Acta Cybernetica **18** (2007) 279–291.

Topic and language specific internet search engine*

Domonkos Tikk[†], György Biró[‡], Ferenc P. Szidarovszky[†§], Zsolt T. Kardkovács[†] and Gábor Lemák[¶]

Abstract

In this paper we present the result of our project that aims to build a categorization-based topic-oriented Internet search engine. Particularly, we focus on the economic related electronic materials available on the Internet in Hungarian. We present our search service that harvests, stores and makes searchable the publicly available contents of the subject domain. The paper describes the search facilities and the structure of the implemented system with special emphasis on intelligent search algorithms and document processing methods.

Keywords: Internet search engine, Web crawlers, Document processing, Text categorization

1 Introduction

In the past 5 years the percentage of home internet users increased from 6% to 30%, and there are about 1.5 million people today who use the internet at least 1 hour a month. The increasing number of internet surfers brought on the expansion of content provision as well, but in addition to the traditional business-based content provision — thanks to the technical development and support — user driven content providing plays an increasingly important role. This expansion generated an increasing demand for search services, the development of which was also stimulated by the dynamic increase of the domestic online advertising market. Furthermore the companies and the academic domain realized the scientific challenge in internet searching, and from the year 2000 research workshops — mainly financed by NKFP

*This work was supported by Program GVOP, Grant no. GVOP-3.1.1.-2004-05-0130/3.0. Domonkos Tikk was partly supported by the János Bolyai Research Scholarship of the Hungarian Academy of Science. Zsolt Kardkovács was partially supported by Mobile Innovation Centre.

[†]Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, H-1117 Budapest, Magyar Tudósok krt. 2., Hungary E-mail: {tikk,szidarovszky,kardkovacs}@tmit.bme.hu

[‡]Textminer Ltd., H-1029 Budapest, Gyulai P. u. 37., Hungary E-mail: gbiro@gmail.com

[§]Szidarovszky Ltd., H-1392 Budapest, P.O. Box 283., Hungary E-mail: ferenc.szidarovszky@szidarovszky.com

[¶]GKI Economic Research Co., H-1092 Ráday u. 42–44., Hungary E-mail: lemakg@gki.hu

and IKTA programs — started to develop new search algorithms and -intelligences. In the past 6 years both the academic and business domains attempted to develop new search methods that tried to improve search speed and efficiency through automatic text processing. In most cases, the utilization of these initiatives didn't take place, which motivated us to establish an R&D project that also targets the commercial utilization.

The goal of our project is to create a topic-oriented internet search service prototype that applies semantic-based technologies and novel visualization methods. We selected the topic of economy as the focus of the prototype. The service enables the user to search the largest available economic content collection, and also allows to view and download the documents if the consent of the content provider or digital right holder is given.

With the topic of economy we focused on a compact but thriving segment of the market of internet search services with a diverse user group. The economic contents in broader sense may equally interest the average users (small individual investors, lay users), corporate leaders, business consultants, and decision-makers, as well as users from academia — lecturers, researchers, students.

The project intends to provide such a search service that satisfies users information need more accurately as current state-of-the-art general internet search applications, and with a new way of visualization that may shorten the time of the search. The know-how created in the project offers the opportunity to set-up similar topic oriented search services for other thematic areas and languages as well.

This paper is organized as follows. First, Section 2 describes the designed search functionalities of the system, then Section 3 presents the structure and main components of the system. Section 4 investigates some important features of the system, including the document processing flow, and the web harvesting module. Section 5 presents the results of related projects. Finally, Section 6 concludes the paper.

2 Supported search functionalities

When defining the functionalities of the system, our goal was to provide to the users:

- advanced search possibility,
- enhanced support for browsing the search results visually,
- adaptive search refinement option.

