Testing Smart Information Retrieval Prototypes

Bert Paepen, Sven Van Hemel
Katholieke Universiteit Leuven, Department of Electrical Engineering,
Research Group on Document Architectures,
Kasteelpark Arenberg 10, 3001 Heverlee, Belgium.
{bert.paepen, sven.vanhemel}@esat.kuleuven.ac.be

Markus Schranz
pressetext.Austria Nachrichtenagentur GmbH,
IT Research & Development,
Josefstaedter Strasse 44, 1080 Vienna, Austria.
schranz@pressetext.at

ABSTRACT

The OmniPaper project has implemented three information retrieval prototypes in the area of electronic news publishing. One prototype uses SOAP as communication protocol between the central system and a number of distributed news archives. The second prototype uses an RDF metadata database, enabling direct metadata queries to the central system. Finally the Topic Map prototype uses query expansion and keyword-concept semantic linking for smart metadata search. An important goal of the OmniPaper project has been to compare these different information retrieval methods and prototypes. This paper explains how this comparison has been done using an Automatic Testing Engine. It further shows how user feedback has been used as an additional evaluation method.

KEYWORDS

Information retrieval, testing, evaluation, Topic Maps, RDF, OmniPaper

1. INTRODUCTION

Since the birth of the Semantic Web [8], more and more metadata has become available across the Internet. Nevertheless, search engines are still mostly based on full-text search mechanisms. In spite of their popularity, full-text search engines are basically brute force machines, crawling the web and indexing its entire contents. This technique has proven to be powerful and fast. On the other hand, its search capabilities are limited to the exact words of the query. By consequence full-text searching is especially powerful for the experienced user familiar with full-text searching and its limitations. Experienced users typically try synonyms or different writings of words and combine them with Boolean operators to get the best results.

The OmniPaper project [1] is investigating how searching can be made smarter and more accessible for inexperienced users using metadata, especially keywords. The project prototypes are applied to the area of electronic news publishing. In a first stage, the OmniPaper consortium has developed a number of prototypes using different technologies: SOAP [2], RDF [3] and Topic Maps [4]. In order to make the testing process easily reproducible, these prototypes have been tested using an Automatic Testing Engine.

This paper outlines the basic principles behind the testing process and the Automatic Testing Engine. First the different prototypes are summarized briefly. Then the principles behind the testing process and the Automatic Testing Engine are explained. Finally the main test results are discussed.

2. OMNIPAPER PROTOTYPES

The key objective of the OmniPaper project is “the creation of a multilingual navigation and linking layer on top of distributed information resources in a self-learning environment” [1]. As a proof of concept, the

project consortium is constructing a system that enables users to have simultaneous and structured access to news articles originating from a large number of digital European newspapers.

In the final OmniPaper prototype the user is able to submit a natural language query in his or her language, retrieving results in multiple languages. The linked keywords that form a navigation layer are automatically translated in the different languages that exist in the various archives. That way, readers can look up news information without having to know anything about the language of each of the archives. Newspaper articles themselves will be in the original newspaper's language, but can be (semi-) automatically translated at the user's request.

SOAP is used to communicate between the central OmniPaper system and a number of distributed electronic news archives. In the **SOAP prototype**, requests from users are forwarded by the central system to the distributed archives, which solve the request and return the results. Search is implemented here as pure full-text search in the entire contents of the news articles. Thanks to the platform- and programming environment independency of SOAP, the distributed database can be diverse in hardware, software and database structure. In practice, Java and a SQL Server database is used for the central system (K.U.Leuven in Belgium), C++ and an Autonomy database are used by one of the archives (My News in Barcelona).

RDF is used as a building block for metadata. The **RDF prototype** stores the metadata of news articles in the central system using RDF Gateway (a native RDF database system for storing and retrieving metadata). When a user submits a request, this is handled by the RDF Gateway system, which returns results directly to the user (using RSP scripts). So the RDF search is implemented as a pure metadata search, searching on the keywords of news articles.

