SMART SEARCH IN NEWSPAPER ARCHIVES USING TOPIC MAPS

SVEN VAN HEMEL[1]; BERT PAEPEN[1]; JAN ENGELEN[1]

[1] DocArch, Department of Electrical Engineering (ESAT), Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, 3001-Heverlee, Belgium
e-mail: {sven.vanhemel; bert.paepen; jan.engelen}@esat.kuleuven.ac.be

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 semantic linking for smart metadata search. The Topic Map prototype enhances the search experience by implementing a knowledge layer that combines the semantic content of a lexical database, consisting of concepts and keywords, with a metadata-set of newspaper articles. The linking between both is currently implemented at the level of keywords but will be developed at the level of concepts in the final prototype. The knowledge layer has been designed from a Topic Map point of view, although the XTM syntax has not been used to avoid performance issues. The consortium's adopted view on information publishing and retrieval considers querying and navigation as two very related actions that can both be captured under the name "search for relevant information". Navigation forces the user to follow predefined paths whereas querying enables the user to look freely for a suitable starting point. The query and navigation functionality is provided through a web engine and is build on top of the information structure of the knowledge layer.

Keywords: Topic Maps; search; newspaper; OmniPaper; knowledge management

INTRODUCTION

Since the birth of the Semantic Web, more and more metadata has become available across the Internet. Nevertheless, popular 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 method is relatively simple and can be implemented very efficiently through advanced indexing techniques. On the other hand, its search capabilities are limited to the exact words of the user query. Therefore full-text search is especially powerful for the experienced user.

The OmniPaper project (1) is investigating how searching can be made smarter 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 small prototypes using different technologies. SOAP (2) is used to communicate between the central system and a number of distributed electronic news archives. RDF (3) is used as a building block for metadata. A "knowledge map" of semantically related concepts, their keywords and articles is stored using the Topic Map paradigm (4,5). Keywords are extracted automatically from news articles using data mining techniques and can be reviewed by journalists or information officers using a dedicated workbench. All these prototypes will contribute to the final OmniPaper system. The aim of this paper is to explain the main principles behind the XTM prototype.

SYSTEM ANALYSIS

The goal of the OmniPaper prototypes is to enhance the user experience in finding online news of interest. The project obtains its data from online news sources managed by a number of news providers and tries to build an intelligent top layer on the data that consists of metadata and an intelligent search interface. In order to achieve this goal the available metadata must be stored in a structured way and a number of dedicated query and navigation mechanisms to access this data must be designed (see figure 1). On the other hand, the data storage component must allow for dynamic adaptation to the never-ending stream of information that flows into the system in the form of fresh news articles.