One possible way to improve search efficiency is to enable the user to define the topic of the search. This option helps the search service to capture the meaning of the user query and to get a more accurate idea about their information need — e.g. when processing polysemous queries. This may also decrease the result set's size and simultaneously the number of irrelevant results. The topic-oriented navigation

and search result browsing can be achieved by thematic organization of the searchable content. The availability of a hierarchical category system with appropriate coverage (topic taxonomy) of the broad topic is a prerequisite for this purpose. The general worldwide search services also provide such a search option (see e.g. Google Directory, Yahoo Directory, the former Zeal search of the Looksmart group[1]), but the significance of this search option is much larger when it concerns a search service on a unilingual and thematically limited document collection. In the case of general search engines, it is much more difficult to create and maintain a fully detailed topic taxonomy with required coverage, and it is also a challenging task to sufficiently fill up the taxonomy with quality content. The focused search topic our project alleviates the difficulties that arise in the maintenance of a dynamically changing topic taxonomy of diverse contents.

Eventually, the user may search for similar documents starting from a sample one — possibly created by him- or herself. General search engines do not support queries longer than a certain limit, therefore if the sample document is not indexed by the search engine — that is quite likely in the case of an own document — these engines are unable to properly execute this task.

The results of a former search can also be a starting point for the search refinement. But, it is far from trivial for the average user — even after having gone through some elements of the search list — how the query should be effectively expanded or modified. This activity can be efficiently supported by offering candidates keywords taken from the search result set.

Based on the previous considerations, we set the following search functionalities:

- keyword based search with taxonomy support,
- similar document search based on user's sample,
- browsing in fixed topic taxonomy,
- query refinement based on keywords from the search result set.

3 The structure of the system

The system of the search service comprises four main components that are depicted on Figure 1. The main components are

- web crawler,
- natural language processing module
- indexer and categorizer module
- graphical user interface.

Next we describe the task and operation of each component.

[1] `http://www.google.com/dirhp`, `http://search.yahoo.com/dir`, `http://www.zeal.com`

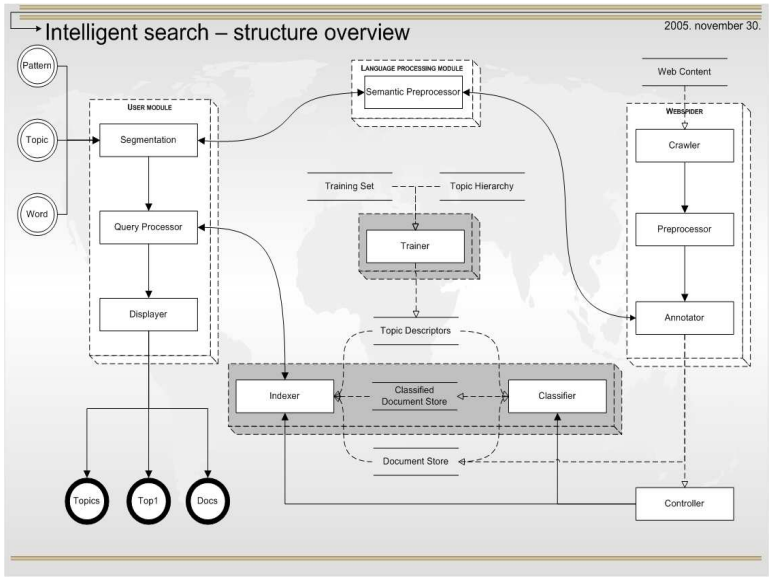

Figure 1: Structure overview of the system

3.1 Web crawler

The web crawler harvests the contents of selected topic-related web pages, and converts them to the prescribed XML format used by the system. Currently, the crawler collects contents from about 50 sources, which are mostly national economic content providers, but also includes relevant pages from some leading portals and regional news providers too.

3.2 Natural language processing module

This module converts the documents with different formats coming from different sources. The document processing flow has been defined to be able to perform different text representation techniques (such as word or character n-gram based), and to integrate various external natural language processing tools.

