AUTOMATICALLY ACQUIRED DOMAIN KNOWLEDGE FOR AD HOC SEARCH: EVALUATION RESULTS

Udo Kruschwitz
Department of Computer Science, University of Essex
Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom
Tel.: +44 1206 872669, Email: udo@essex.ac.uk

ABSTRACT

The automatic acquisition of usable domain knowledge is a challenging issue. Such knowledge can be employed to assist a user in searching a document collection. This can be done by suggesting query modification options based on the knowledge uncovered by analyzing the document collection. We acquire such knowledge by simply exploiting the documents' markup structure. This gives us a domain model tailored to the particular collection. But how good is such a model? This paper will present results of two evaluations. The first one looks at the actual domain model. We will discuss how users judged the relations encoded in the model. The second evaluation is task-based and goes a step further. It investigates how a search system that applies the automatically constructed domain model performs compared to a standard search system.

Keywords: Knowledge extraction, Evaluation, Web search

1 INTRODUCTION

Searches in document collections often return either large numbers of matches or no suitable matches at all. That is not just true for Web search in general but also if the collection is only a fraction the size of the Web and the documents cover a much smaller range of topics. This type of data sources is everywhere, from corporate intranets to local Web sites.

Some explicit knowledge about the domain, i.e. a *domain model*, could be useful to help the user find the right documents. A possible domain model could encode relations between words or phrases that have been uncovered by analysing the document collection. The simple graph in figure 1 is an actual example from one of our sample domains that gives an idea of how such a model may be structured. We see a simple tree of related terms that can be used to either assist a user in the search process, perform some automatic query refinements or allow the user to browse the collection. Note, that the *types* of relation in the sample tree are not formally specified. This is significantly different from formal ontologies or linguistic knowledge sources such as *WordNet* [4].

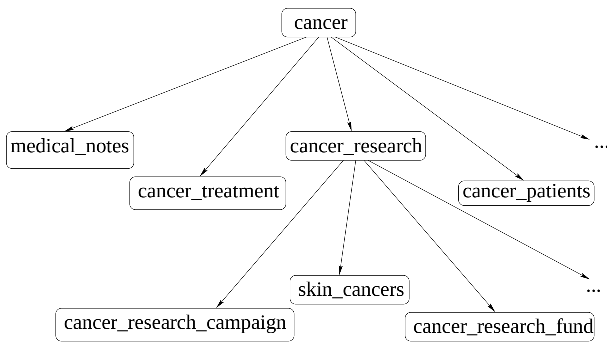


Figure 1: Partial domain model

However, typically a domain model is not available. Existing linguistic knowledge sources and ontologies tend to be either too generic (e.g. *WordNet*) or very knowledge-intensive and tailored specifically to some well-defined domains (e.g. [1, 5]). In other words, they do not reflect the particularities of a newly selected document collection and are therefore not the best candidates for assisting users in a search process. Furthermore, changes in the collection or an entirely new domain would require changes in the ontology, which can be a complex task [10].

What can be done about that? A tremendous amount of implicit knowledge is stored in the markup of documents. But not much has been done to use this particular knowledge. We construct a domain model automatically by exploiting the markup of documents. Detailed information on this process as well as links to related work can be found in [8] and [9]. We will however briefly review this process to give the reader enough background information.

Having built the model it can now easily be incorporated into our generic framework which is an online search system that has access to a standard search engine and the domain model [9]. This search system is a specialized dialogue system that offers and makes choices about search through the set of documents based on the domain model. This dialogue system is unlike *single shot* systems such as standard search or question answering in that it interacts with the user by offering options to refine or relax the query. With a new document collection coming along we just have to press a button to create the appropriate domain model. The rest of the framework can be left unchanged. However, there are two questions that need to be answered:

1. *How good is the automatically constructed domain model?*
2. *How does the outlined search system perform against alternative approaches?*

We will discuss both questions in this paper. But before we can do that we will present an overview of how the domain knowledge is extracted and organized in a model (section 2). We will then look at an evaluation that focuses on the suitability of our automatically constructed domain model for ad hoc search (section 3). Our ultimate goal is to use the potential that some assistance in the search process offers by incorporating our domain model in the search system. Section 4 compares such a system with a baseline approach to find out how much of this aim we can achieve.

2 MODEL CONSTRUCTION

This section presents a short summary of the domain model construction process. This process starts by automatically selecting a small number of terms (nouns and noun phrases) from the document collection. We then organize these terms (which will be called *concepts*) as a set of simple hierarchies which will form the domain model.