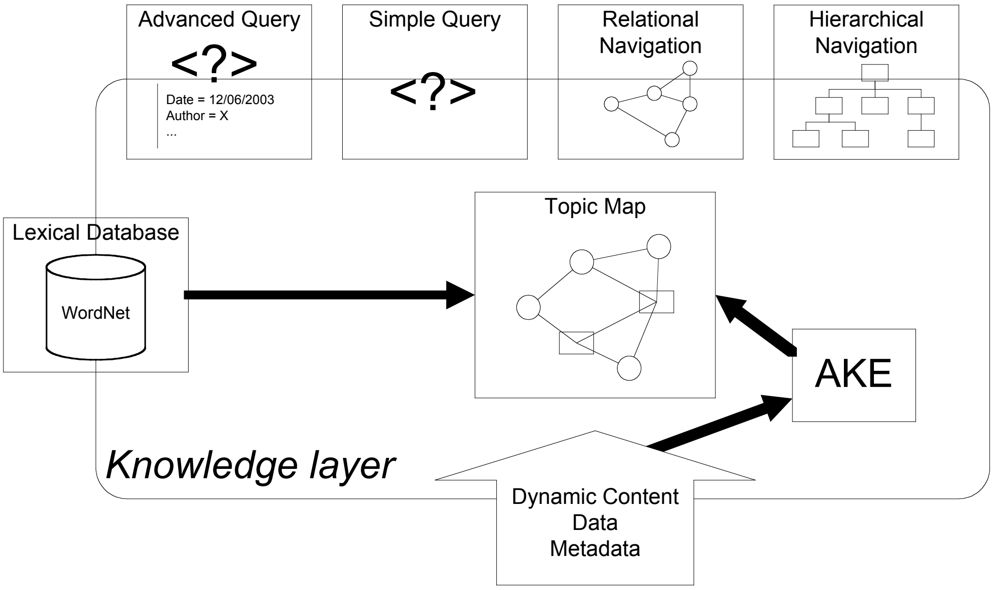

FIGURE 1 – SYSTEM DESIGN DIAGRAM

In this prototype, querying and navigation are considered as alternative methods to find relevant information. Both interact with each other and together they produce a combined user experience that can be expressed as "find what you were looking for and then browse away from it". In fact, the prototype considers both querying and navigation as a kind of search action. The only difference is that in navigation the user follows predefined paths, whereas in querying the user is totally free in what he or she submits as a query. Querying is a way of searching that provides the user with a starting point in the vast amount of available information. The use of Topic Maps for the storage of a concept map provides just the right means to support this point of view. Topic maps have traditionally been used for navigation purposes (5,6,7), but with the emergence of dedicated query languages (8,9), they have also become a useful tool for querying information.

This prototype implements four kinds of query and navigation. A first method allows users to navigate through news subjects (categories) in a traditional, hierarchical way. A more sophisticated tool is the relational navigation, where users can browse through a "web of concepts". The starting point for relational navigation (the "focus concept" in OmniPaper terminology) is the result of the last navigation or query action and the predefined paths that can be followed are paths to concepts that are related to the focus concept in the knowledge map.

Finally smart querying is enabled using a "knowledge map" of semantically related keywords and concepts. The idea is that the exact words of a user query are just a starting point for the search engine. Once the query is analyzed and basic search terms are extracted, they can be applied to the knowledge map, where they can be expanded to other related keywords and corresponding news articles (either semantically widened or narrowed). Two kinds of smart queries exist: simple query and advanced query, where advanced query is to be understood as a kind of filtering to restrict the number of results. When a query term is entered in combination with advanced query options, only the results that satisfy the advanced query constraints will be shown.

As mentioned before, the knowledge map (or knowledge layer as the project consortium calls it) is constructed using the Topic Maps technology. This International Standard provides a standardized notation for interchangeably representing information about the structure of information resources used to define topics, and the relationships between them (4,10). The main building blocks of a Topic Map are often referred to as the "TAO of Topic Maps": Topics, Associations and Occurrences. In the OmniPaper prototype topics consist of real-world concepts, keywords and stemmed keywords. Associations are semantic relations between concepts and links between concepts and their descriptive keywords. Occurrences of concepts are news articles about these concepts.

Topics exist at three levels: concepts at the top level, keywords at the middle level and stemmed keywords at the bottom level (see figure 2). A keyword is a meaningful word that exists in a news article, so excluding stop words. A concept is a broader term than a keyword: it is a real-world topic for which multiple synonymous keywords can exist. Semantic relations only exist between concepts; keywords are expressions of a concept in concrete words. For one concept, multiple keywords can exist in the map, but also a keyword can designate multiple concepts (homonymy and polysemy). The topic map is also enriched with a specific kind of concept, called subjects in the OmniPaper terminology. These are concepts that exist in the hierarchical view (so subjects are in fact predefined news categories).