3.3 Indexer and categorizer module

The indexer engine creates and maintains the index catalog that serves as the basis of answering queries. The categorizer engine administrates and determines the category information to support the taxonomy based search of the system. The categorizer component performs supervised machine learning, i.e. it learns the significant words/expressions of each category of the taxonomy by means of sample training documents. This feature is implemented by integrating HITEC hierarchical text categorizer engine [5, 9]. On the one hand, HITEC's engine is

applied to create category profiles consisting of weighted words/expressions, and on the other hand, HITEC predicts category labels of unlabelled documents being harvested from the web or coming from other sources. On Figure 1 we illustrated by a gray background that these were externally developed modules of the system.

3.4 Graphical user interface (GUI)

The GUI provides the search functionalities to the users, forwards queries towards the search engine and displays and visualizes the results.

3.5 Creation of the taxonomy and training environment

In order to guarantee the efficiency of the internet search service and the quality of the search, it is essential to have a taxonomy that simultaneously represents the selected topic with appropriate details, and vaguely covers topics weakly related to the main theme. We used the subject hierarchy of the Central Library of the Corvinus University Budapest (CUB) as a starting basis of the economic taxonomy used in our system. The graph of the original subject taxonomy included different relation types between subject headings (broader/narrower, used/unused term, related concept), and included undirected cycles. Due to these inconveniences, it couldn't be applied directly for our task that requires an acyclic directed taxonomy based on broader/narrower term relations. The modification of the taxonomy was done by the librarians of CUB in cooperation and with guidance of our stuff. As a result, we got a taxonomy that includes all the subject-headings of the library's original subject taxonomy as categories. During the build-up of the taxonomy some new, connecting taxonomy nodes have also been created. The final taxonomy has 2397 nodes starting from 16 top level categories. Based on the results of preliminary tests, this structure has been further modified by merging rare categories and hence decreasing their number to 1383.

To complete the training environment of the system we needed sample training documents that sufficiently represent the categories of the taxonomy. For this task, we obviously used the subject taxonomy of CUB, since doing so we gained numerous training samples, particularly those electronically available documents that were indexed and stored by the Central Library of CUB. This initial document set has been expanded first by acquiring the electronic versions of already annotated documents, and second, by annotating other electronic documents of the topic.

4 Operation of the system

4.1 Document processing flow

We differentiate two document types in the system: training and harvested documents. The only difference is the present/absent of the category label: training documents have category labels, while harvested ones don't. The user queries are handled analogously as harvested documents in the processing flow except that they

are not stored. The original format of documents can vary. The system processes HTML, PDF, DOC, RTF and plain text files, and converts them to the internal XML representation format. The internal XML format is capable of representing documents at any processing stage of the document processing flow, and though being particularly optimized for text mining tasks, it also can be easily converted to any standard XML document scheme (such as, e.g., NewsML, TEI). At the creation of the DTD the main points were that

1. it should be able to code the required textual and meta-information;
2. the storage capacity to store the XML format of documents should be minimal.

This latter point is crucial in both keeping the storage need of the document corpus as low as possible, and reducing time and memory requirements of the document processing algorithms. The first requirement is guaranteed with a relatively flexible structure definition, while the latter one was obtained by short XML element names and by minimizing the set of required attributes. The values of XML elements are designed to store the textual information of the documents, while the additional meta information (e.g. grammatical features) is stored in attribute values.

The work flow of document processing is presented on Figure 2. One can observe that the document processing flow is identical for each document type (training, harvested, query). After XML conversion the module *Merger/Splitter* unifies the character encodings of the documents. The *Text Extractor* component employs various natural language processing tools, such as:

- *Stemmer*: The system offers two alternatives for this task. First, it includes an implementation of timid stemmer algorithm [4], second, it can employ the stemmer of the *HunMorph* package [10]. The XML format is able to store different grammatical parsing alternatives of a given word (see *g[rammar]* element), such as e.g. various word stems. This information is stored in the stem attribute of g.
- *POS tagger*: This approach also exploits a function of the HunMorph package. The usual implementation of an index catalog is word stem based. This solution merges homonym words. In order to alleviate this deficiency the system stores *[stem, pos-tag]* information in the index catalog. The part-of-speech of the words are stored in the *pos* attribute of *g* element.
- *Word filter*: This component is necessary for the query refinement. When offering keywords for this search feature, the function or stop-words should be disregarded. The filter works based on a stop-word list and patterns. The process sets the *sw* attribute of the filtered word's *g* element to true. These words are used in the index catalog; therefore they cannot be eliminated from the text.
- *Sentence segmenter*: This module segments the text into sentences. Its output is used when displaying the most relevant context of a document. It is a rule-based module: when finding a candidate sentence separator, the matching

rules determine whether it is a valid sentence separator or not. The rules are assigned signed weight. If the analyzed text fragment of the sentence separator matches several rules the final decision is taken by the aggregation of matching rules. An abbreviation dictionary is also used at the process. The detected sentences are labeled with *s[entence]* element.

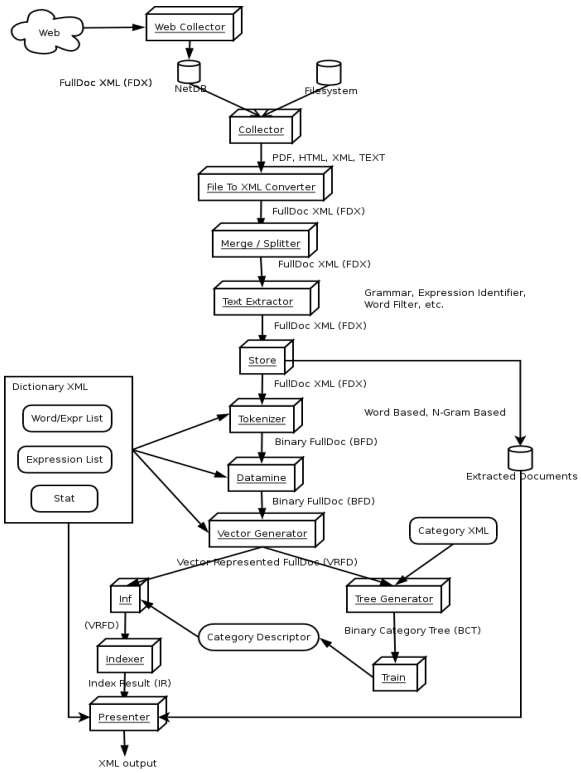

Figure 2: The document processing flow of the system

One can easily code the results of arbitrary natural language processing tools with the internal XML format, e.g. the output of a morphological parser, or named entity recognizer [8]. We will investigate the effectiveness of the integration of such external tools in terms of improving search efficiency.

Each document is stored in three versions coming from different processing stages. In addition to the original format, we save the raw XML file, and final fully processed XML format for each document. This is performed by the *Store* module, which also assigns the category information to the document, if available. The URLs of the different versions are specified in the appropriate attribute of the *document* element, while the category information is encoded into the *mc* (main category) and *sc* (secondary category) fields.

After this stage documents are converted into numerical representation, which is done by the *Tokenizer* method. The *Datamine* module searches frequent word sequences in the tokenized version, and creates new tokens from them. Finally the inner representation form is generated by the Vector Generator module. This creates two vectors: one for indexing — this also includes stop words; and one for categorization and keyword-determination — where stop-words are excluded. The former one applies TF-IDF weighting scheme, while the latter one uses entropy-weighting [3].

Finally, training documents are applied to train the classifier (*Train*), the category information of other documents are inducted by means of the classifier model (*Inf*). The documents are indexed in the next step. The *Presenter* module displays the required category and keyword information towards the user interface, and attaches the matching document for the query. Here the tokenized versions are converted back to text format.