In documents marked up in HTML we can identify a number of frequently used markup contexts such as document *titles*, *bold* text, certain text found in *meta* tags, *underlined* text etc. If the documents were articles in a newspaper archive, we might distinguish markup contexts like article *headings*, *captions* of pictures, *summaries* etc. We then define:

Definition 1 *An index term c is a concept term (or simply concept) of type n for document d, if c occurs in at least n different markup contexts.*

We argue, that this abstraction separates important terms from less important ones in a very efficient and domain-independent way (see [9] for a detailed discussion and supporting evidence that this is indeed the case).

The identification of conceptual terms is the first step towards the construction of a domain model. The next step is to arrange the extracted terms in some simple usable structures.

One of the main motivations for automatically acquiring a domain model is the inadequacy of a domain-independent knowledge source like *WordNet* in search applications for limited domains. Having said that, the *structure* of a knowledge source like *WordNet* looks very promising for ad hoc search assistance. The reasons include the clear and simple organization, its applicability, and the advantage that the model does not have to be rebuilt every time the document collection needs to be re-indexed. The model is independent of the actual documents.

The world model we construct is a set of simple concept hierarchies. The interpretation of these hierarchies is very different from a number of other models. In *WordNet* for example the links between two sets of terms are clearly defined semantic relations (e.g. *hypernymy*, *antonymy* etc.). Our aim is not to capture the actual *semantic* relations that exist between concepts but the fact *that* there is some relation, one that can be used to guide a user in a search process. Two examples of *structurally* similar models are described in [11] and [2]. The construction process is however different to the work presented here.

The construction of our world model is straightforward and is performed fully automatically in an offline process. The process is based on a simple relation:

Definition 2 *Two concepts c_1 and c_2 are related concepts of type n for document d, if c_1 and c_2 are concepts of type n for document d.*

Each of the hierarchies in our model consists of nodes represented by a concept (see for example figure 1). We have found that the concepts that have been identified in the indexing process are likely to be among those terms that users submit as real queries when they search the document collection [9]. Based on this finding the model construction process is a sequence of simulations of user requests that does not use live user queries but the concepts that have been identified in the documents. Each concept is a potential query. Therefore, we construct a hierarchy for every single concept, in other words every concept represents the root node of one hierarchy. We start with a single concept as a possible user query (e.g. *cancer* in figure 1). Utilizing the *related concepts* detected in the source data one can then explore all possible ways of constraining this query by adding a single concept to the query in a query refinement step. The interesting terms which could be added to the query in such a step are all the concepts that were found to be *related* to the original query (concept) term. A new daughter node is created if there are documents in the collection that match the refined query. In the example, we create a daughter node *medical_notes* because *medical_notes* and *cancer* are related concepts and the query that is a conjunction of both these terms returns a non-empty document set.

The model construction is an iterative process that can be applied to the new queries until eventually one ends up in leaf nodes, i.e. nodes that typically represent very specific queries for which only small sets of documents can be found. In each of those iterative steps one would expand the current query by a single concept that is related to *all* query terms collected so far (figure 1 shows that there are documents in the collection that match the query *"cancer AND cancer research AND skin cancers"*).

By limiting the number of branches originating in a node one gets a usable model in which each of the daughter nodes can be interpreted as query refinements of the query represented by the mother node. Weights are associated with each arc indicating the number of matches that each node represents (not displayed in the example).

The resulting hierarchies are applied in a simple dialogue system to assist a user in ad hoc search tasks. Details about how the model is used exactly can be found in [9].

3 EVALUATING THE MODEL

A user study has been conducted to find out whether the relations between term pairs uncovered in the domain model construction process are indeed sensible relations. For this study we adopted an approach used in [11]. That study investigated how "interesting" users would find pairs of words automatically extracted in a process that constructs term hierarchies. Our study aimed at finding out whether such a relation could be "relevant" in an ad hoc search system or not.

Users were asked to judge for term pairs whether they found them relevant, not relevant or whether they did not know. They were told that relevant means that assuming they have submitted the first term as a query, they would find the corresponding second term to be a sensible query refinement in the specified domain.

