Bridging the Search Gap between the Web of Pages and Web of Data by Combining Ontological Document Expansion with Text Search

Matias Frosterus and Eero Hyvönen

Semantic Computing Research Group (SeCo)
Helsinki University of Technology (TKK) and University of Helsinki
http://www.seco.tkk.fi/
firstname.lastname@tkk.fi

Abstract. The Semantic Web extends traditional web documents, i.e. the Web of Pages, with conceptual structures based on ontologies and metadata, i.e. the Web of Data. This paper presents a hybrid document search approach combining the benefits of the traditional text search of literal documents and the semantic search based on their underlying conceptual structures. The approach is based on document expansion, where documents are automatically annotated with not only the concepts explicitly present in a given document, but also with the ontologically related concepts using smaller weights. Our test results using the CLEF Test Suite suggest that document expansion alone achieves better recall than text search at the expense of precision. As a solution, a method of combining document expansion with text search is presented in which better recall was obtained without sacrificing precision. This approach seems promising when integrating unstructured, textual content with the Semantic Web of Data.

1 Text Search vs. Semantic Search

The Semantic Web[1] extends web pages, documents, and other web materials with machine understandable, ontology-based [21] network of metadata attached to the contents. In the case of text documents, a central part of the metadata describes the subject matter of the text, and is directly related with the literal words and expressions of the documents. When searching text documents, two major approaches are available:

1. In *text search* the query is matched against the textual expressions of the documents, and search takes place in a literal space. In spite of the success of traditional text search engines, this approach has severe fundamental limitations [5] in its basic form. For example, recall is lowered because the query word cannot be matched with synonyms or semantically close terms. For example, query “student” does not match documents about pupils, “bird” does not match descriptions about eagles, and query “nordic country” does not match with Finland or Sweden. At the same time, precision is lessened due to polysemy and homonymy of words. For example, query “bank” matches financial banks, blood banks, and river banks. It is often possible to improve precision at the expense of recall, and vice versa [4].

[1] http://www.w3.org/2001/sw/

2. *Semantic search* [8] tries to address the limitations of text search by performing search in a conceptual space, based on disambiguated concepts rather than literal words, and by utilizing semantic networks of concepts underlying the texts. Such a search should be more precise since homonymous queries can be disambiguated before the query, or by clustering the results according to different interpretations of the query afterwards. At the same time, recall can be improved by extending the query to synonyms and semantically related ontological concepts by query expansion [15, 11].

Both of these methods can be further refined in the WWW environment through the use of links between pages to find relevant documents not included in the original result set and to rank the results to better reflect their authoritativeness [16].

Automated query expansion methods can be broken down into 1) methods based on search results and 2) ones based on knowledge structures, the latter of which can be further grouped into collection dependent and collection independent methods [4]. Methods based on search results first perform a query using the query terms as given by the user after which a new query is formed based on terms with high occurrence in the result set. Methods based on knowledge structures either use corpus-based knowledge of, for example, correlations between different terms or use some a priori knowledge like relations between different concepts. This latter approach lends itself well to *document expansion* where the query expansion is not done dynamically in response to a user query but rather in advance during indexing.

Unfortunately semantic search is not a panacea but has its own difficulties and limitations. For example, expanding the query or documents semantically raises recall but may dramatically lower the precision unless the expansion strategy is carefully tuned [11]. On the other hand, matching precise search concepts with conceptual metadata may lower recall because conceptual representations cannot model very well e.g. the uncertain or fuzzy meaning of real world concepts [9]. Furthermore, the research tradition of information retrieval [1] has produced lots of useful methods and techniques such as TF-IDF [19] for ordering search results according to their *relevance* w.r.t. the query. Oddly enough, the issue of relevance has not yet been discussed much in the semantic search community, although its has been a key issue behind the success of search systems such as Google.

It therefore seems worthwhile to investigate whether it is useful to combine ideas from text search with those of semantic search, as suggested e.g. in [10] for an optimal hybrid search strategy. This paper presents such a study in the application domain of news paper articles. In the following, we present a document expansion method utilizing ontologies by which the text documents can be annotated automatically and different semantic search strategies can be performed. We then test some combinations of semantic search and text search and measure the results based on an article data set of the CLEF Test Suite[2] and its golden standard. The results of our experiments suggest that substantial benefits in terms of precision, recall, and relevance can be obtained by combining methods of text and semantic search in smart ways.

[2] http://www.clef-campaign.org/

2 A Hybrid Search and Recommendation Architecture

2.1 Document expansion versus query expansion in the Semantic Web domain

The difference between document expansion and query expansion is basically the timing of the expansion step. In document expansion the terms are expanded during the indexing phase for each individual document. In query expansion only the query terms are expanded and this is done dynamically when a user performs a search into the database.

Document expansion, though less frequently used than query expansion, has some important benefits when used in the Semantic Web domain. First of all it does not put a strain on the search system computationally as it is performed in advance during the indexing stage. Also, when done using ontologies, it expands the documents or other resources to the Semantic Web of Data allowing these documents and resources to be linked together in new ways.

