Knowledge Search within a Company-WIKI

Stephanie Müller[1], Nils Kritzler[1], Alexander Tartakovski[1], Ralph Bergmann[1], and Ralph Traphöner[2]

[1]University of Trier,
Department of Business Information Systems II,
54286 Trier, Germany
{muel4102|krit4101}@uni-trier.de
{Alexander.Tartakovski|bergmann}@wi2.uni-trier.de

[2]Ralph Traphöner
empolis GmbH
Europaallee 10, 67657 Kaiserslautern, Germany
ralph.traphoener@empolis.com

Abstract

The usage of Wikis for the purpose of knowledge management within a business company is only of value if the stored information can be found easily. The fundamental characteristic of a Wiki, its easy and informal usage, results in large amounts of steadily changing, unstructured documents. The widely used full-text search often provides search results of insufficient accuracy. In this paper, we will present an approach likely to improve search quality, through the use of Semantic Web, Text Mining, and Case Based Reasoning (CBR) technologies. Search results are more precise and complete because, in contrast to full-text search, the proposed knowledge-based search operates on the semantic layer.

1 Introduction

The concept of Wiki [Baeza-Yates, 1999] provides a simple and efficient way of creating knowledge and making it accessible. It is suited especially for the purpose of knowledge management within a company because of the great acceptance by employees.

However, the authoring simplicity results over time in a large amount of steadily changing, unstructured documents. Consequently, the users often lose the overview of available content. Full-text search, which is usually implemented within Wiki-systems, does not help sufficiently to overcome that problem [Cesarano *et al.*, 2003]. It does not take the relationships between concepts and objects, synonyms, and multilingualism among other things into account and therefore often provides insufficient search-results [Cesarano *et al.*, 2003; Money and Turner, 2004]. In this situation user acceptance decreases, since the desired information may often only be found after several attempts using full-text search.

The same situation could be observed at empolis GmbH[1] after the introduction of a Wiki for the purpose of knowledge management. Since its implementation in February 2004, the amount of pages increased up to 5,500 in November 2004. Following this considerable increase the user acceptance began to decrease.

empolis GmbH and the Department of Business Information Systems II, University of Trier launched a project with the objective of overcoming the explained difficulties by developing a knowledge-based search function to enable improved access to the information filed in the Wiki.

In this paper, we present the concept and the realisation of the search function using a combination of following technologies: Semantic Web, Text Mining, and Case Based Reasoning (CBR) [Davenport and Prusak, 1998; Davenport and Grover, 2001; Leuf and Cunningham, 2001]. A domain specific ontology provides a vocabulary for the semantic annotation of the content. The annotation is constructed automatically with the help of text mining technology. Similarity-based search on the semantically observed content is done using CBR-retrieval technology.

The approach to use semantic information to improve the search functionality within a Wiki has also been followed in the Semantic MediaWiki project[2]. There, the authors of the Wiki article enter semantic information themselves. In contrast, following our approach, semantic information is allocated automatically to each article.

Section 2 describes the application of Wiki within a company in terms of knowledge management. While section 3 introduces the concept of knowledge-based search, section 4 demonstrates its realisation. The last section concludes the paper with summary and discussion.

2 Wiki for emphasising knowledge-sharing

“Increasingly, knowledge is recognized as an organization’s most valuable resource and the best

[1] empolis is an arvato AG subsidiary, an international media service company and part of Bertelsmann AG. It is supplier of enterprise content and knowledge management solutions.

[2] http://wiki.ontoworld.org/index.php/Semantic_MediaWiki

foundation of sustained competitive advantage" [Maedche, 2002]. Knowledge management is rapidly becoming an integrated business function as companies realise that effective management of intellectual resources is connected to competitiveness [Abecker and Elst, 2004]. The difficulty lies in gathering knowledge as well as its creation, allocation, storage and location.

One instrument to organise and cross-link knowledge is a Wiki [Baeza-Yates, 1999]. This concept offers a forum for its users to share knowledge and look up information. It simplifies and encourages knowledge sharing, as its usage is simple and quick. The handling of Wiki does not require conformity to many rules and there is also no need to setup specific software.