We used two sample domains: the Web site of the University of Essex[1] and the BBC News Web site[2]. For each of these domains users were asked to judge 50 term pairs - 25 of them pairs found in the automatically constructed domain model, the other 25 pairs were pairs selected using a baseline approach. Subjects did not know which technique was used for which term pair. The term pairs were presented in random order.

The selection process for a pair of terms was as follows. Using log files of queries submitted to the existing search engine installed at the Essex Web site, we selected the most frequently submitted queries. For each of these queries we constructed a term pair by first consulting the automatically created domain

[1]`http://www.essex.ac.uk`
[2]`http://news.bbc.co.uk`

model to find the hierarchies whose root nodes contained the query (queries consisting of more than one query term were treated as compounds, trivial hierarchies were ignored). We then selected the arc with the highest weight to find a corresponding term (e.g. *medical_notes* for *cancer* in figure 1).

For each of the terms (i.e. queries) we also constructed a "random pair". For this baseline approach we used Google's API[3] to submit a query specifying the domain, selected the first page of matches Google returned, downloaded these documents and selected the most frequent term found (applying the same indexing steps as we did for building the domain model).

For the BBC News domain we did not have log files. Nevertheless, earlier we indicated a strong overlap between concepts and frequently submitted queries in the Essex domain. Therefore, we decided to select the most frequent concepts found in the BBC News domain. For the construction of related pairs and random pairs we followed exactly the approach used in the Essex domain.

In total 31 subjects were recruited for this experiment, 19 of them members of staff in the Department of Computer Science, 12 of them students from various departments. This is how users judged the term pairs for their relevance:

- BBC News domain: 64% of the potential query refinements using the concept based approach were considered *relevant* (baseline: 48%).

- Essex domain: 59% of the potential query refinements using the concept based approach were considered *relevant* (baseline: 50%).

There is some interesting resemblance with the figures reported in [11]: 67% judged the term pairs in the concept hierarchies to be "interesting". The baseline approach gave 51%. But note, that the two approaches are fundamentally different, one constructs a model in an *offline* process, the other one retrieves documents for a specific query and then constructs term hierarchies *online*.

The differences in judging the term pairs constructed using the two techniques were found to be significant in paired t-tests. For the BBC News data we found a significance with $p < 0.0005$ and for the Essex data $p < 0.003$. Significance can also be shown when analyzing the results obtained for either students or staff members only.

[3] http://www.google.com/apis

4 TASK-BASED EVALUATION

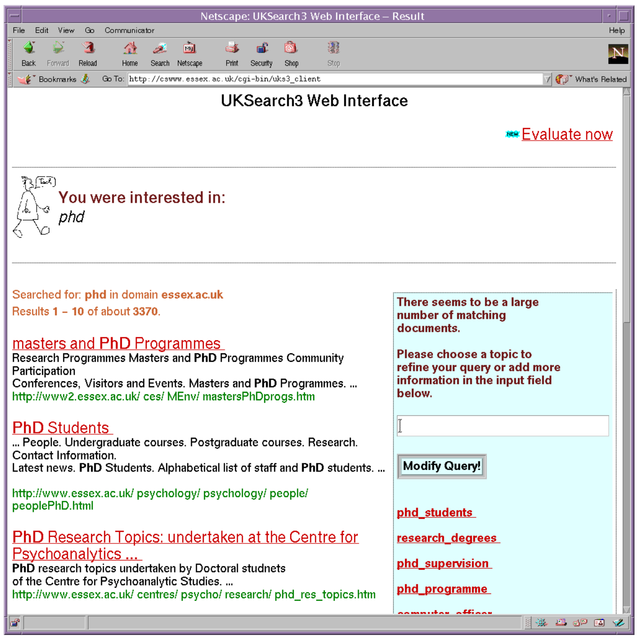
We have just seen that the automatically constructed domain models have the potential of being sensible knowledge sources to assist a user in ad hoc searches. The next step is to actually incorporate such a domain model in a search system. Evaluating such search systems is an important research issue. A problem that arises here is the interpretation of the results. Results that cannot be re-validated or compared against alternative approaches are not very useful. That is why we adopted an existing framework for this evaluation. The TREC[4] conference series has been successful at setting the standards for comparing systems against each other. Since we are interested in comparing two different search systems that work on the same domain, we adopted the evaluation methods developed for the TREC Interactive Track.

