdN:2

1 Introduction

[3] EdN:3

MathWebSearch (MWS for short[4]) is a system to search (latex) documents EdN:4
for mathematical formulae. Additionally it can also search for text
in the documents[5]. However it not only searches for formulae in a simple- EdN:5
minded string way but also includes simple transformation rules, such as
$a + b = b + a$. Additionally, is is possible to search with wildcards such as
$x + \sqrt{x}$. In this example MWS delivers results of the given form where x is
substituted with any sub-formular[6]. EdN:6

MWS has been shown to be very useful for mathematicians[7]. The transformation EdN:7
system it uses can currently be used only for mathematical formulae
which limits its applications. In this paper we propose an extension for
physical units[8]. Instead of transforming mathematical formulae, the search EdN:8

[*] EdNote: Preliminary Title

[2] EdNote: Write abstract properly

[3] EdNote: Write an introductary sentence / paragraph?

[4] EdNote: should I really use abbreviations here?

[5] EdNote: Or is this the multi-faceted search that is currently planned? Do we really need this sentence?

[6] EdNote: Re-formulate this and link to an example

[7] EdNote: Quote neeeded

[8] EdNote: Reformulate this?

engine should transfer physical units. The end-user will search, for example, 100 °C and also get results which show 212°F or 373.15K.

This proposal is organised as follows[9]: In section 2 we shortly describe EdN:9
and discuss the existing MathWebSearch system and then proceed in section 3 to describe in detail the proposed extension. Finally in section 4 we discuss possible problems with this approach and related work.

2 Background - The existing MathWebSearch system

As mentioned above, MathWebSearch is a search engine for mathmatical
formulars in documents. It has a corpus of ??? [10] documents and is currently EdN:10
deployed and used by ??? and ??? [11]. EdN:11

The frontend, running client-side in a web browser, is written in HTML5, CSS and JavaScript. It accesses a REST backend and dependens on MathML support to render Mathematics. It simply accessing a REST backend via AJAX. When the client enters a LaTeX forumar to search for, the backend renders MathML which is then sent back to the client. Next, the client renders the MathML (to show the formular the user is searching for) and also sends back the MathML to the server to search for it. Finally the server sends back results to the client which then shows a list of them.

The backend, written in ??? [12], has 2 independent components. The first EdN:12
component, LaTeX to MathML translation, is not part of MathWebSearch
directly. It is rather uses LaTeXML to work[13]. The searching however is EdN:13
handled by MWS directly.

For each corpus a big search index is generated before search [14]. Then EdN:14
during runtime only this index is searched. [15] EdN:15

[16] EdN:16

Advantages of this approach / how MWS is received [17] EdN:17

Disadvantages of the current approach

- has to be re-generated each time a document is added

[9] EdNote: Update this possible if we change the structure

[10] EdNote: Get an estimate here

[11] EdNote: where?

[12] EdNote: What is the backend written in?

[13] EdNote: Really? Reference needed.

[14] EdNote: Explain this better

[15] EdNote: Explain theory graphs here

[16] EdNote: Explain how MWS uses those here

[17] EdNote: Find points here

3 The proposed extension

The goal of the guided research is to get a complete system that searches a corpus of documents for units as already indicated above. This system should be nicely accessible to the end user. Furthermore, it should be extendable with respect to

- *the unit system,* Adding new units should be easy and most translations should be deducted by the system automatically.
- *and the searchable corpus.* It should be easy to search a different corpus of documents provided units are properly marked up inside it.

As with the existing system, the frontend should be a web page that works in all modern browsers. It should be mobile friendly. It should communicate directly with the backend via a RESTful API. The frontend should incoperate 4 main elements:

1. a search input for a (real) quantity,
2. a search inout for a unit, described further in section 4,
3. an additional search input for text and
4. a result page that displays results and links to the found documents.

The REST-based backend should have 3 major tasks:

1. translate raw unit input into a standardised form (see section 4 for details),
2. search for documents using the input from the frontend and
3. make the documents available to the enduser.

The corpus

- should consist of a lot of tex documents
- should have marked up units
- ideally, if a single document is added, only the new corpus should have to be re-scanned (procedular generation)
- should be easily exchangable

The unit transition system

- should be a graph
- should have few connected components and each of the components should be sparse (i. e. few connections)
- translation are:
  - either a factor towards a single unit
  - or a composition of a factor together with a product or fraction of units [18] EdN:18
  - Perhaps include prefixes somehow?

4 Problems and related Work

There are several problems with this approach. [19] EdN:19

The units have to be entered in some fashion in the search engine. While it is trivial to design an interface where a single unit can be entered, it is non-trivial when we want to recognise composite units as well. Furthermore si-prefixes (such as kilo or mini) should also be recognised which is an additional problem. There are a few alternatives to use as input formats: AsciiMath, LaTeX and MathML, to name only 3. Independent of the input format, the end result (delivered to the backend of the search engine) should be MathML containing the unit in some yet-to-be-determined standard form.

The unit equivalences should be in the form of a theory graph [20] The EdN:20 equivalence (translation) itself contains both a translation for the quantity and the unit. The values searched for should be simple real numbers together with the forumlar. However since all numbers must be represented in a finite fashion, translations formulars can gvie problems. A natural quantity in one unit can be a repeating decimal in a another unit. Also rounding of quantities might depend on the unit used. It is thus preferable not not search for an exact value but instead for a range of values. These ranges should depend on the unit. These ranges have just been implemented in MWS by Radu [21]. [22] [23] EdN:21 EdN:22 EdN:23

[18] EdNote: Figure out more details about this

[19] EdNote: Continue intro pargraph

[20] EdNote: Reference this properly.

[21] EdNote: Quote needed; reformulate

[22] EdNote: Write excatly how to use ranges

[23] EdNote: Check if the reference in (2) is clear

Finally, there are the problems of finding a corpus and marking up units in this corpus. The second problem, marking up units in a corpus, will be taken on by ???[24] in a seperate thesis. The first problem will be postponed for now. In case there is still time, a suitably corpus can be marked up manually and plugged into the system. For first testing, I will create a dummy corpus of lorem-ipsum style documents which will contain a random sampling of units. EdN:24

[25] EdN:25

[24] EdNote: Who is doing unit finding?

[25] EdNote: Write final paragraph