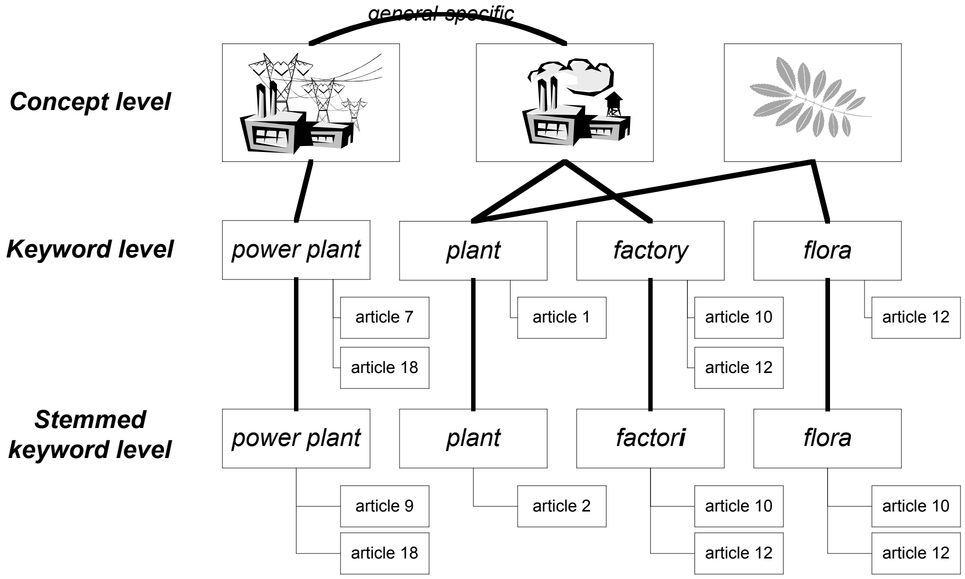

FIGURE 2 – KNOWLEDGE LAYER (TOPIC MAP) DIAGRAM

The knowledge layer consists of two important parts: a static set and a dynamic set. The static part can be considered as the foundation of the knowledge layer. It is a giant lexical database consisting of concepts and semantic relations between them, extracted from WordNet (11). WordNet is a lexical database for the English language and has been developed by the Cognitive Science Laboratory at Princeton University. Its design is inspired by current psycholinguistic theories of human lexical memory. It has been selected because of its applicability to the nature of the available information. Indeed, news articles are not about some specialized domain, but their subject can be practically anything. Therefore a knowledge map was needed that is able to cover all possible subjects. The idea of using WordNet to enhance search is not new. Attempts in this direction have already been made with rather positive results (12).

The dynamic part of the knowledge layer consists of a subset of this giant map that corresponds to the topics that are covered by the news articles in the OmniPaper metadata database. In a later phase of the OmniPaper project, the dynamic part will be updated with new topics and relations that emerge from breaking news. So the dynamic part only contains keywords that are relevant for the current data set.

When a new article is added to the database, automatic keyword extraction (AKE) is performed to attach keywords to this article. The AKE algorithm is based on the "term frequency-inverse document frequency"-model (TF-IDF model) (13). Two variants have been designed: one that extracts regular keywords and one that extracts stemmed keywords. Stemming is a process for removing the morphological and inflexional endings from words. It is mainly used as part of a term normalisation process. Stemming of extracted keywords is believed to improve performance as different morphological variants of a word will be recognized as originating form the same stem, hence the frequency measures can be calculated more precisely. If new keywords do not exist yet in the dynamic knowledge map, the map will be extended appropriately.

At this moment, news articles are considered occurrences of keywords in the map. In a later phase however, keyword extraction will be refined so that it becomes possible to bind news articles to concepts instead of keywords. This will be achieved by grouping extracted keywords into a weight vector, each position of the vector having a weight that corresponds to the belief that some keyword is present in the article. In a next step these vectors will be clustered into similar groups that in fact correspond to non-semantic concepts. In this view, a concept is defined as a typical weight vector. The challenge in this approach is to match the non-semantic concepts of the AKE-algorithm to the semantic concepts of the knowledge map.

When a user submits a query, the keywords of the query are looked up in the knowledge map. Then the search engine locates the corresponding concept(s) for each keyword. 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. A major problem that arises at this point is that of word sense disambiguation: given a certain keyword and its context, which of its associated concepts is to be selected? As the scope of the OmniPaper project does not allow a profound investigation of this problem, it will be handled by techniques that were gathered from specialized literature (14-16). At present however, it is not implemented yet as the system can only benefit from this approach when articles can be linked to concepts instead of keywords. For now the system still has to fall back to keyword level to retrieve articles.

Once a knowledge layer has been constructed for the underlying information pool, the possibilities become unlimited. Depending on the types of associations that exist in the Topic Map, practically any kind of user action and search guidance can be implemented. The OmniPaper prototype supports a number search guides: narrowed, widened and associated search.

SYSTEM IMPLEMENTATION

KNOWLEDGE LAYER DATA MODEL

The Topic Map prototype's data model consists of two parts. At the one hand, keywords and concepts from the lexical database must be stored to form a knowledge layer. At the other hand, there is metadata about news articles, which is stored in an SQL Server 2000 database.

As the topic map XML syntax (4) is rather verbose and the number of keywords and concepts to be stored is rather large (see table 1), it has been decided from the beginning not to use the XML syntax in order to avoid performance issues. Although there are many alternatives for storing topic maps, such as XML native databases that can be extended to dedicated topic map databases (provided by companies as Empolis, Ontopia and Mondeca), the consortium has decided to use the same database as was used for storage of the article metadata. This method has the advantage that the connection of articles to the keywords of the topic map is rather straightforward.