The **Topic Map prototype** [5] includes a "knowledge layer" of semantically related keywords, stored using the Topic Map paradigm[4]. The OmniPaper Topic Map consists of real-world concepts, semantic relations between concepts and news articles about these concepts.

When a user submits a query, the words of the query are looked up in the Topic Map. Then the search engine locates (the) corresponding concept(s) for each word. From these concepts, other related concepts can be found using the existing semantic relations. If a relevant concept is found, its keywords are retrieved, so that its corresponding news articles can be shown in the result list. The key to this search mechanism is the link between a keyword and the news articles (occurrences) it relates to. This link is provided by automatic keyword extraction: a set of statistical data mining techniques to assess the importance of a word in a specific text. The more important a word is in a text, the more important this text becomes in the search results of a query statement. In brief, this prototype uses semantic query expansion in combination with a keyword-based search. No full-text search whatsoever is used.

3. CROSS-TESTING OF PROTOTYPES

Combining these prototypes (SOAP, RDF and Topic Maps) has been only possible after testing the different aspects of each of them thoroughly. Therefore the project consortium has put an important effort in the definition of this testing process. First the criteria for testing where defined. Second, based on these criteria, a number of test sets where created. Then an Automatic Testing Engine was created that allows fast automated testing of all prototypes at once. Finally, the different prototypes where tested. Based on these tests, the prototypes have been combined into the "best combination of parts".

3.1 Comparison criteria

The RDF and Topic Map prototypes have been examined and compared to see the way they describe, store and search the metadata. It is worth noting that the SOAP prototype does not store any metadata by itself, so the comparison between the SOAP and the RDF prototypes is to be distinguished from that between RDF and Topic Maps. Thus, there are two different comparisons: RDF vs. Topic Maps and SOAP vs. (RDF and Topic Maps).

The prototypes have been tested and compared on a numerical and objective basis to measure the efficiency and effectiveness of the technologies used. This has brought the consortium to the use of the following numeric comparison criteria:

- Relevancy: measured by recall, precision and F-value which are traditional measurement instruments in Information Retrieval (IR) [6]
- Response time: elapsed time between the moment a search engine has initiated a query in the system and the moment the results for that query have been calculated and are ready to be returned
- Data size: how much extra bytes each prototype uses for the same set of articles

It must be stressed that this kind of numerical comparison only measures the "data-lookup" capabilities of the prototypes. Other very important aspects that influence the usability of the prototypes, like user interactivity and user friendliness, cannot be assessed by the numerical comparison. For these kinds of evaluation, an observational study has been done using a simple questionnaire for the searcher. These evaluations are described in section 4 of this paper.

3.2 Creation of test sets

A second step in the test process definition was the creation of a test set that can be used for testing the different prototypes. A number of test topics were chosen from four major news domains: "Politics, International, Law and Society", "Business, Industry and Science", "Education and Family" and "Entertainment, Arts, People and Others". Each test topic has a set of queries that the test set creator thinks are relevant for finding information about this topic. A number of metadata restriction fields can be added to these queries if necessary (to invoke an "advanced" query e.g. to restrict the date). The relevant answer set (the collection of relevant documents for each query) has been pre-defined by hand and served as the master answer set to which result sets from the different prototypes have to be compared.

3.3 Automatic Testing Engine

A next step in the testing process was the creation of an Automatic Testing Engine. The purpose of this engine is to automate the cross-testing of prototypes.

This has several advantages. First, testing becomes faster. Since the prototypes have been tested on a predefined set of 20-25 topics, with each topic having 3-5 queries, trying out all these queries by hand and writing down the results is too much work. Therefore, it is easier to adapt the prototypes according to the test results and re-testing the new prototype versions. Finally, reporting is easier, faster and more uniform.

Through a web interface, the test user can submit an input XML file containing the test topics, queries and answer sets. After taking input from the user, the Automatic Testing Engine forwards this input in a certain order to the three prototypes discussed earlier. Because some of the prototypes have parameters that can be tuned, the Automatic Testing Engine can also send the queries to a different number of prototype variants, each having a distinct parameter configuration. The prototypes return their results back to the Testing Engine, which puts the results together and outputs it to the user in the form of test reports. These reports are both in textual (HTML) and graphical (SVG) form. All results can be consulted online or downloaded in a zip file.