These characteristics lead to a lack of formal structure as well as a dynamic changing landscape, which makes it very difficult to keep an overview of the content. In particular the constant growth, which is anarchical and uncontrolled, makes this task more and more complicated. Additionally, many inner-company Wikis are kept in several languages, which aggravates the task.

The main aim of a Wiki is the re-usability of knowledge. Its existence is only of interest if not just the storage of knowledge is realised in an easy and uncomplicated way but also the location of the stored information is quick and simple to reference. To reach this objective, the improvement of the search functionality is needed to enable relevant information to be found more easily and is presented in the following.

3 Knowledge-based search supported by the concept of ontology

The approach most used within a Wiki is full-text search; but it is already widely known that results are not satisfactory. Using full-text search, a result is only a 100% hit, if the title of an article corresponds exactly to the query. The problem of this method is the total ignorance of similarities between words like singular and plural or different words used for the same thing; multilingualism is not cared for either. The listing of results contains many irrelevant articles, misses out several relevant documents and the ordering of relevance does not reflect the real order of importance of the results. This insufficient search functionality leads to the decrease of usage of Wiki for knowledge sharing.

To improve the insufficient search functionality, the approach of a knowledge-based search function is presented in this paper. Following this approach, ontology provides the necessary background knowledge. Outgoing from this knowledge, text miner software automatically annotates the unstructured documents with semantic information. Then a case base reasoning suite is used to represent the achieved semantic content of the articles as cases in a case base and to perform the similarity based search.

3.1 Semantic annotation and knowledge-based search

The first step while developing knowledge-based search functionality is the creation of semantic annotations, which represent the content of every Wiki article. According to the approach presented in this paper, every single annotation consists of a set of concepts, which are identified within a Wiki article by the text mining software. Search is then performed by comparison of the query with the annotations of the several Wiki documents. This kind of search provides a faster access to the content of a Wiki. Regarding its usually large number of documents, which is also constantly increasing, this methodology is most appropriate in that context. If annotations are constructed accurately by the text mining software it is also possible to offer better finding of relevant information.

The main problem that has to be solved is the ambiguity of natural language; it manifests itself in the synonym and polysemy phenomenon [Money and Turner, 2004]. The synonym phenomenon refers to the problem that the same concept can be represented in many different ways. The fact that words can have different meanings in different contexts is defined by polysemy [Cesarano *et al.*, 2003]. To provide a fast and reliable knowledge-based search, the knowledge of the language use within a Wiki is essential. In particular, the problem of imprecise interpretation of the search-query and the consequential need of "processes to ‚interpret' the query, to retrieve the expanded query condition according to the interpretation, and to evaluate the closeness of the result to the original query" [Liu, 2001] require the knowledge of the language use within the Wiki. Without this interpretation based on the knowledge, satisfactory results for the user query are not possible.

As the Wiki landscape is changing constantly, manual annotation of the various articles is not appropriate in this context. For automatically annotating documents, its content has to be classified by the text mining software without human intervention. Consequently, the text mining software requires a knowledge base including domain specific language knowledge in order to use it for the annotation purposes.

3.2 Ontologies to provide the necessary background knowledge

The domain specific knowledge is represented by the usage of ontologies. It provides a common understanding of things of the world and for that reason is means to bridge the 'semantic gap' existing between the actual syntactic representation of information and its conceptualisation [Davenport and Grover, 2001].

Ontologies are the key means to annotate unstructured documents with semantic information, to integrate information and to generate specific views that make knowledge access easier [Davenport and Grover, 2001]. They provide the domain knowledge for the realisation of knowledge-based search.

Before mapping Wiki articles according to the domain ontology, the latter one demonstrating the conceptual model of the Wiki has to be created. This scheme represents the set of concepts, instances and relationships which map the content of the Wiki. A thesaurus completes that model; synonyms, pseudo-synonyms as well as acronyms are included to enhance semantic understanding [Cesarano *et al.*, 2003].