It must be stressed that, although the XTM syntax has not been used for this prototype, the ideas behind topic maps are still valid. The topic map database design has been built from a "topic map point of view" and with topic map terminology in mind. The use of a relational database is just considered another way to capture these ideas.

TABLE 1 – CONTENT DESCRIPTION OF XTM PROTOTYPE DATABASE

Item type	Number of items
Article	1881
Article-Keyword association	94076
Article-Stemmed keyword association	136185
General-Specific relation	111298
Equivalence relation	24312
Association relation	27026
Concept	111349
Subject	115
Keyword	138714
Stemmed keyword	120286
Concept-Keyword association	195953

OmniPaper uses WordNet 1.7.1 for the static part of the topic map. It is meant to constitute a firm basis for navigation and searching to which dynamic content can be added. The concepts and relations established in a wordnet are considered a static knowledge source because they are fixed semantic relations that will not evolve in time. As the WordNet database is freely available in ASCII format, a conversion step had to be taken to gather the wanted information and store it in the database. This extraction was done with OmniMark 5.3.

WordNet distinguishes about 50 different types of semantic relations. This means that there are a large number of possible relation types to connect two concepts. This is not desired for navigational purposes, because it will make the user interface look chaotic. Therefore, for the purposes defined in this project, the number of different relations has been reduced to three basics types: "equivalence", "general-specific relation" and "association" for all types that are not covered by the first two. Each of the WordNet types can be reduced to one of these three basic types.

The subject-topics for the hierarchical view are interconnected via "superclass-subclass" relations to reflect the hierarchical structure. Because these subject topics are nothing but special instances of concept topics, they can also have basic concept relations with concepts to which they are related in a non-hierarchical way.

Each concept has a preferred keyword and is linked to other descriptive keywords through "concept-keyword" relations that are taken from WordNet. On the other hand, each keyword is connected to its stemmed form. Because the predefined testing set of news articles consists of English documents, stemming has been performed with the Porter stemmer (17), the most wide-spread stemming algorithm for the English language.

In a later phase, also user-defined and content-defined relations will be added. These reflect the dynamic content of the system. User-defined relations will be extracted from data mining results on user behaviour whereas content-defined relations can be found by performing data analysis on the incoming news articles.

USER INTERFACE

The Topic Map prototype operates within a web server. When a user sends a search request (either a navigation action or a query) using his web browser, the web server activates the Topic Map Prototype to processes the request and return the result to the client. The layout of the user interface is shown in figure 3.

The relevant articles that result from a search request are initially shown as a limited-length list of items in the "content frame". The user can modify the number of items on one search result page using a drop-down list. Each item shows the resulting article's title, date, publisher and abstract and each title is a hyperlink to the full text content of the article. When an article has been selected, its full text will be retrieved from a local news archive via SOAP and displayed in a separate window.