3.4 Test reports

The goal of the test report generator is to create clear, easy to interpret and uniform summaries of each testing session. A test report contains a number of summary tables that compare the different prototype's performance on the predefined sets of topics and their queries. The same prototype can be tested with different parameter settings and the results will be indicated as coming from a distinct prototype variant, so that the influence of the parameters can be examined within a single testing session.

The values that will be compared are relevance, precision and F-value for the retrieval performance and network, database and processing time for the response time. The results for a complete topic are calculated by averaging over all queries defined for that topic. The overall results will be calculated in two ways: by averaging over topics and by averaging over queries.

Besides the summarizing tables, a test report will also contain a number of graphs for enhancing quick visual inspection and comparison of the results. Three interesting graphs can be distinguished: the precision-recall graph, ROC curve and the cumulative response time graph. These graphs will be drawn for each topic and for the overall result, after applying appropriate averaging over the queries. For each topic, all the distinct

prototype variants can be drawn on one graph to facilitate inter-prototype comparison of the performance. Another important use of the graph is to determine optimal cut-off values for relevance rankers.

Table 1. Example summary table for search performance (note: numbers are fictitious)

Performance	SOAP	RDF	Topic Maps variant 1	Topic Maps variant 2
Topic: military action in Iraq	0.556	0.727	0.555	0.500
Query: Military force Iraq	0.535	0.545	0.545	0.455
Query: War Gulf military	0.606	0.500	0.534	0.318
Topic: Museum exhibition in London	0.486	0.143	0.500	0.357
Query: museum exhibition London	0.464	0	0.381	0
...				
Average over queries	0.525	0.334	0.316	0.156
Average over topics	0.621	0.465	0.367	0.230

A **recall-precision graph** shows recall in the abscissa and precision in the ordinate and can be used for both rankers and classifiers. A good performance is characterized by both high precision and recall. Most rankers produce a curve that has high precision at low recall levels, with decreasing precision at high recall. It is possible to draw curves for an optimal and a random ranker and to compare the actual performance with these curves.

ROC curves show the FPR (false positive rate) versus recall and can be used for both classifiers and rankers. ROC stands for Receiver Operating Characteristic analysis and was originally used in signal detection theory, but the graph and its interpretation can also be applied to information retrieval. False positive rate is defined as the proportion of documents in the answer set that are not in the relevant set (false positives) as compared to the total number of non-relevant documents.

A last interesting graph is the **cumulative response time** as function of the number of returned results. This graph shows the dependence of the response time on the number of search results. Database access time, network time and internal processing time can be plotted cumulatively so that the evolution of their relative importance with respect to the number of returned results can also be examined.

