A Knowledge Discovery Workbench for the Semantic Web

Jens Hartmann and York Sure
Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS/
email:{hartmann,sure}@aifb.uni-karlsruhe.de

Abstract

We present a workbench for integrating Web documents into semantically enriched representations suitable on the Semantic Web. The approach benefits on the one hand from the facilities provided by Semantic Web technologies and on the other hand from the applicability of well-known knowledge discovery techniques. The main achievement of our contribution is an up-and-running, open and component based prototype which can be easily extended by 3rd parties.

1 Introduction

The World Wide Web consists of information concerning nearly every imaginable topic represented by weakly structured Web documents. The process of searching and accessing relevant information on the Web leads often to a practical problem [1] hampered by the lack of semantic markup and missing inference capabilities [2, 3].

As an evolutionary step the Semantic Web [4] tends to overcome these problems by applying formal knowledge representation languages such as OWL [5] and enabling inferencing capabilities. Consequently, existing Web documents have to be translated into knowledge representations suitable for the Semantic Web, e.g. RDF(S) [6] or OWL [5]. Hence, we argue that the task of integrating Web documents for the Semantic Web acts a key challenge for the Semantic Web.

Our approach relies on a combination of knowledge discovery and semantic web technologies. It is built on top of the knowledge discovery process by [7]. Each step of the process is implemented by a component of our system. The developed system ARTEMIS is freely available[1]. We argue that extensibility of knowledge discovery systems and data mining algorithms is essential for successful real-world applications, as discussed in [8]. Hence, ARTEMIS is open and can be easily extended by 3rd parties. Further, we extend existing data mining methods with ontologies as background knowledge to improve (i) the mining task and (ii) the quality of created data models. This philosophy is also reflected by the software architecture itself: ARTEMIS uses semantic technologies in a component oriented software architecture.

[1] see http://artemis.ontoware.org

2 An Example of learned Document Models

We illustrate the impact of the ARTEMIS approach using results we achieved on classifying the Web site of the University of Bremen[2]. The goal was to learn classification rules that uniquely identify pages of the research group on theoretical computer science. For this purpose we used about 150 pages of that group as positive and about 300 other pages from the university Web site as negative examples. Table 1 shows generated rules for the different mode declarations and the accuracy of the rules.

Experiment A1-0		TrainingSet0
	TZI - Theory	
Mode Dec.	*Hypotheses*	*Acc.*
H 1	document(A) :- doctitle(A,research).	100
H 2	document(A) :- metatag(A,keywords, theoretical).	100
H 3	document(A) :- relation(A,B), relation(B,C), mail(C,helga,'informatik.uni-bremen.de').	86,82
H 4	document(A) :- relation(A,B), url(B,'[URL]/cs/ref.num.html'). document(A) :- relation(A,B), url(B,'[URL]/projects.html').	86,82
URL: `http://www.tzi.de/theorie`		

Table 1: An Example of Document Models

The results show the different kinds of classification rules (models) we get when using different elements of Web documents. Using the page title as a criterion, we find out that the pages of the theoretical computer science group are exactly those that contain the word 'research' in their title (H1). An analysis of metadata (H2) shows that the keyword 'theoretical' uniquely identifies the pages we are interested in. We get even more interesting results that still have an accuracy of more than 85% when analyzing e-mail addresses and links to other pages (H3). For the case of e-mail addresses we find out that most pages are linked over steps with a page that contains the mail address of the secretary of the group. If we only consider links (H4), we see that most pages are linked to pages containing references and to a page listing projects of the group.

3 ARTEMIS Workbench

The ARTEMIS Workbench represents a tool for knowledge engineers and industrial practitioners required to integrate large and heterogenous sets of documents whereby it provides functionalities of well-known knowledge discovery tools to generate semantic enabled document models to apply them on the Semantic Web.

To avoid such intricateness, ARTEMIS combines well-known knowledge discovery methods on the one hand and semantic technologies such as ontology-based knowledge engineering and reasoning techniques on the other hand. This combination is realized by an expressive and easily extendable component architecture with semantic enriched interfaces.

[2] http://www.uni-bremen.de

3.1 General Overview

The workbench consists of three main blocks: (i) The **ARTEMIS Core System (ACS)** as surrounding technology, (ii) the **Workflow Model (WM)** providing a knowledge discovery workflow and (iii) the **Component Model (CM)** instantiates the workflow by extensible components as presented in figure 1.

