Domain Knowledge Based Document Retrieval

Csaba Dezsényi, Tamás Mészáros

Department of Measurement and Information Systems
Budapest University of Technology and Economics
Műegyetem rkp. 9., Budapest, Hungary, H-1521
E-mail: dezsenyi@mit.bme.hu, meszaros@mit.bme.hu

Abstract

In this paper authors present a system for automatic document retrieval from the World Wide Web. Systems on the market typically use traditional search engines for document retrieval that are not effective enough. The proposed information retrieval agent uses domain knowledge for supporting the information extraction from documents, and it also uses a model of the human information retrieval behavior. The architecture and knowledge base of the agent is discussed in detail.

Keywords: text analysis, information retrieval, intelligent agent, document retrieval

1. INTRODUCTION

The most impressive achievement of technology in the past decade was the spread of the Internet, which affects the behavior and attitude of the whole world from both aspects of technology and culture. The recent, rapid growth of the World Wide Web has led to enormous amounts of on-line information. However, as the volume of this information has increased, so have the problems encountered by users dealing with it. The millions of documents on the net are not as well structured as in a library, so spending hours in front of our PC screen, searching for information is an everyday situation.

This paper presents a web based document retrieval application, which is a part of the Information and Knowledge Fusion (IKF) project [1,2][1]. The purpose of the project is the development of knowledge-based information retrieval systems for financial corporations and banks. The system searches and retrieves topic specific information from different sources (Internet, intranet resources, data warehouses, etc.) and provides this information for the users in a structured way. The process involves the methods for gathering information from unstructured text documents using natural language processing, text mining and indexing.

[1] This work was also supported by the Hungarian Scientific Research Fund project No. F-030763.

The aim of the document retrieval system is to search for and gather domain relevant documents from the source environment (commonly the Internet) and make them accessible for further analysis. There are some other applications that provide a similar functionality. These document acquisition systems however use commercial search engines that are typically not effective and precise enough for finding truly relevant documents [3]. The developed application relies on these common engines, yet other methods have been added to provide the expected quality of delivery.

First, the proposed system enhances the document parsing with deeper domain knowledge. Members of the project consortium are working on target domain models. These models contain simple keyword lists, concept networks, document structure samples and rule based knowledge. This knowledge base can help in parsing downloaded documents, deciding whether they are relevant, and retrieving some specific content information from them.

Another improvement is the replacement of a traditional search engine with an agent-based web-robot that models the human behavior of document searching and uses graph-handling algorithms to provide a powerful method for document gathering.

2. MODELING DOCUMENT SEARCH

The problem with the commercial applications is that they use common, keyword based search methods and simple recursive traverse of the web for document acquisition. With the additional use of specific knowledge, a document retrieval system can produce better results.

For realizing the architecture of the proposed application, modeling the human behavior of searching information is required. The activity when a client (human or software agent) tries to find the most relevant documents can be represented as a cyclic process (Fig. 1). Each sub-process is related to some part of the knowledge base that ensures the required intelligent behavior. There are two kinds of knowledge that are important for successful operation. The *domain knowledge* is about the current target domain, about the topics we are interested in. The *search knowledge* represents the client's skills in searching methods. Depends upon tasks, the knowledge about the domain could change occasionally, but the searching skills are usually fixed. Both can be improved with suitable forms of learning.

The first step of the operation is to create the starting set of document addresses (URL-s) and put them into a list, where all the known addresses are stored with some additional attributes (e.g.: visited already or relevant). When the list has been initialized, the cycle starts with selecting one address and downloading the addressed document. The selection is based on some global strategy, determined by the *search knowledge* (e.g.: „at first, it would be useful to start with a commercial search engine query"). The next step is to analyze the text of the document and decide whether it is relevant or not. If it is, it will be saved for further (and deeper) analysis.

The third step is to extract some useful information from the content of the text that can increase the success of the further search or parsing. For example, extract the links that seem to be good for traverse, or learn new domain keywords that can be used for more precise formulation of the query, etc. Then, the extracted information is used to refresh the knowledge base, and the cycle can continue with selecting the next address. The whole operation will be concluded, when the client finds the required information or when all possibilities have been tried.

