Development of an Open-Source Extension to support the **management of metadata** within **Semantic MediaWiki**

META MAN

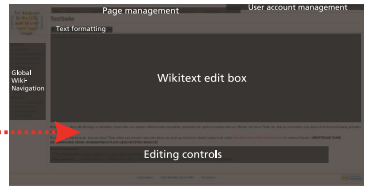
BY TIMO TAGLIEBER

VS.

THE MACHINE-UNREADABLE PILE OF WIKI PAGES

1 The default MediaWiki Layout

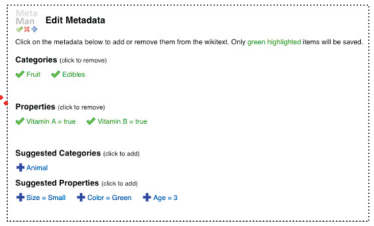
The MediaWiki user interface is focused on version control of the wiki's content and navigation. Metadata like **categories** and **attributes** can only be applied in code by trained users. This lack of a WYSIWYG input mode is repellent to **less technology-affine users**, who are therefore excluded from participation.



Huge wikis like Wikipedia have a whole community, including many power users, taking care of metadata management. Wikis for smaller groups (corporate wikis, organisations) need to manage that effort in a different way. Especially in semantic wikis, users need support in fighting **inconsistency** and **redundancy** of facts.

2 The improved editing interface

The MetaMan extension adds a **new UI element** to the edit view of wiki pages. It shows the page's **current metadata** and **suggests potentially relevant metadata** to add. Through a intuitive visual representation, the user gets a quick overview of the page's categories and semantic attributes, and can easily **remove items or add new ones**.



3 How to hook into the inner workings of MediaWiki

The MediaWiki PHP codebase is extensible in various ways, and many people have done so. MetaMan and Semantic MediaWiki are just two of over **1200 extensions** currently registered. MediaWiki provides "hooks" (**code insertion points**), for altering its behavior. MetaMan uses a hook called ***EditPage::showEditForm:initial***, which translates to "***If the user is about to see the edit view of a page, first run the following extension code***".
The UI can then be modified by injecting custom elements in the HTML output.

4 Where do the metadata suggestions come from?

The suggestions are made based on the assumption, that **similar pages need similar metadata**. So if one page is similar to a set of other pages in the wiki, those might have categories and semantic attributes which could also be relevant for the current page.

Example: A user editing an article about "Albert Einstein" (without any metadata set), might benefit from suggesting the category "Physicist". "Physicist" was detected as relevant, because MetaMan's ranking algorithm found it in similar pages.

MetaMan uses the **cosine similarity measure** to create a **similarity matrix** of all wiki pages.

This measure favours pages approximately dealing with the same topic as being similar. Other information retrieval techniques involved are filtering of stopwords and term vectorization.

In simple words: Pages are more similar when they have a lot of (semantically meaningful) words in common.

MetaMan communicates with the **database backend** of the Wiki engine to retrieve the metadata suggestions. This happens in the **background via Javascript and AJAX**, so the page load is not blocked by time-consuming computations.

5 Extracting helpful metadata

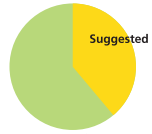
In order to rank the relevance of a metadata ***m*** to suggest, the following formula is used. Note: ***T*** is the set of pages similar to the page ***s***. ***freq*** counts the occurrences of ***m*** in a page. ***sim*** computes the similarity of two pages.

$$relevance(m) = \sum_{i=1}^{|T|} \sum_{j=1}^{freq(m,t_i)} sim(s,t_i)$$

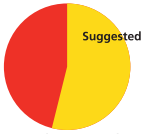
Basically, this is an iteration over all metadata found in all similar pages.
The **more often a metadate occurs**, and the higher its **origin page's similarity** is, the more relevant it is considered. MetaMan suggests the 10 most relevant items to the user.

6 Does it really work?

Obviously, MetaMan can only suggest metadata which already exists somewhere in the wiki. Therefore, the quality of suggestions depends on (at least) partially annotated wiki contents. However, a small evaluation* generated promising results:



Total number of categories



Non-unique categories

In the experiment, MetaMan suggested over 50% of the correct categories** in the test set. In a real situation, where pages might need unique categories (that only apply to one page in the wiki), MetaMan would still suggest of 39% of all correct categories.

* 30 Wikipedia articles, selected from the domain "programming and software".
** Categories set by Wikipedia users.

7 So where's the catch?

Creating and maintaining a similarity matrix is a computation with **quadratic complexity**. Ranking the metadata suggestions is a quicker, ***O(n)*** operation, but can get very **memory-consuming** with a growing number of pages in the wiki.

Tests with several hundres pages showed acceptable performance (<10 seconds) so far. In any way, MetaMan will be rendered inoperabe in huge wikis.

http://timotaglieber.de/metaman