We compare *UKSearch* - an instance of our generic search system that incorporates the domain model - and a baseline system. Both systems access the same document collection, the University of Essex Web site. Our backend search engine is *Google* which has a good coverage of the sample domain and gives a very good baseline. The two systems can be characterized as follows:

- *System A* is the baseline system which functions like a standard search engine: the query is submitted by the user, Google (accessed via the Google API) returns the results and the first ten matches are displayed. A user can then either modify the query, go back to start a new search or click through the result set via a link that takes the user to the next 10 matches.

- *System B* is the *UKSearch* system that uses the automatically constructed domain model to assist a user in the search process by offering ways to relax or constrain the query. It also uses

[4] http://trec.nist.gov


Netscape: UKSearch3 Web Interface – Result
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Go To: http://cswww.essex.ac.uk/cgi-bin/uks3_client What's Related

UKSearch3 Web Interface

Evaluate now

You were interested in:
phd

Searched for: **phd** in domain **essex.ac.uk**
Results **1 - 10** of about **3370**.

masters and PhD Programmes
Research Programmes Masters and **PhD** Programmes Community Participation
Conferences, Visitors and Events. Masters and **PhD** Programmes. ...
http://www2.essex.ac.uk/ ces/ MEnv/ mastersPhDprogs.htm

PhD Students
... People. Undergraduate courses. Postgraduate courses. Research. Contact Information.
Latest news. **PhD** Students. Alphabetical list of staff and **PhD** students. ...
http://www.essex.ac.uk/ psychology/ psychology/ people/ peoplePhD.html

PhD Research Topics: undertaken at the Centre for Psychoanalytics ...
PhD research topics undertaken by Doctoral studnets
of the Centre for Psychoanalytic Studies. ...
http://www.essex.ac.uk/ centres/ psycho/ research/ phd_res_topics.htm

There seems to be a large number of matching documents.

Please choose a topic to refine your query or add more information in the input field below.

Modify Query!

phd_students
research_degrees
phd_supervision
phd_programme

Figure 2: System B: a user has typed in *"phd"*

Google's API to display the best matching documents alongside the options that have been found.

Both systems look almost identical, the difference is only that in *System A* no query modification options are constructed. Apart from that the GUI and functionality is exactly the same in both systems. Figure 2 is a screenshot of the actual *UKSearch* system showing the system's response to the user query *"phd"*.

16 subjects and 8 search tasks are needed to perform an evaluation according to the guidelines originally developed for the TREC-9 Interactive Track [7]. We will first discuss the experimental setup in more detail. After that we will discuss the results.

4.1 Search Tasks and Procedure

Apart from adopting the TREC Interactive Track guidelines we also wanted to address one particular limitation of the experimental setup in that track, namely the fact that for the experiments the "conditions are artificial" [6]. Unlike in the TREC Interactive Track we did not want to construct hypothetical search tasks but use the logfiles of queries submitted to the existing search engine at the University of Essex to make the search tasks as realistic as possible. Therefore, we constructed search tasks which:

- are based on *frequently submitted queries*;
- seem realistic (although we cannot be sure about the actual information need a user had by just looking at logged user queries, we can try to construct tasks that could have resulted in the query the task was derived from);
- aim at single documents that contain the answers for the particular tasks;
- are not trivial, in the sense that we tried to prevent queries that return an appropriate answer for the search task among the top 10 ranked documents.

The last point in particular proved very difficult due to the fact that we used Google as the backend search engine.

The following tasks were constructed for this evaluation (the query that the task is based on is shown in brackets and is not part of the actual search task).

- **Search Task 1** (*accommodation*): Your department invites a seminar speaker from Edinburgh. The speaker will need accommodation for one night. Locate a document which contains information about suitable accommodation.
- **Search Task 2** (*password*): You have forgotten the password for your Essex account. Find a document that tells you who to contact.
- **Search Task 3** (*scholarships*): You want to find out what external scholarships are available for postgraduate students from Asia - that is scholarships *other than* those offered by the University or the individual departments. Locate a document that has information about such scholarships for study in the UK.
- **Search Task 4** (*football*): You want to play football using one of the University football pitches. Find a document which tells you what you need to do to book the football pitch.
- **Search Task 5** (three frequent queries: *vacancies*, *jobshop* and *jobs*): You are looking for a

job as a student to work at the University. Locate a document which has a list of jobs currently available. Documents that only list jobs outside the University or jobs not available to students are not relevant.

