Semantic Search for Quantity Expressions
Guided Research Proposal

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

January 27, 2015

Abstract

In this proposal we describe how to approach Semantic search for Quantity Expressions, in particular how to make the existing MathWebSearch system aware of physical units. [1] EdN:1

1 Introduction

In this paper we want to give an approach to Semantic Search for Quantity Expressions. A quantity expression is a number[2] together with a physical unit, for example $25\frac{\mathrm{m}}{\mathrm{s}}$ where $\frac{\mathrm{m}}{\mathrm{s}}$ the unit. This quantity expression is equivalent to $90\frac{\mathrm{km}}{\mathrm{h}}$ (with equivalent in this sense meaning it expresses the same quantity) although the units are different. In a semantic search for quantity expressions we want to be able to search for a certain quantity expression and find any equivalent ones as well. EdN:2

MathWebSearch (MWS for short) is an existing semantics-aware system to search LaTeX documents [1] for mathematical formulae [2]. As it is semantics-aware it not only searches for formulae in a simple-minded "text search" way but also includes simple transformation rules, such as $a + b = b + a$. Additionally it can deal with wildcards such as $x + \sqrt{x}$. For this query MWS would deliver forumlae as above where x has been substituted with any sub-formular.

[1] EdNote: Continue and finish abstract
[2] EdNote: Find a better expression for this.
[1] Technically, MWS itself can only search XHTML documents. However with the help of LaTeXml [5] it also handles LaTeX documents.

So far the transformation system has been used by MWS exclusively for mathmatical forumlae. In this paper we propose an extension for quantity expressions which should prove useful for physicists. 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: In section 2 we shortly describe the existing MathWebSearch system and then proceed in section 3 to describe in detail the proposed extension. Finally in section 4 we discuess challenges as well as related work.

2 An overview of the existing MathWebSearch system

As mentioned above, MathWebSearch is a search engine for forumlae. MWS consists of 3 main components as well as a frontend [2] [3].

The backend consists of 3 main components,

1. a crawler,
2. a core system and
3. a public REST API.

The crawler, as its name suggests, crawls corpera for forumlars. For each corpus MWS uses, a seperate crawler has to be implemented. The crawled formulars are then transformed into a format easily accessible by the search engine. Next, the core system builds a search index. The core system is also responsible for parsing queries and sending results back to the REST API.

The frontend, which is not part of MWS itself but running client-side in a web browser, is written in HTML5, CSS and JavaScript. It accesses the REST backend and depends on MathML support to render Mathematics. When the client enters a LaTeX formular to search for, the LaTeXml daemon [1] is used to transform the query in content MathML. Next, the client renders the MathML (to show the formular the user is searching for) and then sends the query off to the MWS API. Upon receiving results, the client renders them and links to the original documents.

There are several implementations of frontends and crawlers as well as extensions of MathWebSearch. One particular implementation is capabale

[2] The frontend is not part of MWS directly but rather built on top of the REST API, more on this later.

of crawling the arXMLiv corpus, which contains approximatly 750.000 documents. A list of demos can be found at [4].

[3] EdN:3

3 The goal - a semantic search for quantity expressions based on MathWebSearch

The goal of the guided research is to get a complete system that can perform a semantic search of quanity expressions on a specific corpus. As such it needs 4 components: (1) *A crawler* that searches through documents and finds units which can be indexed, (2) *a core system* that takes care of storing indexes and processing queries and is aware of units with the help of a theory graph, (3) *a rest API* that has to have a way of sending units from and to a client and (4) *a frontend* that runs in the web browser.

Furthermore the system will be flexible with respect to

- *the quanity expressions it understands,* Adding new units should be a simple process so that the system can be easily extended to also capture rare units and
- *the corpus* which should be exchangable easily.

The crawler will be based on existing MWS crawlers. As it is very difficult to find and mark up all units in a corpus automatically, we will restrict ourself to an artificial corpus containing only lorem ipsum documents with randomly inserted units. This does not solve the problem of writing a crawler, however it will enable us to build a proof-of-concept system first.

The core system will almost completly be inherited from existing implementations. As it does not search formulae, it has to be made aware of unit translations. This will be done with the help of theory graphs. It is not yet clear how exactly these will look, a speculation can be found below in 4.

The rest-based API will have to take the user input from the client and pass the entire quantity expression on to the core system. Once it receives a result, it will have to translate these back into human-readable form and then pass these on to the frontend.

As with the existing system, the frontend will be a web page that incorperates 3 main elements:

1. a search input for a number,

[3] EdNote: how MWS uses Theory graphs

2. a search input for a unit, with the format as described below in 4 and

3. a result page that displays results and links to the found documents.

[4] EdN:4

4 Challenges and related Work

The biggest challenge with the approach will be to find a standardised representation for quantity expressions, mainly for units[3]. Because we want to translate between these representations however, it is insufficent to just have such a representation. We additionally need to find a standardies way of translating between units need to exist as well.

[5]. [6] EdN:5

The units will have to be entered in some fashion in the frontend. While EdN:6
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 units with (si-)prefixes (such as kilo or mega) can either be recognised seperatly or as part of the unit. There are several input formats that can be used: AsciiMath, LaTeX and MathML, to name only a few. Independent of the input format, the end result (delivered to the backend of the search engine) will have to be represented in one way.

Finally, there is the problem of choosing a suitable corpus and implementing a crawler which will find and mark up quantity expressions. The latter will be taken on by ???[7] in a seperate effort[8]. The task of finding EdN:7
a suitable corpus is postponed for now. In case there is sufficient time a EdN:8
corpus can be marked up manually and crawled afterwards.

[9] EdN:9

[4] EdNote: Mayve have a final pargraph in this section? Or write more?

[3] Here, standardised means that all quantity expressions are represented in a standardised, machine-readable fashion, not equivalent quantity expressions being representeted in exactly one way.

[5] EdNote: Describe a possible unit translation system here, have a small example graph with the help of @miancu

[6] EdNote: Mention: Ranges have been implemented in MWS by Radu; Quote needed

[7] EdNote: Who is doing unit finding?

[8] EdNote: Reference here

[9] EdNote: Write final paragraph

References

[1] Deyan Ginev. The LaTeXml daemon: Editable math for the collaborative web.

[2] Radu Hambasan, Michael Kohlhase, and Corneliu Prodescu. MathWebSearch at NTCIR-11. In Noriko Kando and Hideo Joho andKazuaki Kishida, editors, *NTCIR 11 Conference*, pages 114–119, Tokyo, Japan, 2014. NII, Tokyo.

[3] Michael Kohlhase and Corneliu Prodescu. Mathwebsearch manual. Web manual, Jacobs University.

[4] Math WebSearch a semantic search engine. web page at http://search.mathweb.org. seen September 2008.

[5] Bruce Miller. LaTeXML: A LaTeX to XML converter.