Improving Intranet Search-Engines Using Context Information from Databases

Christoph Mangold, Holger Schwarz, Bernhard Mitschang
Universität Stuttgart, IPVS
Universitätsstr. 38, D - 70569 Stuttgart
firstname.lastname@informatik.uni-stuttgart.de

ABSTRACT

Information in enterprises comes in documents and databases. From a semantic viewpoint, both kinds of information are usually tightly connected. In this paper, we propose to enhance common search-engines with contextual information retrieved from databases. We establish system requirements and anecdotally demonstrate how documents and database information can be represented as the nodes of a graph. Then, we give an example how we exploit this graph information for document retrieval.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Design, Management.

Keywords: Intranet search, context.

1. INTRODUCTION

Enterprise intranets contain two classes of information. On the one hand, enterprise workflows require documents such as email notifications, plans, reports, meeting minutes, web pages etc. Typically, documents reside in content management systems or shared directories. On the other hand, databases contain information for planning, operational management, controlling etc. Our approach bases on the notion that effective document retrieval in enterprises requires both information kinds.

Current search engines exploit the document content and limited meta data such as document title, author, and file name. So-called "semantic" search engines additionally involve context information from domain-specific ontologies which are provided by domain experts and knowledge engineers. In contrast, our approach relies on the expertise and quality of the enterprise's own information systems. Typically, these information systems rely on (relational) database technology. They are highly accurate and comprise all mission critical data. We open up these information systems to enhance common document retrieval systems.

CIKM'05, October 31–November 5, 2005, Bremen, Germany.
ACM 1-59593-140-6/05/0010.

In Section 2 we discuss important requirements. We present a model to capture document context, the ContextGraph, in Section 3. Section 4 covers related work and Section 5 concludes the paper.

2. REQUIREMENTS

In this section, we discuss four of the main requirements for a system that exploits database information to improve document retrieval capabilities.

Information kinds. We do not only want to exploit context information from databases, but also we want to open up the system for ontological information (if available). We discuss related work in the semantic search area in Section 4.

Keyword search. Many semantic search engines require the user to encode his information need as a formal language expression. We align our engine with average intranet users who have no formal training. Consequently, the engine has to cope with search keywords.

Performance. Performance and scalability are crucial requirements for search engines. Hence, similar to ordinary search engines we need an index-based document-retrieval system, i.e. in an off-line process the system analyzes and pre-processes relevant information and builds up an index. The index is an efficient data structure which allows to retrieve relevant documents for sets of search terms.

Appearance. In terms of user interaction, our system should behave similar to ordinary search engines. This not only comprises performance but also search semantics and ranking. We plan to achieve this by extending an existing search engine. Among others, this implies that we have to find meaningful ranking measures that naturally enhance current measures such as, e.g. tf.idf.

In the next section, we show how we deal with different information kinds and give an impression how we realize keyword search.

3. THE CONTEXT GRAPH

To enrich document context with information from databases we propose a graph-based model which is comparable to the RDF representation of an ontology. The structure of the graph is determined by the enterprise's database schemas. Documents, relational tuples, and values are represented as graph nodes. Attribute and foreign-key relationships are represented as edges. Since the graph models the context of documents we call it ContextGraph.

Figure 1 gives an impression of a ContextGraph in a car-engine factory scenario. The middle node represents a ma-

chine report written by Peter Brown, which is stored in file:// Z:/docs/xr/2341.pdf. The report contains information about an oil flux problem on finishing machine N23-17. For machine reports and service reports, the enterprise's document management system stores the following meta data: Document type, author, concerned machine, and document abstract. Additionally, from a production-planning database, we know that engine blocks of type K1000 are finished on machine N23-17.

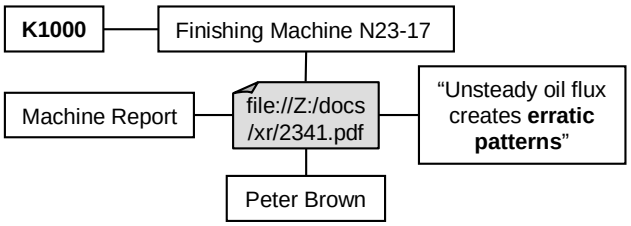

Figure 1: Small fraction of a ContextGraph.