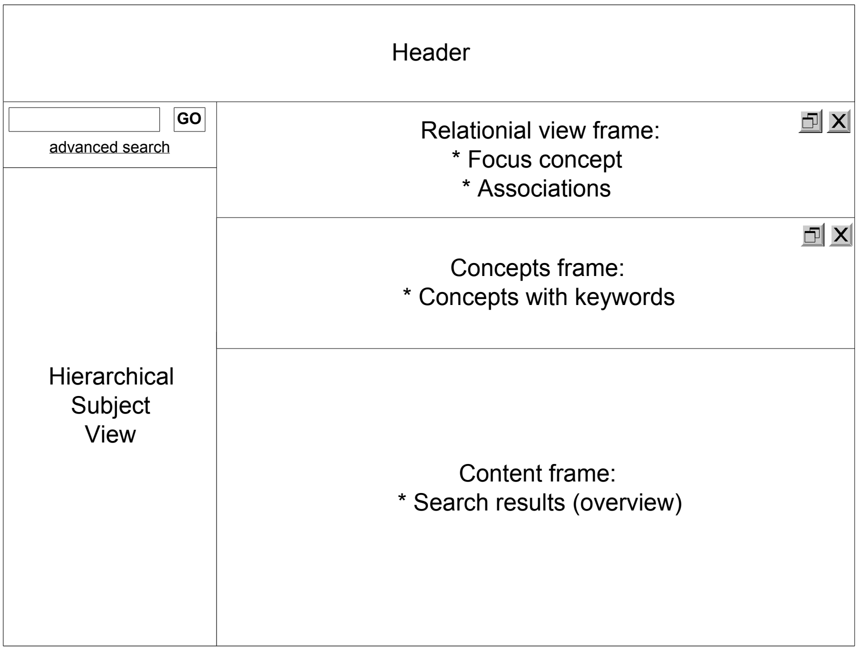

FIGURE 3 – USER INTERFACE LAYOUT

In the "query frame" (top-left below the header), users can submit Boolean queries to the system. A Boolean query consists of basic search terms, either words or phrases, and a set of Boolean operators ("AND", "OR", "–" and "+"). Submitting a query results in retrieving, for each basic search term, a list of concepts that matches the term (see below for a more detailed explanation).

While performing a search or navigation action, the system will maintain a list of "current concepts" that contains all the concepts that searched within at that moment. Each concept is shown as its preferred keyword followed by the other keywords that describe it. The keyword list helps the user to understand the semantic meaning the concept. Different concepts that are associated with an equivalence link are shown as only one concept.

Two main navigational views exist: hierarchical and relational. Both views approach the knowledge map in a different way. The relational view always shows the "focus concept". This is a concept from the current concepts list that can be examined in more detail and that can be refined in the relational view frame. For the focus concept, its related concepts are shown (according to the semantic WordNet relations). When a user clicks on such a related concept, that concept will be focused upon and the results page will be updated appropriately.

When a user clicks on a news subject in the hierarchical view (left-bottom frame), it is unfolded into its subcategories. The relational view is adapted so that the current subject becomes the focus concept. From this point, the user can browse to related concepts. When a leaf-subject (subject at the lowest level) is clicked, a list of news articles is shown that corresponds to the current subject. This list is constructed using the article's subject metadata field.

Two display modes for the relational view are foreseen: a textual mode and a graphical mode. In textual mode, the focus concept will be displayed with its relations and associations in a similar way as the Omnigator tool from Ontopia (18): information about the current topic is displayed and for each association type, a list containing the associated topics is shown. Links to these associated topics enable the user to redefine the focus concept.

In graphical mode, the relational view-frame shows a visualization of the topic map. Distinction (e.g. by means of colours) is made between concept-topics, keyword-topics and subject topics. Also the different relations should be displayed in arcs of different colours. The user will be able to navigate through the map by changing the focus concept. Different software packages exist to display topic maps in a graphical way such as Inxight's Star Tree (19) or Apache's TMNav (20) and the consortium is now figuring out the best way to implement this.

QUERY PROCESSING

As mentioned before, users can submit Boolean queries to the system, combining basic search terms with the Boolean operators "AND", "OR", "–" and "+"). An overview of the query process is shown in figure 4. When a query has been entered, it will be parsed into an *n-ary* query syntax tree that reflects the query's structure (see figure 5).

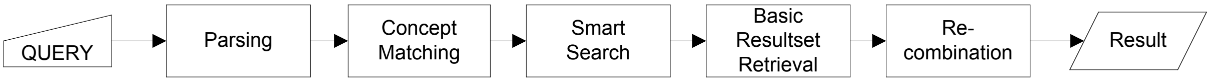

FIGURE 4 – OVERVIEW OF THE QUERY PROCESS

The "AND" and "OR" operators are considered binary commutative and associative operators and their use is straightforward. The unary "–"-operator, however, should be interpreted with care as it does not correspond to the classical Boolean "NOT" operator (13). This is because the "NOT" operator is an unsafe operator in the sense that is makes reference to a universe of elements. Instead the "–"-operator is only allowed in combination with an "AND" operator, in which case the query "A AND –B" should be interpreted as "A BUT NOT B". In other words, the "–"-operator works as a restriction on the results of A. The other unary operator is the "+"-operator. A query term that is preceded by a "+" is interpreted as a term of higher importance. This means that the "+"-operator is just a way for the user to indicate the relative importance of different query terms.

In the Concept Matching step, the topic map will be consulted to find for each basic query term the basic concept that matches it. This can be done using either the original query string or the stemmed query.