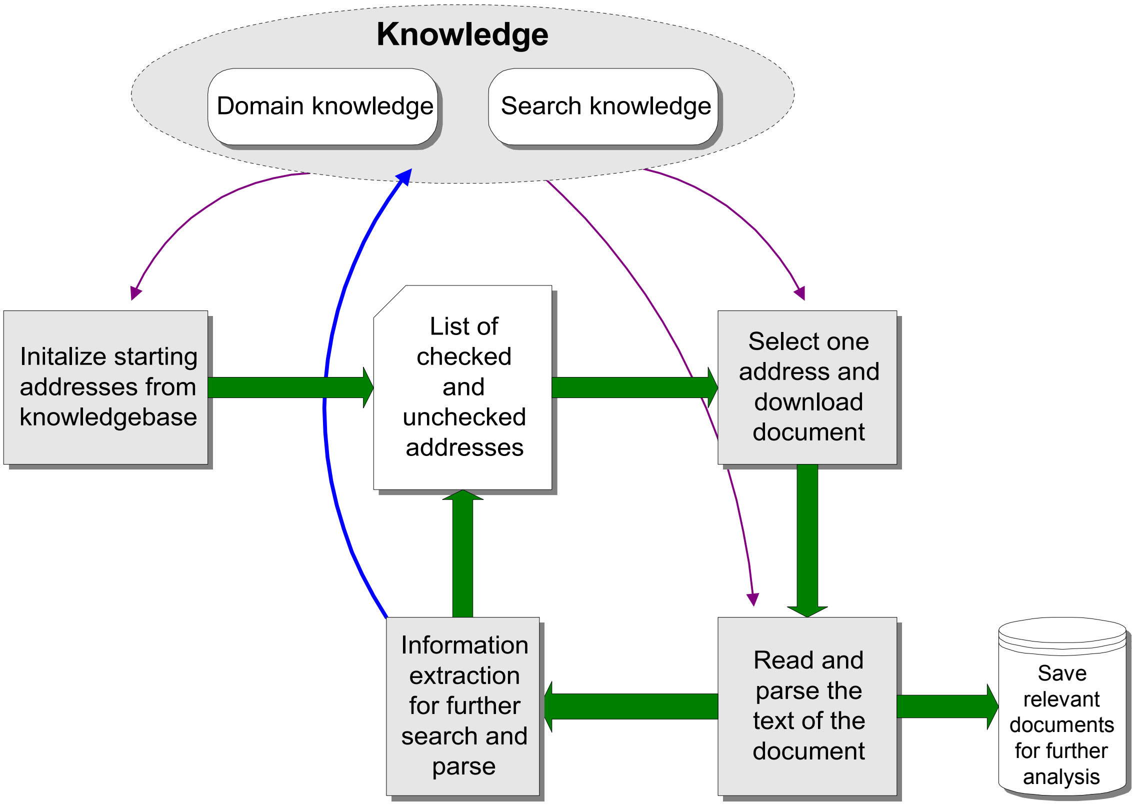
Knowledge
Domain knowledge
Search knowledge
Initalize starting addresses from knowledgebase
List of checked and unchecked addresses
Select one address and download document
Information extraction for further search and parse
Read and parse the text of the document
Save relevant documents for further analysis

Figure 1: General document search model

The effectiveness of the detailed search mostly depends on the knowledge base that consists of the *domain* and the *search knowledge*, as previously mentioned. Another approach (for the same distinction in knowledge) is to examine the agent's type of acting. The environment of the agent can be described like an enormous graph consisting of addressable documents (mostly in HTML format) with links between them. In this environment the agent can act in two ways: pacing from one document to another following the links, or parsing a document at the current point, trying to retrieve useful information.

The *search knowledge* related to the traverse acting of the agent and contains graph based intelligent search algorithms for selecting the most promising links to follow. It also involves additional knowledge about special search methods (e.g.: the use of commercial search engines, etc.).

The *domain knowledge* is related to the analysis of the downloaded documents. Its purpose is to recognize specific characteristics about the incoming documents and try to determine relevance measures and extract information that helps further searching.