4.2 Web harvesting

The task of the web crawler is to keep track of, archive and annotate the contents of selected web pages. The harvesting has two main functions:

1. Monitoring of selected web-sites, downloading new information (briefly: harvesting).

2. Conversion and structural annotation of downloaded documents.

The selected web pages have to be visited regularly. The harvesting is performed by a so-called daemon — termed as *Crawler* — that has four simple functions: start, harvest, stop and delay.

The *Crawler* starts the download process, where the downloaded content is typically a pre-specified, regularly visited URL — the main page of a portal, or an RSS-channel[2]. Having downloaded the raw content of the visited page, the Crawler analyzes and annotates its content, selects the documents to be retrieved, saves them to the document archive, and finally preprocess them before the next step of the document processing is started.

The specification of the *Crawler* does not contain topic specific details, therefore it is applicable for any topic domain with proper parameterization. Having said this, one should observe that there is no uniform algorithm to separate the relevant and irrelevant parts of the document. The automatic selection of the coherent and connected segments of a document with 100% accuracy cannot be guaranteed even with deep semantic analysis of the text. (On accuracy we mean that the selected segment includes all topic-related material from the downloaded text, and only that.)

However, the relevant text can be identified with about 90% accuracy by means of the analysis of some key factors, such as, the displayed and real title of the

[2]Real Simple Syndication; `http://blogs.law.harvard.edu/tech/rss`

document, and the mutual relation of the following 5 fields: date, author, title, abstract, text body.

In practice, the annotation of different portal sites is performed by means of a limited set of auxiliary software, called plug-ins. These programs convert the input HTML text into the required XML format defined by *fulldoc* schema using the structural characteristics of the harvested site.

Since the characteristics of harvested sites can change in time, the *Crawler* has to be prepared for such structural changes that are originated from, e.g., the modification of the portal engine, or the revision of the web site. In such cases, the harvested documents are likely to become invalid. Therefore all XML output of the *Crawler* is parsed syntactically before archiving. It often happens that the output created by an outdated plug-in misses some relevant fields (e.g. title or text body), and this can be detected by parsing. The structural changes of the harvested portals can thus be detected with high probability. The syntax parsing check might only fail to detect a problem with the input, if the structure of the portal does not change, but the topic of the content is altered. Such modifications cannot be detected automatically in the current version of the *Crawler* without continuously re-retrieving articles from the portal.

4.3 Graphical user interface

The visualization of the search results was an important factor at the design of the system. The result documents are displayed in two alternative ways: the document map and the document list. The document map is a visual display form (see Figure 3), which places the result documents on a circle, where the distance from the origin represents the similarity of the document to the query. Documents of different categories are marked with different colors. The content of a document can be viewed by clicking on the document icon. The document list (see Figure 4.) is a traditional form of displaying the search results. Here, by default, the query refinement tab are also displayed. This tab can also be switched on at the document map view. The taxonomy of the service can be applied to perform category based filtering of the result (also at the initial search). The service will be made publicly available at the beginning of 2008.

5 Similar initiatives

The goal of the *Information & Knowledge Fusion* project (IKTA3-181/2000) was the analysis, design, and establishment of a new intelligent knowledge warehousing environment, capable of efficient information- and knowledge management on specific vertical application domains [2, 1]. The project developed a knowledge-based information retrieval system for the financial sector that is based on various data sources, and generates reports according to the needs of the field of application in a structured format.