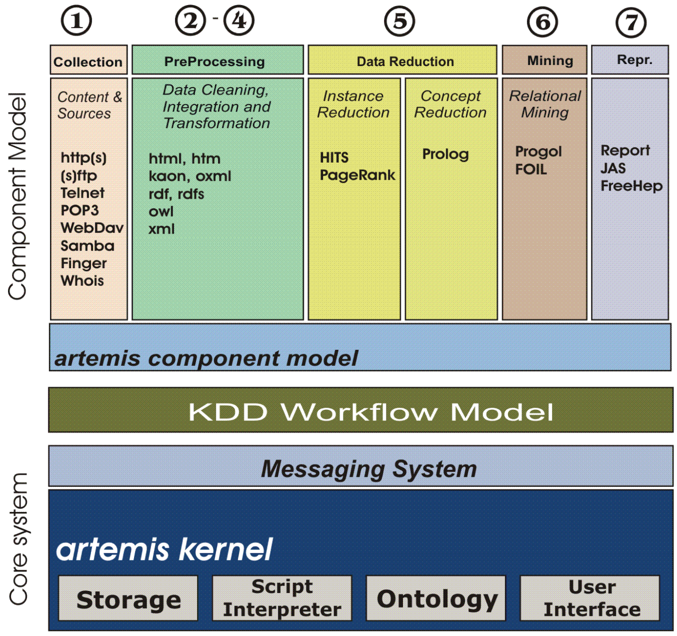


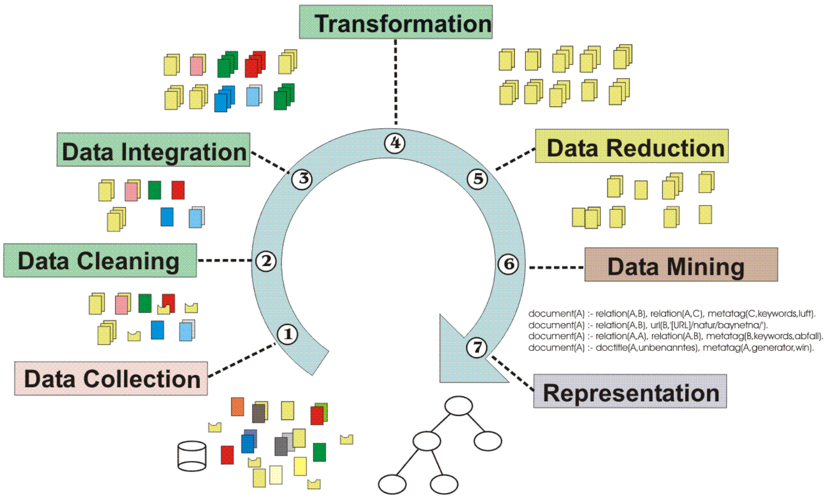
Figure 1: ARTEMIS Architecture

The ARTEMIS *Core System* contains the main system functionalities which are subdivided into the *kernel* and the *messaging system*. The *kernel* provides core functionalities for the workbench like realising **storage mechanisms**, running a **script interpreter** and providing the **ARTEMIS ontology** for the components.

The *Component Model* provided by ARTEMIS instantiates the knowledge discovery process of the *Workflow Model* and provides components for each process step. A component used within ARTEMIS provides a semantic description in form of an ontology which (i) allows to classify the type of component according the workflow model and (ii) provides a set of services to the ARTEMIS workbench, e.g. a text classification algorithm.

3.2 Workflow Model

The accomplishment of a knowledge discovery process is handled by the **Workflow Model** which provides a workflow manager to monitor the flow of data and extracted information. Further, it assures the application of components depending on the current process step. Our approach instantiates the knowledge discovery process presented in [7, 9]. As indicated in Figure 2 ARTEMIS provides for each step of the process specialised components.



```
document(A) :- relation(A,B), relation(A,C), metatag(C,keywords,luft).
document(A) :- relation(A,B), url(B,'[URL]/natur/baynetna/').
document(A) :- relation(A,A), relation(A,B), metatag(B,keywords,abfall).
document(A) :- doctitle(A,unbenanntes), metatag(A,generator,win).
```

Figure 2: ARTEMIS Workflow

4 Knowledge Representation

In order to use the PROGOL system for generating document models , we have to encode knowledge about Web documents and their internal structure in PROLOG. For this purpose, we developed a representation scheme consisting of a set of pre-defined predicates.