3. ARCHITECTURE OF THE WEB ROBOT

Software applications that traverse the Internet and collect information about web pages are called web robots. An intelligent web robot is the implementation of the proposed software agent, which task is to find relevant documents in a specific domain [4]. An agent is an intelligent, autonomous and persistent computer program. It can act and plan to achieve its specific goal. It can take decisions and can communicate with other agents. In this paper we concentrate only on the architecture and knowledge base aspects, which are relevant to the domain-based document retrieval. Below, Fig. 2 presents the architecture of the agent. Its structure and working mechanism are based on the previously detailed model (Fig. 1). The tasks of the main modules are the following:

The *URL register* builds an internal model from the known source environment. With this, the agent not only just locally senses its surroundings, but it has a general view of all the web places it was before. This component is very useful for creating and implementing effective graph based intelligent searching methods. In the simplest case, the register can consist of queues, but a more promising solution is an exact connectionist image of the web, representing all the important retrieved information at the proper place (document). The register is initialized and refreshed with the help of the domain knowledge base and the selection of the next URL is controlled by the search knowledge.

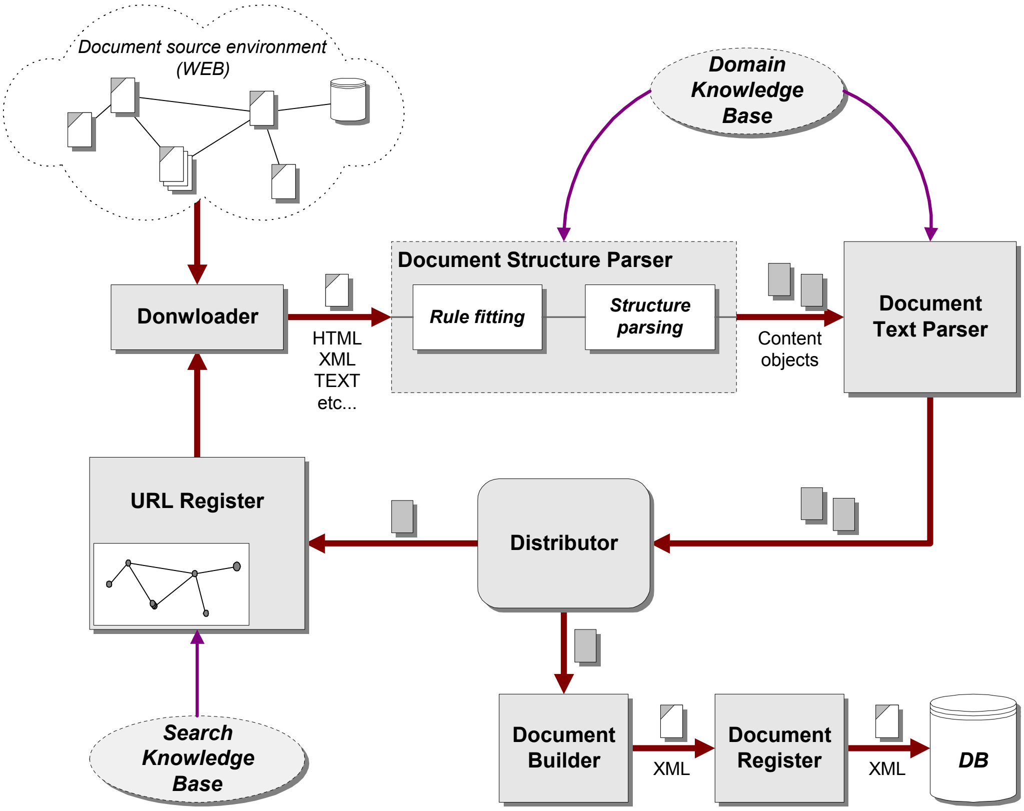

Figure 2: Architecture of the web robot

The task of the *Downloader* module is to download the selected document from the source environment. The application can only handle text resources, so we expect documents of HTML, XML, PDF, plaint text, etc. formats. These are forwarded to the parsers.

The first parser module is the *Document Structure Parser* that is responsible for recognizing the received document and for assigning the proper structure parsing rules to extract the most information for further analysis. It is performed with the help of the *domain knowledge base*.

There are three ways a document can be characterized or recognized: by its URL, its structure and the textual content. All three can be the basis of the recognition process and also of the information extraction process. These two processes are performed by similar parsing operations, except that the recognition uses additional sample fitting methods for matching the extracted information to patterns.

Parsing the structure of a document means to mark some of its typical parts that have a specific meaning. For example an XML document already has been tagged by the content of the text, so it is easy to find its requested parts. However, the structure of plain HTML files describes only the appearance of the document, it does not contain any information about the real meaning. To solve this problem, an automatic tagging tool has been developed that is based on fitting regular expressions to find the start and stop tags of a part of the text. It can be configured with a specific XML based language.