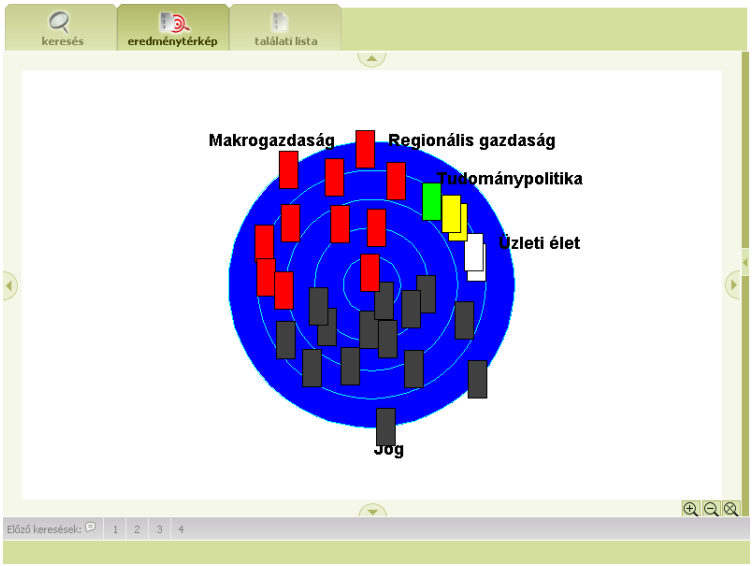

Figure 3: The document map view

The project, named *In The Web of Words* (NKFP 0019/2002) aimed the establishment of a complex internet based search/query application, introducing new search methods in both the deep web search (online accessible contents hidden from traditional search engines, such as databases or not directly linked contents), and image search [6, 7]. Supporting image search, a visual thesaurus has been developed that serves as a text based image-content category system used in characterizing and indexing image contents. In the deep web search, the system allows search queries in the form of natural language sentences in the Hungarian language.

The *Semantically organized lexical network and internet content research* (IKTA5-123/02) project intended to produce an internet content search technology based on a semantically organized and interpreted lexical network. The project tried to reach its goal by researching the possible connections of meaning centers (the basic unit of the lexical network that is a structure of natural language designators — words, phrases, etc. — organized around a common meaning), and by building the appropriate connecting elements. The semantically organized, communicative lexical network — assembled by the linking of meaning centers — is developed in a way that it can efficiently operate in applications based on language-technology (like natural language processing systems, interpreted information-search in electronic texts and structured text bodies, content monitoring, machinery translation, context- and style sensitive spell checker).

The *Knowledge based Hungarian semantic search system* project (IKTA 00148/2002) was led by the National Traumatology and Emergency Institute (NTEI). In addition to the statistical control of data, the project includes de-

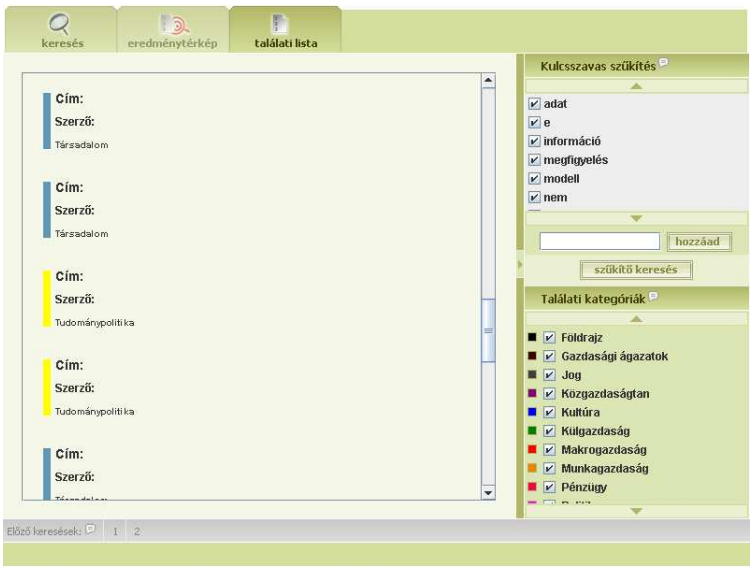

Figure 4: The document list view

termination of extensional relations based on meaning coherency (clustering) and logical connections. The developed technology recommends the use of "knowledge retrieval" by means of machine learning and processes based on neural network, together with classical data mining methods (drilling-up, drilling-down etc.). The test of the system executed at the NTEI, because the necessary medical ontology was available there, and at them it is vitally important to find the relevant document as fast as possible. The developed knowledge based search technology can be used widely as a search engine for libraries, archives, medical-, legal- or corporate data- and knowledge-bases or any commercial applications where the goal-oriented searching has an important role.