- `document(object)`: the constant ’object’ represents a document
- `url(object, ADRESS)`: the document represented by ’object’ has the URL ’ADDRESS’
- `relation(doc1, doc2)`: there is a directed link between the document ’doc1’ and ’doc2’
- `structure(object, CLASS)`: the constant ’object’ represents an element tag of type ’CLASS’
- `contains(doc, object)`: the document contains the tag ’object’ as a top level element.
- `attribute(parent, object)` the element tag ’parent’ contains the attribute ’object’
- `contains(parent, object)` the element ’parent’ contains the element ’object’ as a child element
- `value(object, ’VALUE’)`: ’object’ is an element or attribute and it has the value ’VALUE’
- `text_value(object, ’TEXT’)`: ’object’ is an element or attribute and it has the text ’TEXT’

In order to be able to use an ILP learner for the acquisition of document models, the structure of the documents serving as positive and negative examples have to be translated into the representation described above. Unfortunately, most of the documents came in less standardized form, partly containing syntactic errors. Therefore all training examples were semi-automatically cleaned and tidied up. We use HTML Tidy[3] and its Java pendant JTidy[4] for this task.

The next step to obtain a usable training set is the *syntactical translation* of the training examples. A Web document like a HTML or an XML Document contains predefined tags which describes structure (in particular relations inside a document or between other documents) and layout of documents. The complete translation process is described here in a very abstract way: (i) Every document is parsed into a DOM tree. We use Apache JXERCES 2.0 for this task. (ii) ARTEMIS then walks through the DOM tree. Depending on a predefined translation scheme all desired tags are translated into PROLOG clauses. (iii) The positive and negative examples are stored into a database which represents the training set. (iv) In order to enable the system to perform a restricted kind of learning on the text of a page, simple normalization techniques are applied that convert the words of a text into lower case letters, removes special symbols as well as words from a stop list and inserts a list of the remaining words in the PROLOG notation. More details can be found in [10].

5 Conclusion

We presented an approach for automatically acquiring models from Web documents applicable on the Semantic Web. The approach can be used to integrate Web documents with semantic markup in terms of an assignment to certain ontologies for building repositories or data warehouses. We discussed the architecture and its provided component model extensible by 3rd parties.

Acknowledgements

Research reported in this paper has been partially funded by EU in the IST project SEKT (IST-2003-506826), the network of excellence Knowledge Web (IST-2004-507482) and has been funded by the german BMBF (federal ministry of education and research) project SemIPort. We thank all our colleagues at AIFB for fruitful discussions.

References

[1] Chakrabarti, S.: Mining the Web: Discovering knowledge from hypertext data. Morgan Kaufmann, San Francisco (2003)

[2] Rindflesch, T., Aronson, A.: Semantic processing in information retrieval. In Safran, C., ed.: Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC 93), McGraw-Hill Inc., New York (1993) 611–615

[3] http://www.w3.org/People/Raggett/tidy/

[4] http://lempinen.net/sami/jtidy/

[3] Zweigenbaum, P., Bouaud, J., Bachimont, B., Charlet, J., Séroussi, B., Boisvieux, J.: From text to knowledge: a unifying document-oriented view of analyzed medical language. Methods of Information in Medicine **37(4-5)** (1998) 384–393

[4] Berners-Lee, T.: Weaving the Web. Harper (1999)

[5] Smith, M.K., Welty, C., McGuinness, D.: OWL Web Ontology Language Guide (2004) W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/owl-guide/.

[6] Brickley, D., Guha, R.V.: RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004 (2004) available at http://www.w3.org/TR/rdf-schema/.

[7] Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: Knowledge discovery and data mining: Towards a unifying framework. In: Knowledge Discovery and Data Mining. (1996) 82–88

[8] Wrobel, S., Wettschereck, D., Sommer, E., Emde, W.: Extensibility in data mining systems. In Simoudis, E., Han, J.W., Fayyad, U., eds.: Proc. 2nd International Conference On Knowledge Discovery and Data Mining, Menlo Park, CA, USA, AAAI Press (1996) 214–219

[9] Chang, G., Healey, M.J., McHugh, J.A.M., Wang, J.T.L.: Mining the world wide web (2001)

[10] Stuckenschmidt, H., Hartmann, J., van Harmelen, F.: Learning structural classification rules for web-page categorization. In: Proceedings of FLAIRS 2002, special track on Semantic Web. (2002)