Query

(A and –B) or (C and +D)

Interpretation

Show me all articles that are about A but not about B or all articles that are mainly about D but also a bit about C

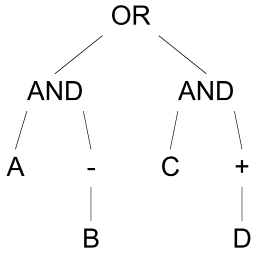

FIGURE 5 – (LEFT) EXAMPLE OF BOOLEAN QUERY AND ITS INTERPRETATION; (RIGHT) CORRESPONDING QUERY SYNTAX TREE

First the system looks for a keyword that corresponds to the basic search term and then it retrieves this keyword's concept. When a keyword with multiple senses (i.e. connected to multiple concepts) has been found, the system should try to disambiguate the keyword's sense. As this task is very difficult to automate, the prototype uses a simple approach for now: the first concept that has a preferred keyword that equals the basic query term is selected. If no such concept is found, the concept matching is broadened to the first concept that shares *some* keyword with the basic query term. For stemmed queries, the system first looks for a matching stemmed keyword, retrieves the regular keyword and then goes to the concept level.

Next, a Smart Search will be invoked upon the found basic concept, which means that the system will look for concepts that are equivalent to the selected basic concept. The basic concept and its equivalent concepts are joined together in a Smart Search concept list for that particular basic search term.

In the Basic Resultset Retrieval step, all articles that are linked to concepts of the Smart Search concept list are gathered to constitute the basic result set corresponding to the basic query term. Also in this step, two variants exist: one that finds articles based on their extracted keywords and one that matches extracted stemmed keywords. Note that in the future these variants will become obsolete, as articles will be linked directly to concepts.

The last step is the Recombination step. In this step article result sets from the different basic query terms are combined into the final query result. Recombination can be carried out using different models.

At the one hand a Boolean query can used to perform a "hard" classification of the basic result sets. This means that the Boolean operators are applied in the strict sense. This model is called the Boolean model and provides a classifier that labels articles as either being relevant or irrelevant.

A more advanced approach is to use a ranking model, such as the Extended Boolean model (13). This model is based on a combination of keyword ranking and operator ranking. A keyword ranking number can be calculated from the keyword's weight (as given by the AKE). This relevance number can be made more precise by incorporating a distance measure that indicates how far the actual used keywords are from the original queried terms. The operator ranking defines how to combine the keyword ranking results for each of the operators.

Consider the conjunctive Boolean query given by "A and B". According to the Boolean model, a document that contains either the term A or the term B is as irrelevant as another document that contains neither of them. However, this binary decision criterion frequently is not in accordance with common sense.

Instead of allowing the relevance value for documents with respect to a query to be either 0 or 1 (meaning the article is relevant or not), we could allow this value to obtain any value between 0 and 1, according to the belief that this document is indeed relevant and informative with respect to the query. A document's relevance is based on the weights of its extracted keywords. The weight $w_{k,j}^{(0)}$ of an extracted keyword k in a document j is calculated as

$$w_{k,j}^{(0)} = f_{k,j} \cdot idf_k$$

$$idf_k = \log_2\left(\frac{N}{n_k}\right) + 1$$

with $f_{k,j}$ the frequency of keyword k in document j and idf_k the inverse document frequency of the keyword (n_k is the number of documents in which keyword k occurs). In order to apply the extended Boolean model the keyword weights should be normalized between 0 and 1. This can be done as follows

$$w_{k,j}^{(1)} = \frac{f_{k,j}}{\max_x f_{x,j}} \cdot \frac{idf_k}{\max_i idf_i}$$

The normalized weight $w_{k,j}^{(1)}$ is the product of the normalized term frequency and the normalized inverse document frequency. Term frequency can be normalized with respect to the maximum term frequency of any keyword x in the considered document j. Inverse document frequency can be normalized with respect to the maximum possible document frequency (this value is attained for a term that only occurs in 1 document, i.e. $n_x = 1$). However, it has been found empirically that this kind of normalization leads to relatively low relevance numbers, as the weight can only obtain 100% relevance for a term that a occurs a maximal number of times in only one document. Therefore, $w_{k,j}^{(1)}$ has been normalized again with respect to a "relevant keyword" that occurs 75% of the maximum term frequency and in 12.5% of all documents. As this formula can lead to weights larger than 1, all results above 1 are cut off.