- **Search Task 6** (*gallery*): Essex University has its own art gallery. Find a document which has information about what is currently being shown at the gallery (or has been shown earlier this year).
- **Search Task 7** (*phd*): The University offers a number of different research degree schemes. Imagine you are interested in doing a PhD at Essex University. Locate a document that informs you about what area you can do your PhD in and how long it typically takes to do a PhD here. Documents that inform you about PhDs in individual departments only are not relevant.
- **Search Task 8** (*student support*): Find a document with contact details about help for students who have problems with their landlord and need support. Documents that indicate help for University accommodation only are not relevant.

The questionnaires used in this study are based on the ones proposed by the TREC-9 Interactive Track (using a 5-point Likert scale where appropriate). We used the following questionnaires: entry questionnaire, post-search questionnaire, post-system questionnaire, and exit questionnaire.

The procedure that every subject had to go through has been adopted from [3]: Subjects started by filling in the entry questionnaire. This was followed by a demonstration of the two systems. The users were informed that they will be using two different search engines, but they were not told anything about the technology behind them, nor about the use of Google as the backend search engine.

Users were then asked to perform four search tasks on one system followed by four tasks on the other one according to a searcher-by-question matrix (see [7]). After each task users were asked to fill in the post-search questionnaire. After completing all four search tasks on one system users were asked to fill in the post-system questionnaire. Finally, after finishing all search tasks users had to fill in the exit questionnaire.

4.2 Results

Due to space restrictions we will concentrate on the most interesting results. In all cases, t-tests have been used for significance testing.

We had 16 volunteers for this experiment, 8 of them male and 8 female. The majority of the test persons (14) were postgraduate students (Master or PhD students) with various backgrounds, the others were undergraduate students. All subjects are currently studying at Essex University.

When asked for their searching behaviour, the average value was 4.75, where 5 means daily, 4 means weekly. No one selected values 1, 2 or 3. An interesting observation is that only one subject strongly agreed (i.e. value of 5) with the statement *"I enjoy carrying out information searches."*, two others selected 3, everybody else selected 4 (i.e. average of 3.94).

The average length of the initial user query was 2.35 words (*System A*: 2.41, *System B*: 2.30). This figure is very similar to 2.3 that was calculated as the average length of Web search queries in a study that evaluated nearly a billion queries submitted to the *AltaVista* search engine [12].

Overall, the average time spent on a search task on *System A* was 277.5 seconds, on *System B* 293.7 seconds (no significant difference has been found). The figures show that in average users were able to find answers quicker with *UKSearch* for half of the search tasks, whereas the standard search system was quicker for the other half. However, the search time is only one aspect and the results do not correlate exactly with user satisfaction as we will see shortly.

We also looked at the number of "turns", i.e. the steps it took to find the answer for a search task. A turn can be inputting a query (or modifying the query), selecting a modification option, following the link to the next ten matches or following a hyperlink to open a document. We found that in general if a user needed more time on one system to complete a task, then the user would go through more turns than a user on the other system. There is only one significant result. For task 2 users needed significantly fewer turns on *System B* ($p < 0.05$).

After finishing each search task a post-search questionnaire had to be filled in. Most interestingly, users were overall slightly (though not significantly) more satisfied with *System B* than with *System A* (4.05 *vs.* 3.95). For search task 8 users were significantly more satisfied with *System B* ($p < 0.01$). The other differences are not significant. The results indicate the tendency that the more difficult questions are better handled by *System B,* whereas the basic system is better at dealing with the more straightforward questions.

In the exit questionnaire users were asked to answer the question *"Which of the two systems did you like the best overall?"*. 9 users prefered *System B,* 6 prefered *System A* and 1 found no difference. This is a very encouraging result, also because the underlying search engine in both systems was Google, which means that *System B* essentially competed with a version of Google. Note, however that the users were not told anything about the underlying search engine.

So far we have mainly been concerned with the statistical evidence. We also want to discuss some of the other issues, such as feedback that came from the subjects as well as patterns in users' search behaviour.

Many users liked the general idea of having refinement options (as they were offered by *System B*). Some problems with these options were that they were not always good and that there should be more help so that one knows exactly what the options are. It was also noted that more time would be required to learn to use the system more efficiently. One user commented that *System B* may be better when searching a broad topic or searching huge document collections.