After the ontology is created, the metadata of various articles can be produced, which means annotating the documents. The extracted words of an article are mapped to the concepts of the ontology. The annotation resulting from this process consists of ontology-concepts, which present the content of each article. It is stored together with the corresponding article and is available for the search function from then on. To support the search functionality it is appropriate to annotate each article after creation or editing.

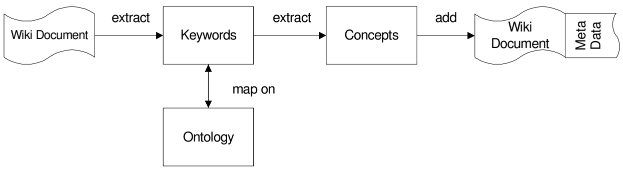

Figure 1 Process of the creation of metadata

A document retrieval process can be carried out in the following way. First, a query has to be annotated with metadata in the same manner as every Wiki article. During the search process, the annotation of the query is compared with the metadata of the articles. Afterwards, the articles having metadata with a high similarity to the metadata of the query are presented to the inquirer, ordered with respect to similarity and diversity.

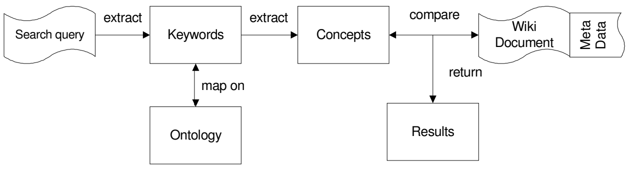

Figure 2 Query processing for retrieval

4 Realisation

The realisation of the above explained theoretical approach can be divided into two parts: The creation of the ontology and the development of the search functionality itself.

4.1 Identification of the specific domain knowledge of the Wiki

As the ontology needs to model the content of the Wiki, the domain knowledge of the Wiki has to be detected.

The starting point for this procedure is the collection of all articles contained in the Wiki; this set is named 'corpus' in the following text. To find out which words of the corpus reflect the content of the Wiki, word frequency lists are used. The word frequency list is built with the help of concordance programs. A concordance can be described as an alphabetical index of all the words in a text or corpus of texts, showing every contextual occurrence of a word.

The first step is sorting out useless words that do not describe the content of the articles. These words are called 'stop words' and are so common that they are worthless in giving any information about the essence of an article. The result is the concordance of the remaining words. These are ordered according to their frequency. Next, it is necessary to define how often a word has to be present to be important enough for assimilation into the ontology. Words with a lower frequency are deleted accordingly. That choice is dependent on the total number of words on the list.

After this proceeding, the word frequency list still contains several words that do not imply any relevance for the ontology. As language is ambiguous, it is not possible to eliminate every useless word during the concordance process. This has to be done manually. There is also a need to remove words that might be significant but are used in several contexts and thus their meaning is not definite. After the removal of these insignificant words, the content of the resulting list reflects the environment of the Wiki. The list at this stage is the initial point in creating the ontology.

4.2 Ontology creation process

Starting with the modified word frequency list, the manual creation process of the ontology can take place. To keep the overview, a visualisation tool is used.

"There is no 'correct' way or methodology for developing ontologies" [Maletic and Marcus, 2001], several approaches exist, depending on the application that one has in mind. One possible way is to start with a rough first pass, which is then refined in an iterative process [Maletic and Marcus, 2001]. As this methodology fits into the given context we decided to use it.

This is done by allocation of a concept to every word of the list. The resulting concepts have to be ordered by the 'kind-of' relation, which is well known from the oo-modelling. Thus, while filling the ontology it has to be decided where to put every concept into the hierarchical scheme. Following this procedure, additional facts have to be taken into account: sometimes more than one place exists where the concept belongs or there is the necessity to create additional concepts to merge several concepts. The last step within this process is the consideration of whether there is a need to create additional concepts, which fit into the given context and therefore enrich the ontology.