$$w_{x,j}^{(2)} = \frac{w_{x,j}^{(1)}}{0.75 \cdot \frac{\log_2\left(\frac{N}{0.125 \cdot N}\right) + 1}{\max_i idf_i}} = \frac{w_{x,j}^{(1)} \cdot \max_i idf_i}{0.75 \cdot (\log_2(8) + 1)} = \frac{w_{x,j}^{(1)} \cdot \max_i idf_i}{3}$$

$$w_{x,j}^{(3)} = \min(w_{x,j}^{(2)}, 1)$$

This formula provides normalized keyword weights for extracted keywords. To find the article relevance with respect to a basic query term, the basic query term is matched with extracted keywords in the Concept Matching and Smart Search steps as explained above. The relevance $r_{q,j}^{(0)}$ of an article j with respect to a basic query term q is given by the weight of the article's extracted keyword k that matches the basic query term: $r_{q,j}^{(0)} = w_{k,j}^{(3)}$ where q matches x.

Because the last step requires lookup of equivalent concepts in the topic map to expand the basic concept and keyword list, this figure can be further refined by incorporating a number that is a decreasing function of the (semantic) distance d travelled in the topic map (i.e. the number of associations between basic concept and expanded concepts). In this prototype the relevance of articles that have keywords from equivalent concepts has been diminished with a factor 0.9, but other schemes are also possible (21).

$$r_{q,j}^{(1)} = f(d(q,k)) \cdot w_{k,j}^{(3)}$$

Finally, operator ranking can be implemented by considering for each article its relevance with respect to the *m* basic query terms in an m-dimensional space. For a binary query, we have two basic terms and hence a 2-dimensional space. For an "AND" query, we want the combined relevance to be high only if the article has high relevance for both basic terms, for an "OR" query the relevance must also be rather high when the article has high relevance for only one of both basic terms. Final relevance measures can now be determined, based on a normalized p-norm distance.

$$r_{OR,j} = \sqrt[p]{\frac{\left(r_{q,j}^{(1)}\right)_1^p + \ldots + \left(r_{q,j}^{(1)}\right)_m^p}{m}}$$

$$r_{AND,j} = 1 \square \sqrt[p]{\frac{\left(1 \square r_{q,j}^{(1)}\right)_1^p + \ldots + \left(1 \square r_{q,j}^{(0)}\right)_m^p}{m}}$$

with *m* the number of query terms. For "–"-operators, $r_{a,j}$ should be replaced by $(1\text{-}r_{q,j})$ and vice versa. Priority terms (preceded with a "+") can be handled by diminishing the weight of all non-priority terms, so that the priority term will become more prominent. The value for *p* can still be chosen. However, *p=1* is not a good choice, as it takes just the average of both weights and hence makes no distinction between "AND" queries and "OR" queries.

The relevance for an article with respect to the complete query can now be calculated in a modular way, following the structure of the query tree. At this moment, the consortium is testing the different approaches in query processing (stemming or no stemming, ranking or no ranking, …) to determine the optimal solution.

CURRENT AND FUTURE WORK

In order to compare the different OmniPaper prototypes (including the Topic Map prototype variants described above) the consortium has put an important effort in the definition of a testing process. Different kinds of testing strategies are being implemented.

First, the prototypes will be tested and compared on a numerical and objective basis to measure the efficiency and effectiveness of the technologies used. Criteria for testing have been defined and a number of test sets were created, based on these criteria. For this purpose, the consortium has developed an Automatic Testing Engine. The main advantage of this testing engine is that it allows easy, uniform and well-defined comparison of the objective test criteria over time, so that performance of newly developed prototype variants can be tested practically "on the fly". Comparison criteria that are included in the analysis are divided into three groups: relevance (measured by precision, recall, F-value and ROC-measurements) (13), timing (process time, database time and network time) and data size. The numerical analysis will be used to select for each prototype the best variants and to combine the best aspects of each into an overall prototype.

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 will be done using a dedicated user testing workbench.

Another issue is user feedback. User feedback can greatly improve the user's perception of an intelligent system. An interesting issue to investigate is how WordNet can be used to provide more detailed user feedback. For example, the most relevant keywords that were used to assign an article as being relevant could be displayed in the result set. The use of WordNet makes it possible to provide the user with an explanation why these keywords have been considered relevant to the query. The trick is to use the semantic information contained in WordNet. For example, consider a simple query that has only one term: "car". Suppose the system returns an article having the keyword "engine" in it. This is possible, because "engine" is a meronym (part-of relation) of "car", so it is connected via a general-specific relation. The system can now justify its selection towards the user by stating that the selected article has been found relevant because it has a keyword "engine" and because "an engine is a part of a car".