Users typically liked the simplicity of *System A*. One user noted *"I like system A because it more or less reflects my web searching habits and feels more natural to use."*

Misspellings are a problem. The word *accommodation* was misspelled frequently (16 out of the 128 search tasks contained queries with typos; of those there were 11 in tasks 1 or 8, topics that had to do with accommodation). The actual problem is the fact that there is indeed a sufficiently large number of documents matching the misspelled query. A possible solution for such problems is Google's approach, i.e. to present the results as usual but also ask the user explicitly *"Did you mean: ...?"*

Two users ignored all the options that *UKSearch* proposed altogether and instead used the system just like a standard search engine. This is not a problem. In fact, it is one of the intended features of the *UKSearch* system - users may select one of the proposed modification options or completely ignore them if they wish to.

Finally, it must be said that such an evaluation is a difficult task. On the one hand it is desirable to have a selection of real users (i.e. Essex students in this case), on the other hand their previous knowledge will vary dramatically. An indicator are the queries users submitted for search tasks 1 (*"Accommodation"* vs. *"Wivenhoe House Hotel"*) and task 4 (*"university football pitches"* vs. *"Sports Centre"*). Larger scale evaluations will be necessary to address this problem.

5 CONCLUSIONS

We discussed the usefulness of automatically acquired domain models that can be constructed by exploiting markup structure found in documents. We concentrated on the question of how useful such models are when applied to ad hoc search tasks. We found that term pairs encoded in the automatically extracted hierarchies seem to be significantly better suited as query modification terms than those constructed on the fly using a baseline approach. We also found that the first task-based evaluation which applied such a model has been successful, although significance could only be shown for certain aspects.

The task-based evaluation presented here is the first step towards a more comprehensive investigation into the usefulness of markup-based knowledge extraction applied to ad hoc search. One aspect to be looked into is the choice of domain. Will users be happier (or find documents quicker) with a system that assists them if the document collection is much larger/smaller or more/less homogeneous etc.? The user study which focused on the quality of the domain models would suggest that the domain model for the BBC News domain might lead to more conclusive results in a task-based evaluation.

Another future issue is the question of how we can better combine the benefits of a standard search engine with the options proposed by *UKSearch*. The results seem to suggest that each of the systems is better suited for a particular type of tasks. The question is how to use the potential of both systems to find a solution that is better than any of the two individual systems.

REFERENCES

[1] ALANI, H., KIM, S., MILLARD, D. E., WEAL, M. J., HALL, W., LEWIS, P. H., AND SHADBOLT, N. R. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems 18*, 1 (January/February 2003), 14–21.

[2] ANICK, P. G., AND TIPIRNENI, S. The paraphrase search assistant: terminological feedback for iterative information seeking. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA, 1999), pp. 153–159.

[3] CRASWELL, N., HAWKING, D., THOM, J., UPSTILL, T., WILKINSON, R., AND WU, M. TREC11 Web and Interactive Tracks at CSIRO. In *Proceedings of the Eleventh Text Retrieval Conference (TREC-2002)* (Gaithersburg, Maryland, 2003).

[4] FELLBAUM, C., Ed. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[5] GUARINO, N., MASOLO, C., AND VETERE, G. OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems 14*, 3 (May/June 1999), 70–80.

[6] HERSH, W. TREC 2002 Interactive Track Report. In *Proceedings of the Eleventh Text Retrieval Conference (TREC-2002)* (Gaithersburg, Maryland, 2003).

[7] HERSH, W., AND OVER, P. TREC-9 Interactive Track Report. In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)* (NIST Special Publication 500-249, 2001), pp. 41–50.

[8] KRUSCHWITZ, U. A Rapidly Acquired Domain Model Derived from Markup Structure. In *Proceedings of the ESSLLI'01 Workshop on Semantic Knowledge Acquisition and Categorisation* (Helsinki, 2001).

[9] KRUSCHWITZ, U. An Adaptable Search System for Collections of Partially Structured Documents. *IEEE Intelligent Systems 18*, 4 (July/August 2003), 44–52.

[10] MAEDCHE, A., MOTIK, B., STOJANOVIC, L., STUDER, R., AND VOLZ, R. Ontologies for Enterprise Knowledge Management. *IEEE Intelligent Systems 18*, 2 (March/April 2003), 26–33.

[11] SANDERSON, M., AND CROFT, B. Deriving concept hierarchies from text. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA, 1999), pp. 206–213.

[12] SILVERSTEIN, C., HENZINGER, M., AND MARAIS, H. Analysis of a Very Large AltaVista Query Log. Digital SRC Technical Note 1998-014, 1998.