The *WebKat*[3], developed by Neumann-ház in 2002 in the framework of a national R&D project, is a topic map based solution for searching Hungarian contents. The topic map supports the search by the visualization of its structure with internal relations. This search engine retrieves documents from a dedicated a separate database, so the search is not performed directly on the Internet.

Polymeta[4] is a general purpose meta-search engine, that enables the user to search the Internet by using simultaneously several selected search engines. An aggregated result page is generated from the various result sets. On this page, single hits are displayed in order of importance. A "table of contents" is also created, where hits are clustered and displayed as directories. This allows the user to capture the various meanings and associations related to the query and the

[3] `http://www.webkat.hu`
[4] `http://polymeta.hu/`

results.

The new initiative called *Vipkereső* cannot be reached in full functionality by this time, but according to recent news, it will be a full text web search engine, offering image-, blog- and news search options as well.

6 Summary

This paper presents the functions and structure of a topic oriented semantic-based Internet search engine developed in the framework of a Hungarian R&D project. The prototype of the system performs intelligent search on Hungarian economic related content. The paper describes in detail the main components of the systems, the document processing flow, the document harvesting solutions, and also describes the graphical user interface.

References

[1] Cs. Dezsényi et. al. Tudásalap információk kinyerése: az IKF projekt. *Tudományos és Műszaki Tájékoztatás*, 2004. `http://www.neumann-haz.hu/tei/publikaciok/2004/biro_ref_ikf_hu.html`.

[2] Dezsényi, Cs., Varga, P., Mészáros, T., Stratusz, Gy., and Dobrowiecki, T. Ontológia-alapú tudástárház rendszerek. `http://nws.iif.hu/ncd2003/docs/ahu/AHU-118.htm`.

[3] Salton, G. and Buckley, C. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1998.

[4] Tikk, D., Biró, A. Törcsvári Gy., and Bánsághi, Z. Szótövező eljárások hatása magyar szövegek automatikus kategorizálásánál. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 430–434, Szeged, Hungary, 2005.

[5] Tikk, D., Biró, Gy., and Yang, J. D. Experiments with a hierarchical text categorization method on WIPO patent collections. In Attok-Okine, N. O. and Ayyub, B. M., editors, *Applied Research in Uncertainty Modelling and Analysis*, number 20 in Int. Series in Intelligent Technologies, pages 283–302. Springer, 2005.

[6] Tikk, D., Kardkovács, Zs. T., Andriska, Z., Magyar, G., Babarczy, A., and Szakadát, I. Natural language question processing for hungarian deep web searcher. In Elmenreich, W., Haidinger, W., and Tenreiro Machado, J. A., editors, *Proc. of the IEEE Int. Conf. on Computational Cybernetics (ICCC'04)*, pages 303–308, Vienna, Austria, 2004.

[7] Tikk, D., Kardkovács, Zs. T., and Magyar, G. Searching the deep web: the WOW project. In *Proc. of the 15th Int. Conf. on Intelligent Systems Development (ISD'06)*, Budapest, Hungary, 2006. To appear.

[8] Tikk, D., Szidarovszky, F. P., Kardkovács, Zs. T., and Magyar, G. Magyar nyelvű kérdő mondat elemző szoftver. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 455–459, Szeged, Hungary, 2005.

[9] Tikk, D., Yang, J. D., and Bang, S. L. Hierarchical text categorization using fuzzy relational thesaurus. *Kybernetika*, 39(5):583–600, 2003.

[10] Trón, V., Halácsy, P., Rebrus, P., Rung, A., Simon, E., and Vajda, P. Morphdb.hu: magyar morfológiai nyelvtan és szótári adatbázis. In Alexin, Z. and Csendes, D., editors, *III. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY'05)*, pages 169–179, Szeged, Hungary, 2005.