A Real Live Web Service using Semantic Web Technologies: Automatic Generation of Meta-information

Sebastian Brandt, Damir Ćavar[1] and Uta Störl

Dresdner Bank AG
Software Technology and Architecture for Allianz Group Germany
IT-Research
D-60301 Frankfurt am Main
Contact via Uta.Stoerl@Dresdner-Bank.com

Abstract. Meta information in documents is very useful for the management of documents and for information retrieval. Assigning meta information, e.g. topic or keywords to documents helps in identification of relevant documents in search and retrieval processes. However, assigning keywords to documents is a problematic task for human editors. In this paper we describe the project AME^{WS}, a generic Web Service based solution for seamless automatic assignment of topic and keywords to documents in different formats. Furthermore we present a first realization integrating the automatically generated meta information into Word documents themselves and generating additional XML-documents using the Semantic Web standards RDF and Dublin Core.

1 Introduction

Growing amount of digital textual information is one of the most important challenges in business nowadays. Search processes are very time consuming and the quality of search results is often very bad. One way to improve the quality of search results is the use of meta information that refers to the content, rather than the data. For example, assigning topics or keywords to document content can help for identification of relevant documents in search and retrieval processes. Most editing tools and office systems provide the necessary extensions for meta annotation of documents and document content.

However, (semi-) structuring of documents or assigning keywords and topics on the basis of the content as meta information to them is a very painful task. On the one hand, the attempt to define keywords for a document confronts the author with cognitive effort that consumes too much time and energy. On the other hand, the assigned keywords seem to be sensitive to subjective mood, varying on the situation of the author etc. Often enough, such keywords tend to be too general, thus their usefulness is extremely low. Furthermore, a unified linguistic basis seems to be difficult to establish (e.g. some keywords appear preferably in their plural form others in singular etc.). Beside all these

[1] New Address: Indiana University, East 8th St., Bloomington IN 47408

problems, there are legal restrictions on meta information for internet documents in Germany, probably in other countries as well. Beside the ban on the use of brands that are not mentioned in the content, the use of keywords that are not obviously related to the content is forbidden as well. Basically, the standard seems to be that most documents in intranets or in the internet are unstructured and purely or not at all annotated anyway.

A potential solution might lie in automatic annotation of unstructured documents (e.g. keyword generation, topic detection, language detection). Reliable automatic keyword generation for example might generate more standardized meta tags for documents, i.e. keywords in a unique and standardized linguistic form, e.g. the lemma or lexical base form rather than some morphological variant. Such an automatic meta tagger reduces the time and effort for the author and might even be plugged into standard applications seamlessly.

In the following we shall describe the project AME^{WS} (*Automatic Metatagging Engine – Web Service*) for automatic meta tagging of standard office documents and document corpora with the use of text-mining and standard linguistic components. The system we describe is realized as a Web Service that wraps a linguistic tagger, language recognizer and finite state automata (FSA) based text mining components. It generates RDF-conform data with e.g. Dublin Core tags in RDF-format that are stored together with the corresponding document. Alternatively, the meta information is stored directly in a Microsoft Word or PDF document.

2 Meta Tagging Documents

In the following we describe the basic design principles that underlie our first prototype of AME^{WS}. Further, we give some information about the process of how meta information is extracted and stored.

2.1 Design Principles

One important design principle is to minimize the additional effort for users and administrators. We stick to the principle of "minimally invasive knowledge management" [2], i.e. the technology is seamlessly integrated into standard tools and existing infrastructure, without consequences, on the one hand, for the habits and practice of the users and, on the other hand, without changes of business processes, with no or minimal administrative effort. As an example for this principle the functionality of AME^{WS} is seamlessly integrated in Microsoft Word. After pressing a button in the menu the keyword list and additional meta information about the content of a document is generated by AME^{WS} and stored in the properties fields automatically and seamlessly.

Other design criteria that are important for the AME^{WS} prototype are scalability as well as platform and language independence, since we intend to provide the functionality to a big number of employees with different client platforms. Hence we decided to realize AME as a Web Service with the possibility to use both, Microsoft's .NET and Java frameworks (e.g. Eclipse) on the client side. Detailed information about the architecture is given in chapter 3.

Further, AME^{WS} provides a service that can be used together with existing technologies. The automatically generated meta information can be accessed via standard search engines that already exist in the intranets of different related enterprises.