Apart from the 'kind-of' relation, which is already contained in the hierarchical scheme, other sorts of relations between several concepts are created as well. Because the CBR-based search functionality is intended to be performed it is sufficient to define solely the strength of connection between concepts, instead of complete and explicit definition of all relations. The strength of a connection can be evaluated to find relevant search results. This is realised by similarity weights, which range from 0 for 'no relation' to 1, which means 'equals'. This provides a model that is applicable for any CBR-suite and enables it to retrieve documents that contain similar keywords to those formulated in the query.

The last step is the creation of a thesaurus for every concept. This feature supplies a base of keywords, which stands for the respective concept and serves for the semantic annotation executed by the text mining functionality. The figure below shows an extract of an example ontology.

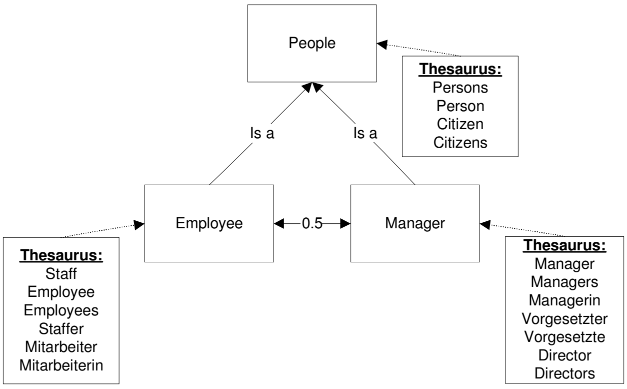

Figure 3 Extract of an example ontology

4.3 Embedding the ontology into CBR-Suite

To execute the search it is necessary to transform the ontology into a data model of a CBR-suite.

Most CBR-suites adopt a classic structural CBR approach, where each case is described by a finite and structured set of attribute-value pairs. The data model defines attributes allowed to be used for case description, and a global similarity function, which is used to compare queries with cases. Attribute names and respective types representing feasible value ranges define attributes. The global similarity function is usually defined according to the local-global principle. First, local similarity functions are defined for every attribute. Such a function compares two values from a certain attribute type; i.e. it compares a query and a case only with respect to a chosen attribute. To achieve the global similarity measure, local similarities are aggregated by an aggregation function, which provides a possibility to compare any query and any case to each other.

Unfortunately, an ontology that encodes an object oriented data model cannot be mapped one-to-one to the attribute-value structural CBR model. In the following the transformation approach is presented, which has been developed during realisation of the knowledge based search functionality.

1. At the beginning, the first proportion of attribute names has to be defined. Candidates for these attribute names are the names of those concepts that are located at the top of the inheritance hierarchy within the ontology. According to the extract of the ontology displayed above, the corresponding case model would include the attribute 'People'. A final version of the case model which was developed for the empolis Wiki includes also attributes: 'Software', 'Hardware', 'Companies', 'Platform', 'Service', and so on. Within the final ontology these are the names of the concepts at the top of the hierarchy.
2. The next task is the definition of attribute types. Since the attribute-value structural CBR approach does not support inheritance explicitly, a slightly unnatural modelling manner has to be chosen. All the names of subconcepts, which are located within the inheritance tree of the ontology under a certain top-level concept, can be understood as symbols, which build the type of the attribute originated from that top-level concept. According to the extract of the ontology, the type of the attribute 'People' is {Employee, Manager}, since the concepts 'Employee' and 'Manager' are both subconcepts of the top-level concept 'People'. An improvement of this approach, in order to reflect the Wiki articles more precisely, could be achieved by the usage of power sets instead of simple sets of symbols. Consequently, if within some Wiki article both concepts 'Employee' and 'Manager' are found, the value of the attribute 'People' could be {Employee, Manager}.
3. In order to reflect the inheritance relation, the symbols have to be ordered with a taxonomic relation in exactly the same way as the 'kind-of' relationship. The taxonomic relation provides important information for the calculation of the local similarity. Based on the location of two symbols within taxonomy, a similarity to each other can be expected.
4. The next step is the transformation of non-inheritance relations, which are implicitly defined within the ontology by the strength of connection between concepts. For every relation, an additional attribute within the case model should be introduced. An attribute type has to include all symbols that originate from the names of concepts affected by the relation. The connection strength between concepts from the ontology can be directly taken over as similarity between appropriate symbols.
5. The last step is the accomplishment of the similarity function. The local similarities are already modelled within the previous steps. For the first part of attributes defined in the steps 1-3 the local similarity is given by taxonomy. It can be further adjusted depending on the concrete realisation within the CBR-suite. For the second part of attributes, which encode non-inheritance relations, the local similarity measure is explicitly given by the connection strength between concepts. There is no further necessity to adjust this measure. In order to get the global similarity measure, the local similarities are usually aggregated by the normalised, weighted sum. The weights can be adjusted during the tuning of the search function.