Consider the following scenario: An employee in the quality-assurance division detects erratic finishing patterns on an engine block of type K1000. She inputs the query "K1000 erratic pattern" to her search engine. If there is no document containing all search terms, a standard document retrieval system returns many documents that contain an arbitrary subset of the given terms. The system has no means to decide which document fulfills the user's information need best. Consequently, the employee has to skim a large result set or to rewrite her query which possibly requires her to ask colleagues or to query the enterprise's databases. Our approach supersedes these activities by returning the machine report that describes that erratic patterns may occur on machine N23-17 that *produces* engine blocks of type K1000 and rank it as highly relevant. This can be achieved by exploiting connection information from the ContextGraph.

4. RELATED WORK

In the database area, related work focuses on database exploration. One typical scenario comprises a user who is running keyword queries against a database with unknown schema. Approaches from this domain do not consider documents, but databases only. Where the approach in [5] deals with Lorel databases, there are also a number of solutions for relational databases, e.g. [1, 8]. The latter aim at joining tables and then retrieving tuples that contain the search terms each.

The BANKS system [2] captures relational databases by means of a graph model that is similar to the ContextGraph. The system accepts keyword-queries as user input. It returns a set of subgraphs, each of which comprises at least one hit-node for each search keyword. Subgraphs are ranked according to node importances and edge weights which are determined at indexing time. BANKS is not aware of documents.

In the area of web information retrieval, there are approaches that use spreading activation algorithms on an interlinked web space to find related pages for a given set of web pages [4] or to realize keyword search [9]. These approaches are "highly costly" [9] since they do not make use of indexes.

In the Semantic Web area there are a number of solutions, e.g. [3, 6, 7, 10], that require web pages that commit themselves to ontologies with certain structures. These approaches do not apply to the ContextGraph since they mostly rely on concept hierarchies and linguistic information like synonyms or antonyms. Furthermore, they are restricted to deal with ontological concepts and can not exploit text fragments like document title or abstract. Consequently, they are not capable to exploit the high quality information from the enterprise's databases. In general, ontologies that match these requirements need to be created and custom-tailored manually for each application domain. This is not only costly but also provokes correctness and consistency issues. In contrast, the ContextGraph represents established and reliable information that is consistent with and relevant for the enterprise's business.

5. CONCLUSION

In this paper, we proposed a database supported approach for index-based semantic document-retrieval in enterprises. The motto of our approach is to exploit existing knowledge. Instead of relying on domain experts to create domain-specific ontologies, we rather use the valid, consistent and heavily used information that is available in the enterprise's databases. Currently, we are implementing a prototype that shows promising results.

Acknowledgements

The work in this project is partially funded by the German Research Foundation (DFG) in the scope of the Collaborative Research Center (SFB) 467.

6. REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In *ICDE*, 2002.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *ICDE*, page 431, 2002.

[3] A. Burton-Jones, V. C. Storey, V. Sugumaran, and S. Purao. A heuristic-based methodology for semantic augmentation of user queries on the web. In *Intl. Conf. on Conceptual Modeling, ER'03*, 2003.

[4] F. Crestani and P. L. Lee. WebSCSA: Web search by constrained spreading activation. In *IEEE ADL 99 - Advances in Digital Libraries Conf.*, 1999.

[5] R. Goldman, N. Shivakumar, S. Venkatasubramanian, and H. Garcia-Molina. Proximity search in databases. In *VLDB*, 1998.

[6] R. Guha, R. McCool, and E. Miller. Semantic search. In *WWW*, 2003.

[7] J. Heflin and J. Hendler. Searching the web with SHOE. In *AAAI-2000 Workshop on AI for Web Search*, 2000.

[8] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-style keyword search over relational databases. In *VLDB*, 2003.

[9] C. Rocha. A hybrid approach for searching in the semantic web. In *WWW*, pages 374–383, 2004.

[10] N. Stojanovic. On analysing query ambiguity for query refinement: The librarian agent approach. In *Intl. Conf. on Conceptual Modeling, ER'03*, 2003.