2.2 Analysis of Documents for Meta Tagging

While some meta information is difficult to extract from unstructured documents, there are different properties of documents and text that can be generated automatically. Among others, we assume that with existing technology the following information can reliably be extracted from structured and unstructured text automatically:

- Language of document, paragraph, or sentence
- Topic of document (e.g. business report, …)
- Specific keywords in document
- Specific keywords for one document in relation to a collection of documents
- Named entities (e.g. company, organization, name etc.)
- Term pairs, potentially with labeled relations[2]

There are, of course, many ways and approaches for keyword extraction, clustering, classification and topic detection. Since our focus lies on meta tagging and annotation of documents, we are not primarily interested in the underlying technology of the backend, but rather in the quality of the results. We evaluate statistical as well as purely linguistic technologies in the backend.

At present language recognition, tagging, lemmatization and named entity recognition is realized with the Temis Information Discoverer Extractor provided by the Temis Deutschland GmbH based on finite state automata for German and English. The terms extracted from documents are linguistically annotated (e.g. syntactic category, morphosyntactic features, lemma). In order to determine the relevance of keywords for one document we use different weighting schema, e.g. simple text frequency and inverse document frequency (tf+idf).

2.3 Storing Meta Information about Document Content

Having generated meta information about document content, the gained information has to be stored and managed appropriately. On the one hand, one has to decide on the data format, on the other hand on the storage location itself. With respect to the format, our aim is to use non-proprietary standards to store the documents and meta information and make use of established technology for processing and analysis. We decided to use XML-based Semantic Web Technologies (http://www.w3.org/2001/sw/) with the different coding standards or dialects RDF (http://www.w3.org/RDF/), Dublin Core Metadata Initiative (cf. [4]) and Text Encoding Initiative (http://www.tei-c.org/).

---

[2] We use these term pairs in our K-Net [1] project of the IT Research group at the Dresdner Bank AG. We developed a semantic net generator, basically as an add-on to standard search engines with the capability to provide a view on the term relations found in documents in the intranet.

With respect to the location of the generated information, one has to take into account that business documents are typically available in different proprietary formats, like for example MS Word or PDF, rather than XML. Approaches to use some XML format as a general document format, as realized in Open Office 1.0 or Star Office 6.0, are rather exceptional [5]. The challenge thus is to cope with the different document formats and add the new information to the proprietary document formats. There are numerous approaches to store the meta information. Either one adds it to the different document formats or generates for example a new XML document. The advantages and disadvantages of the different approaches are discussed in [3]. A global conclusion on the preferable variant cannot be given. The decision for one variant depends on the used document types and many other conditions like homogenous or heterogeneous document types or type of document storage. In the first realization described in the next section the generated meta information is stored in the Word documents in proprietary format and additionally in a XML document with RDF and Dublin Core tags.

3 Architecture

In the following we shall describe in more detail the architecture of the AME^{WS} server and first realized clients, as well as some perspectives on further development of the service.

3.1 Server Architecture

AME^{WS} provides different functionality via its Web Service interface, which is for the time being mediated by a Systinet Server (http://www.systinet.com/). The next release will be migrated to the IBM WebSphere platform.

AME^{WS} itself is a container for different analysis components. It contains a Temis Information Discoverer Extractor (IDE) that is a Java-based service, accessible via Java-RMI.

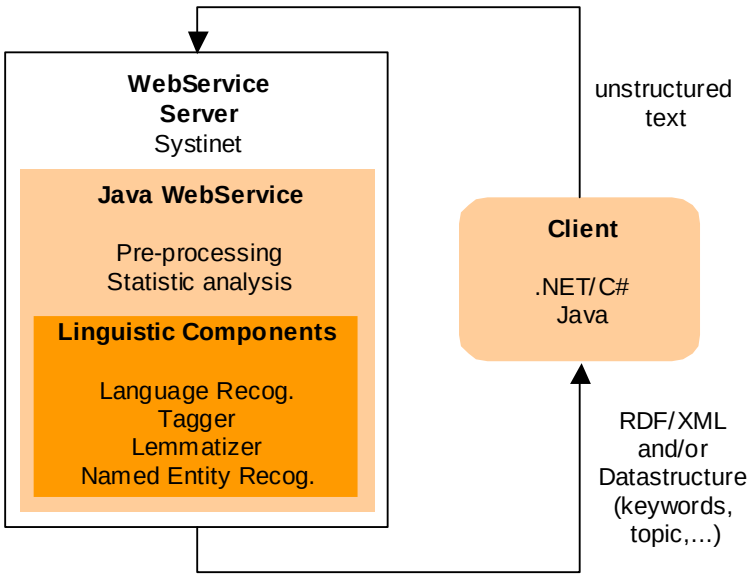
The IDE is responsible for the linguistic analysis of documents, including language recognition, tagging, lemmatization and named entity recognition. The IDE itself as a component of AME^{WS} receives pure text (ASCII) or documents in different formats (DOC, HTML, PDF, PPT, PS etc.) for analysis and returns a concept graph that contains the results of the extraction as specified in different cartridges.