3.5 Test results

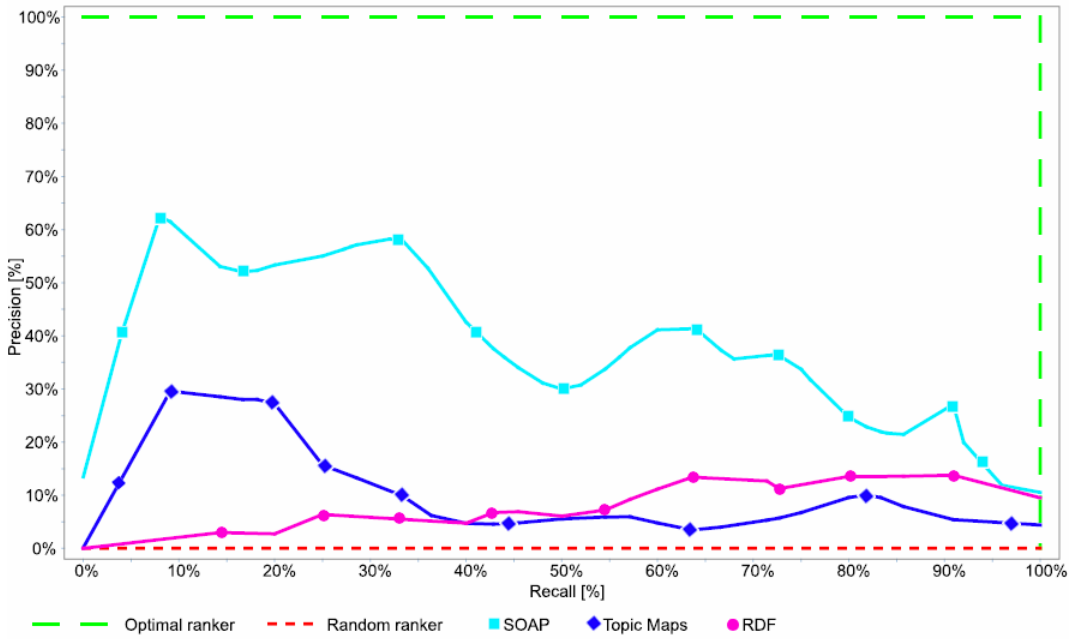
The figure below shows the recall-precision graph of the three prototypes. This is an averaged curve, taking into account 45 predefined queries clustered into 18 topics. It can be noticed that the precision for all prototypes is very low. This is partly due to the fact that all prototypes have the tendency to return lots of results, of which only a small fraction is relevant. However, for a good ranking algorithm, it is expected that the top results contain a considerable amount of relevant results. This means that high precision for low recall values should be seen on the graph. A quick inspection of the graph learns that this is the case for SOAP and Topic Maps (in lesser amount), but not for RDF.

This fact must be interpreted with care however: a more detailed study of the results on a per query basis shows that there exists a great variety in the quality of the results. For RDF 22 out of 45 queries do not return any results at all, for SOAP 20 out of 45 and for Topic Maps 14 out of 45. These results have a considerable influence on the averaged recall-precision graph. The fact that Topic Maps have less queries that have no results, is partly due to its stemming and to the fact that words found in the topic map will be expanded to synonyms that do return results. This means that the semantic search (Topic Maps) is behaving worse than the full-text search (SOAP); RDF is performing even lower. Keyword expansion however is an interesting aspect from the Topic Maps prototype, proving to give a good ranking behaviour.

Two smaller tests were performed (among others) that are worth mentioning here. First, the influence of the automatic keyword extraction (AKE) algorithm on the Topic Map search engine was tested. This test shows the improvement of the accuracy of the Topic Maps prototype due to a new version of the AKE algorithm. The average F-value has doubled to about 23% and only 6 queries do not return any results. This

test shows that the quality of the Topic Maps prototype depends on all parts of the search process. The success of the approach is highly (but not exclusively) dependent on the quality of the AKE process.

Figure **1**: Recall / precision graph comparing SOAP, Topic Maps and RDF prototype



A second test showed the influence of the Porter stemming algorithm [7] on the search performance. Stemming is the automatic reduction of a word to its basic form (stem). If stemming is applied to a query and to the keywords in the Topic Map, more accurate search results can be expected. Tests revealed that stemming of the article keywords alone leads to a small gain in search performance, but the variability in the different topics and queries to too high to judge the significance of this increase. On the other hand, no stemming is better than only query stemming (without article stemming). This means that stemming can only be valuable if used carefully.

4. USER EVALUATION

During project execution, thorough end user tests have been scheduled at specific states of prototype development. The evaluation of the system prototype, focusing on the user interface has been realised by giving individuals (of the identified user target group) specific tasks and using an online questionnaire for recording their difficulties, needs and impressions. The evaluation focused on system usability, quality of service, system features, general utility of the prototype, and quality of the information presentation. The questionnaire contained multiple choice and open answer formats and was combined with a set of tasks that users had to fulfil using the prototype.