After creation of the case-model, all available Wiki articles should be analysed with text mining software. Hereby, the text miner applies the thesaurus defined within the ontology in order to identify a set of concepts for every Wiki article. The resulting sets of concepts are saved as cases according to the case-model. The following figure shows some example documents with corresponding cases.

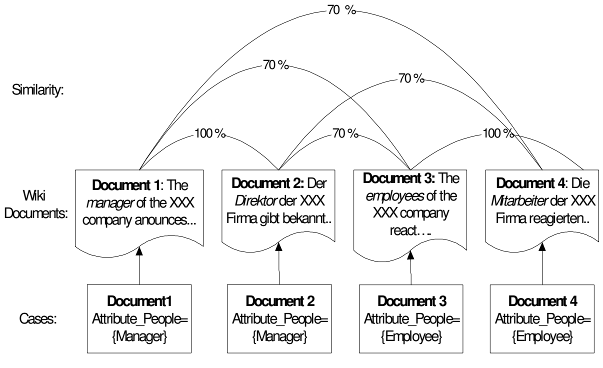
70 %
70 %
70 %
Similarity:
100 %
70 %
100 %
Wiki Documents:
Document 1: The manager of the XXX company anounces...
Document 2: Der Direktor der XXX Firma gibt bekannt..
Document 3: The employees of the XXX company react…
Document 4: Die Mitarbeiter der XXX Firma reagierten..
Cases:
Document1 Attribute_People={Manager}
Document 2 Attribute_People={Manager}
Document 3 Attribute_People={Employee}
Document 4 Attribute_People={Employee}

Figure 4 Creation of the case base

The attribute values of cases, which are mapped to Document 1 and Document 2, are equal. The reason for that is the fact that the words ‘manager’ and ‘director’ are included in the thesaurus and are mapped to the same concept ‘Manager’ which is transferred to the value ‘Manager’ of the attribute ‘People’ within both cases. As the cases of both documents contain exactly the same attribute values, they have a similarity of 100 %. To compare document 2 and document 3, their cases show attributes with similar values. The reason for that is the fact that attribute values ‘Manager’ and ‘Employee’ are similar with respect to the taxonomy in the case model. As a result these cases are rated by similarity of 70 %.

The performance of the similarity based search starts with the input of the search query. With the usage of the text miner and the case model, a new case outgoing from the query is created. Next, the similarity function calculates the similarity of this new case to the ones in the case base and retrieves the set of most similar cases. They are represented to the user in order of relevance.

To realise the search functionality we used the open retrieval engine orenge, which has been developed by the project partner empolis GmbH. It has been created specifically to execute intelligent case-based and knowledge-based searches and for that reason it was useful for our purposes.

5 Outlook and Conclusion

This paper addresses the problem of decreasing acceptance of inner-company Wikis, which occurs when the content becomes large and chaotic. The acceptance by employees decreases on the one hand because of loss of an overview over the available information and on the other hand because of lack of search functionality. Full-text search, which is usually implemented within Wiki-systems, does not support the users sufficiently since the quality of the search-results is low.

The intention of the project described in this paper is to make the application of Wiki for the purpose of knowledge management within companies more attractive by development of improved search functionality.