The cartridges provide different extraction services. They are basically finite state automata (FSA) for the extraction of specific terms for specific languages. We use cartridges for term extraction, where basically the important nominal groups in a text are recognized and returned as concept graphs. The concept graphs are processed and statistically analyzed by the wrapping Web Service class of AME^{WS}. The concept graphs contain the full forms of the extracted terms, as well as the lemma (i.e. lexical base form) and categorial information (morphosyntactic and semantic). The semantic information about certain terms means their specification with respect to the set of named entities we use, e.g. person names, organizations, date and time expressions, locations etc.

Morphosyntactic information means linguistic specification of the word itself, e.g. noun, verb, adverb, preposition etc.

Preprocessing of the IDE-input and post-processing of the IDE-output is done in the Web Service component itself. For the generation of keywords we use a simple frequency based weighting (text frequency) that is based on the lemmata and not on the full forms of the extracted terms. In a second step we optimize the term weights by subtracting the inverse document frequency of the terms on the basis of a set of terms extracted from the documents analyzed so far. That is, the AME^{WS} keeps track of the term array and weights in a persistent memory.

The basic architectural design is shown in the following picture.



In fact, AME^{WS} provides different ways for the keyword extraction via different methods in the Web Service. On the one hand, it can return a list of all terms found in a text. This list can be sorted on the basis of their text frequency. The threshold of the amount of keywords in the keyword list can be specified as well. Similar possibilities are given for the extraction of other meta information as well, e.g. language of document, named entities, collocations of terms and named entities etc.

Beside the possibility to return basic data types, AME^{WS} can return a complex data structure as a RDF-document. The idea behind this functionality is to be able to generate document independent meta information in a standardized form, that can be stored independently. This allows for meta tagging of document formats that either do not provide the necessary extensions for storage of meta information, or that cannot be changed, due to legal restrictions and copyright issues. The RDF-data structure is generated with the use of the Jena toolkit (http://www.hpl.hp.com/semweb/jena-top.html).

3.2 Client Architecture

The first client is based on a C# component that interacts with AME^{WS} and is able to control applications like Microsoft Word via the automation API (and a COM-Interface). The client is closely linked to the functionality of Word. Pressing a button in the menu the document content is automatically send to AME^{WS}. The returned data structure contains information about the relevant lemmatized keywords in the document and other available meta information. This information is stored in the document properties via the functionality provided by Word. Furthermore a RDF file is stored together with the tagged document. The RDF file contains the extracted meta information in the DC-format and can be analyzed by search engines.

Another client is realized as a batch-analysis client. Documents in different formats (DOC, HTML, PDF, PPT, PS etc.) can be send to AME^{WS} and a RDF file with the extracted meta information is generated and stored as described above.

3.3 Further Steps

The next steps on the AME^{WS} roadmap are on the one hand a detailed evaluation of the quality of the extraction results (i.e. named entity recognition, keyword extraction). On the other hand, the effect of the extraction for search processes has to be evaluated on the basis of real search experiments.

One weak point of the linguistic analysis is the lack of anaphoric resolution, i.e. the reference of for example pronouns is not analyzed so far. This leads to a mismatch in the weighting schema of single terms. We will try to set up the necessary tools to analyze as many anaphoric references as possible in the subsequent versions of AME^{WS}.

The second version of AME^{WS} will be realized with a client that will be able to annotate HTML documents for intra- and internet sites.

References

1. Ćavar, D.: Generating Semantic Networks for Knowledge Management. 9th AIK Symposium Semantic Web. Karlsruhe/Germany, April 2002.
2. Ćavar, D. and Kauppert, R.: Strategies for Implementing IT-based Knowledge Management Solutions: Minimally Invasive Systems. C. Prange (ed.) Organizational Learning and Knowledge Management: Case Studies. Gabler, 2002.
3. Ćavar, D. and Störl, U.: Automatic Generation of Meta Tags for Intra Semantic Web. Proc. of XML Technologies for Semantic Web XSW 2002. Springer, June 2002.
4. Kokkelink, S. and Schwänzl, R.: Expressing Qualified Dublin Core in RDF/XML. http://dublincore.org/documents/2001/08/29/dcq-rdf-xml/. 2001.
5. Störl, U. and Deppisch, U.: XML-based Content Analysis: Vision and Reality. Proc. of Datenbanksysteme in Büro, Technik und Wissenschaft (BTW), Springer, March 2001.