After recognizing the document and finding the right parsing rule, the document can be structured to fit one or more *content objects*. These objects contain the original text source (also with the original HTML or XML tags), but it has been extended with extra tags, that describe content specific information. Each content object has a *content model type* specifying a fixed structure (using a document type definition – DTD) and indicating the meaning of the content. For example a common and simple content type can be the list of links present in a document.

After transforming the downloaded documents into content objects, the *Document Text Parser* module performs textual analysis. It is here, where the system decides about the relevance and assigns that information to the content objects. The operation of the text parser is based on indexing and information retrieval methods [5,6]. It is based on the statistical evaluation of the words in the text. The *domain knowledge base* contains keyword lists and concept networks, which are used for this process. With these, the topic of the document can be recognized.

The *Distributor* receives already the well characterized content objects. Its task is to forward the appropriate type of content object to the right place. For example, a type that contains links should be sent to the URL register for building the internal web model. Other types about the relevant textual content will be forwarded to the *Document Builder*. In the figure above (Fig. 2), only two modules connect to the *Distributor*, but any number can be added, if required for further tasks.

The *Document Builder* module is responsible for assembling complete document objects by filling specific XML structures that represent them for other components of the system. It can join documents that consist of more than one web page. Then the *Document Register* assigns a unique ID to each document and registers them to the system database. Thus, other system components can access them for further analysis.

The *domain knowledge base* can be enhanced with methods that improve the effectiveness and precision of the document retrieval process:

- Information about URL-s with specific structure parsing rules
- Typical document structures with proper parsing rules
- Keywords that can be used to formulate queries for commercial search engines
- Domain keyword lists that can be used to decide whether a document is relevant to the domain
- Concept networks that can improve text indexing and content description

4. CONCLUSION

In this paper, a domain based document retrieval (agent) system was presented. Its architecture and working mechanism are based on modeling the human behavior of document search. Its effectiveness is increased using knowledge base.

As indicated also in other papers [7], the deeper knowledge of the domain lead to promising increase of the effectiveness and precision in document search. In the proposed system, knowledge of search methods was also implied to improve the web traversal of the document retrieval agent. With these, a powerful and useable retrieval application can be developed.

The key factor of the domain knowledge is how documents can be characterized. In the present research, not only standard statistical methods were used, but documents have been analyzed also by their addresses (URL) and their structures. For structure parsing an XML based auto-tagging tool was developed that is based on fitting regular expressions. Applying it to document analysis helps to extract more specific information.

The purpose of the complete development of the searching agent is to create a powerful and easily usable generic system for many kind of applications that demand domain specific information gathered from the Internet or other sources.

5. REFERENCES

[1] Tadeusz P Dobrowiecki, György Strausz and Tamás Mészáros, *„Knowledge Fusion for Financial Advisory Systems"*, The 7th Biennial Conference on Electronics and Microsystem Technology, Baltic Electronic Conference, BEC 2000, October 8-11, 2000, Talinn, Estonia

[2] Tamás Mészaros, Zsolt Barczikay, Ferenc Bodon, Tadeusz P. Dobrowiecki and György Strausz, *„Building an Information and Knowledge Fusion System"*, IEA/AIE-2001 The Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence and Expert Systems, June 4-7, 2001, Budapest, Hungary

[3] Venkat N. Gudivada, Vijay V. Raghavan, William I. Grosky and Rajesh Kasanagottu, *„Information Retrieval on the World Wide Web"*, IEEE Internet Computing, Sept-Oct, 1997, pages 58-68

[4] Stuart J. Russell and Peter Norvig, *"Artificial Intelligence. A Modern Approach"*, Prentice Hall Inc., a Pearson Education Company, 1997

[5] C.J. van Rijsbergen, *„Information Retrieval"*, Butterworth, 1979

[6] Christos Faloutsos and Douglas Oard, *„A Survey of Information Retrieval and Filtering Methods "*, Technical Report on University of Maryland, Department of Computer Science, August, 1995

[7] Andrew McCallum, Kamal Nigam, Jason Rennie and Kristie Seymore, *„Building Domain-Specific Search Engines with Machine Learning Techniques "*, AAAI-99 Spring Symposium, 1999