We introduced a realisation approach to knowledge-based search functionality, which is likely to outperform full-text search. Several characteristics of this approach indicate better search results. But, this statement still has to be proven by a following evaluation. Another interesting idea for a future evaluation is the comparison of the developed search functionality according to other information retrieval approaches.

According to the realised approach, the domain knowledge of the Wiki is represented via ontology, which is created in the semiautomatic manner. For this purpose, the whole document corpus is analysed using concordance programs and, after manual validation, the remaining data can be taken over into the ontology as concepts and relations. Following, after manual validation and extension, the ontology is embedded in a CBR-suite.

Each document from the corpus and each query is semantically annotated with text mining software contained in the CBR-suite, which has access to the constructed domain ontology. Each semantic annotation of any document or any query is regarded as a single CBR-case. The search for the relevant documents is then carried out as a CBR retrieval process.

Based on the meaningful domain model, the quality of search results is expected to increase to a high extent. The provided knowledge guarantees that the annotation is of a high quality and matches the content of the articles. Knowledge-based search copes easily with the weaknesses of full-text search such as “the gap between the user’s information need and the actual query strings they specify” [Cesarano *et al.*, 2003]. It finds relevant articles regardless of which synonym is used to formulate the query. Another advantage is the support of multilingualism and different word forms. Results are represented according to relevance; that means not only 100 % hits are displayed, but also articles with related content.

However, it has to be considered that good search results depend exclusively on a good data model. The richer it is the better are the results. As it is of such importance, attention has to be paid that the model is extensive and correct. Furthermore, the search only operates sufficiently if the data model spans the whole context of the provided database. This makes a continuous improvement of the model extremely important. But it has to be considered that this process is extremely time-consuming as well as costly. One approach that can be investigated in a further attempt is the utilisation of the cross-linking characteristic of a Wiki to automatically build and maintain the data model. Generally, effective maintenance is extremely important in order to achieve good results. If done continuously, this guarantees a good search functionality that works within unstructured documents and outranges full-text search to a great extend.

References

[Abecker and Elst, 2004] A. Abecker and van L. Elst van. Ontologies for Knowledge Management. In: Staab, S., Studer, R. (Editors), Handbook on Ontologies, Berlin 2004, Pages 435-454.

[Baeza-Yates, 1999] R. Baeza-Yates. Modern information retrieval. Addison-Wesley, New York 1999.

[Cesarano *et al.*, 2003] C. Cesarano, A. Acierno d’, A. Picariello. An Intelligent search Agent System for Semantic Information Retrieval on the Internet: In: Proceedings of the 5th ACM international workshop on Web information and data management. New Orleans, Louisiana, USA, 2003, Pages 111-117.

[Davenport and Grover, 2001] T. Davenport and V. Grover. General Perspectives on Knowledge Management: Fostering a Research Agenda. In: Journal of Management Information Systems, Volume 18, Issue 1, 2001.

[Davenport and Prusak, 1998] T. H. Davenport and L. Prusak, Working Knowledge: How Organisations manage what they know, Harvard Business School Press, Boston, Mass. 1998.

[Leuf and Cunningham, 2001] B. Leuf and W. Cunningham. The Wiki Way Quick Collaboration on the Web, 1. Print, Addison-Wesley, Boston, Mass. [and others] 2001.

[Liu, 2001] H. Liu. Intelligent Search techniques for large software systems. Thesis. Ottawa-Carleton Institute for Computer Science, School of Information Technology and Engineering, University of Ottawa 2001.

[Maedche, 2002] A. Maedche. Ontology Learning for the Semantic Web. 1. Edition, Kluwer Academic, Dordrecht 2002.

[Maletic and Marcus, 2001] J.I. Maletic and A. Marcus. Supporting Program Comprehension Using Semantic and Structural Information. In: Proceedings of 23rd ICSE, Toronto 2001, Pages 103-112.

[Money and Turner, 2004] W. Money and A. Turner. Application of the Technology Acceptance Model to a Knowledge Management System. In: Proceedings of the 37th Hawaii International Conference on System Sciences, 2004.