The OmniPaper prototype has been tested by 38 experts during the end user evaluation. The evaluation concentrated on news professionals, allowing experts in the field of news aggregation, news distribution and content creation to judge the qualities of the OmniPaper system. The demographic data collected during the test created a picture of the typical OmniPaper end user by selecting the most frequent answer given by the end users: the average OmniPaper end user is 26 to 35 years old, attended University or College, has more than 5 years experience with using computers, uses the Internet several times per day via a broadband connection.

The end user questionnaire was structured in specific sections on system usage, system features and overall user experience. The following sections provide an excerpt of the test results, explaining well

accepted system attributes as well as neglected and less useful features that are valuable results to refocus the user interface and system design for further improvements.

The majority of the end users assessed the OmniPaper prototype as easy to use. 58% of the end users took advantage of the query refinement although it was unexplained in the interface. More than 77% evaluated the question "Do you think your final result set consists of the most relevant articles?" on task1 in the questionnaire positively (i.e. grade 1-4 out of 7).

Whereas more than 60% of the users accepted the simple search feature as very useful, the full text search was graded rather low and assessed on an average of 3 of 7. Notable results from these sections were that the feature of showing all relevant results and the capability of sorting the retrieved items according to specific criteria are most relevant and well accepted. Regarding the specifically new features, introduced with the OmniPaper prototype, the features including the "web of concepts", i.e. the semantic relations between articles were accessed as very useful and important for the quality of the service, thus rated above medium level for 86% of the users. In contrast, the hierarchical subject view has been neglected by the end users.

The last section of the end user test focused on the personal experience with the OmniPaper prototype and was targeted at the impressions and emotions of the users. Two out of three end users answered that the OmniPaper prototype would allow them to find the requested information more quickly than in other services and that this service will be helpful in their job. 75% see OmniPaper to fulfil their professional needs in news information retrieval and 90% see especially the semantic associations in the web of concepts very useful for high quality content retrieval.

The overall satisfaction with the OmniPaper service was rated 4 or higher on a scale of 7 by 61% of the end users.

The early integration of end users into the OmniPaper system tests provides a sophisticated approach in service design and refinement. Although promising results have been collected in specific areas, some numeric and even textual remarks in the end user tests leave space for system improvement.

5. CONCLUSION

The development of an Automatic Testing Engine – although not planned from the project start – has proven to be a valuable investment in the OmniPaper project. It allowed statistical cross-testing of different information retrieval prototypes, resulting in a combined prototype.

Currently additional features are being added to this prototype, such as multilingual and multi-archive search. This way, users can enter a query in their own language, returning results from different newspapers in different languages. Based on the test and user evaluation results, additional system features will be added. For testing these additional features, the Automatic Testing Engine will be a very useful asset. For allowing better comparison, the next testing phase will be using a larger test collection from the CLEF campaign.

Future information retrieval projects can also benefit from a system like the Automatic Testing Engine. If this would be an open-source tool made available to all scientists working in this field, a lot of effort for generation statistical test results, creating reports, etc. could be avoided. Unfortunately, the Testing Engine needs several person years of work before it can really take up such a role – in other words, it could become a project on its own...

REFERENCES

[1] Bert Paepen et al, 2002, OmniPaper: Bringing Electronic News Publishing to a Next Level Using XML and Artificial Intelligence. e*lpub 2002 Proceedings*, Karlovy Vary, Czech Republic, pp. 287-296.

[2] Simple Object Access Protocol: http://www.w3.org/TR/SOAP/

[3] Resource Description Framework: http://www.w3.org/RDF/

[4] XML Topic Maps: http://www.topicmaps.org/xtm/index.html

[5] Bert Paepen et al, 2003, Smart Search in Newspaper Archives Using Topic Maps. *Proceedings of the 7th ICCC/IFIP International Conference on Electronic Publishing*, Guimarães, Portugal, pp. 251-259.

[6] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, 1999. *Modern Information Retrieval*. ACM Press, New York.

[7] The Porter Stemming Algorithm: http://www.tartarus.org/~martin/ PorterStemmer/

[8] Tim Berners-Lee et al, 2001, The Semantic Web. *Scientific American*, May 2001.