The last issue to address is multilinguality. The architecture of the topic map prototype offers a very elegant way to include multilingual search capabilities. Whereas more simple solutions require translation of the query in the different languages or even translation of the data to one common language, the topic map prototype can incorporate wordnets for each of the desired languages into the knowledge layer. An interlingual index provides the translation between concepts of the different languages. This multilingual wordnet architecture has been developed in the EuroWordNet project (22) and the consortium is looking for ways to test its applicability in the topic map prototype.

CONCLUSION

The OmniPaper project aims at enhancing the user experience in searching for online news articles. Querying, hierarchical and relational navigation are combined into one intelligent search engine by implementing a knowledge layer on top of a number of distributed news archives. This knowledge layer consists of a static part extracted from WordNet and a dynamic part extracted from analysis of new incoming data. News articles are incorporated into the map as occurrences of their keywords. Different variants of the prototype exist and all will be tested using the consortium's Automatic Testing Engine and a dedicated testing workbench to select the best performing variant. The ultimate goal is to integrate the topic map prototype with the other prototypes to come to the final OmniPaper system that will combine the best parts of each approach.

REFERENCES

1. PAEPEN B., ENGELEN J., SCHRANZ M. and TSCHELIGI M. OmniPaper: Bringing Electronic News Publishing to a Next Level Using XML and Artificial Intelligence, in Carvalho JÁ., Hübler A., Baptista AA. (Eds), *Elpub 2002 Proceedings: Advances in Media Technology*, pp. 287-296, 2002.
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. PARK J. (Ed.). *XML Topic Maps: creating and using topic maps for the web*, Boston: Addison Wesley, 2002.
6. PEPPER S. *Euler, Topic Maps and Revolution.* Paper presented at XML Europe 99, Granada, 1999, Available from: `<http://www.infoloom.com/tmsample/pep4.htm>`.
7. LE GRAND B. and SOTO M. *Topic Maps et navigation intelligente sur le Web Sémantique*, Paper presented at les journées de l'Action Spécifique CNRS Web Sémantique, Paris, 2002, Available from: `<http://www.lalic.paris4.sorbonne.fr/ stic/octobre/octobre1/Le_Grand.pdf>`.
8. Topic Map Query Language, `<http://www.isotopicmaps.org/tmql/>`.
9. Empolis k42: Topic Map Query Language, `<http://k42.empolis.co.uk/tmql.html>`.
10. ISO/IEC 13250, *Topic Maps: Document Description and Processing Languages*. Available from: `<http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>`.
11. WordNet: a lexical database for the English language, `<http://www.cogsci.princeton.edu/~wn/>`.
12. MOLDOVAN DI., MIHALCEA R. *Using WordNet and lexical operators to improve internet searches*, IEEE Internet Computing, vol.4 no.1, pp. 34-43, 2000.
13. BAEZA-YATES R. and RIBEIRO-NETO B. *Modern Information Retrieval*, New York : ACM Press, 1999.
14. PATWARDHAN S., BANERJEE S. and PEDERSEN T. *Using Measures of Semantic Relatedness for Word Sense Disambiguation*, Paper presented at the Fourth International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City, 2003.
15. MAGNINI B., STRAPPARAVA C., PEZZULO G. and GLIOZZO A. Using Domain Information for Word Sense Disambiguation, in *Association for Computational Linguistics SIGLEX Workshop*, pp.111-114, Toulouse, 2000.
16. YAROWKSY D. Unsupervised Word Sense Disambiguation rivalling Supervised Methods, in *Proceedings of the 83rd Annual Meeting of the Association for Computational Linguists*, pp. 189-196, 1995.
17. The Porter Stemming Algorithm, `<http://www.tartarus.org/~martin/ PorterStemmer/>`.
18. Ontopia Omnigator, `<http://www.ontopia.net/omnigator/models/index.jsp>`.
19. Inxight Star Tree, `<http://www.inxight.com/>`.
20. Apache TMNav, `<http://www.folge2.de/work/tmnav/tmnav-doc-0-2-0/ credits.html>`.
21. BUDANITSKY A. and HIRST G. *Semantic Distance in WordNet: An Experimental, application-oriented evaluation of five measures*, Workshop on WordNet and Other Lexical Resources, Second meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, 2001.
22. EuroWordNet, `<http://www.illc.uva.nl/EuroWordNet/>`.