If the ontological expansion is done in the query phase, it expands only the query and not the actual documents and resources. This means that in order to link resources through more complex relations the links are made longer which, in the case of larger ontologies, can expand the query originally comprised of a few terms into thousands of terms. An example of this would be if the user searches a database for all documents that are about birds. Using a bird name ontology like AVIO[3] with 9740 individual species would find all the documents mentioning any individual bird species even if they lack the original query term, but the query itself would be almost ten thousand terms long. Most search engines have limits to the size of the input which renders queries of thousands of terms impossible.

2.2 Ontological Concept Clustering

The hybrid search architecture and process used in our case study, based on text search and semantic search using ontology based document expansion, is depicted in Figure 1. The process starts with the lemmatization of a given document (cf. the upper left corner of the figure). After this, stop words are filtered out and the text is indexed into a conventional term-document matrix [20].

In order to facilitate semantic search, each lemmatized term in a document is matched with ontological concepts using labels present in an ontology (cf. box “concept matching” in the figure). If a match is found, then the concept’s URI is added to the document’s metadata as a subject annotation. Several ontologies can be used during indexing and the concepts found from each ontology are saved in their own subject fields. The characteristics required of an ontology are simply the existence of concept labels and some sort of relations between concepts, so in theory thesauri can also be used, though the typically more limited relations would result in worse performance overall. A separate index is built for each ontology/field. The process is fully automatic since we are dealing with thousands of news paper articles, which makes manual checking infeasible. Homonymous terms are not disambiguated by human intervention or using other techniques semantically, but are indexed using multiple meanings.

[3] http://www.yso.fi/onki/avio/?l=en

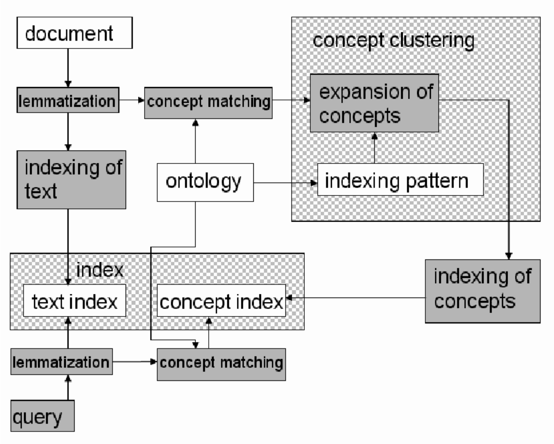


Fig. 1. The process of document expansion through ontological concept clustering

The document expansion is performed in the box “concept clustering” by expanding each matched concept c into a larger set that consists of other concepts semantically related to c in the ontology. The goal of using document expansion is to provide the user with larger, relevant result sets by using the semantic information underlying the documents. Document expansion is done by following an ontology specific pattern expressed in a pattern language developed for the task. A pattern is comprised of paths made up of the relations in the target ontology. Each path specifies the relations, or steps, that make up the path, the depth to which those relations are to be followed, as well as a weighting coefficient which determines the importance of related concepts found using the path. The idea of giving a weight to related concepts is an extension to earlier query expansion patterns such as [11]. For example, the direct superclasses of c can be given a high weight value, and the subclasses a smaller weight. The pattern language used is presented in more detail in section 2.3.

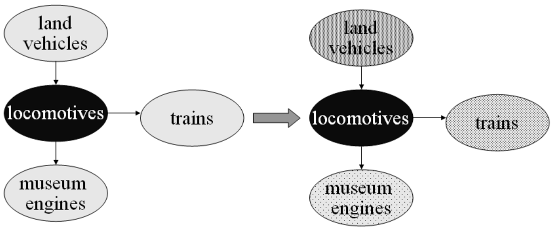


Fig. 2. The ontological expansion of a single concept

Figure 2 shows an example of ontologically expanding a single concept. The vertical arrows denote subclass relations while the horizontal arrows show related concept relations. When the concept 'locomotives' is found in the text, ontologically related other concepts are given weight (shown with black in the figure) depending on the nature of the relation.

The result of the document expansion is a cluster (set) of concepts annotating the subject of the document. The concepts literally present in the document have a weight of one and the semantically related concepts have typically smaller values between zero and one. In practice the weights of related concepts should be kept low and balanced so that they summarize the subject matter in a semantically correct and balanced way. For the final cluster, the weights are multiplied by the square root of the frequency of the occurrence of the original concept. This means that a concept gains more weight when it has a relation to several different concepts that are present in the document. The use of a balancing function (square root) in this step is needed in order to avoid a single concept with a high occurrence frequency from dragging its whole cluster up too high in the final index. Instead of square root some other balancing function could be used, which has some impact on the results. Finding the optimal balancing function is a whole new problem and is disregarded here.

After the concept clustering step has been performed for every concept found in the document, the clusters are added together and the weights are rounded to the nearest integer. The rounding is done because the URI of each concept is added to their respective ontological index as many time as the rounded sum of its weight indicates and this in turn allows the use of TF-IDF balancing in the concept index. This keeps the highly connected ontological concepts from dominating the search results.

When a query is issued into the system (cf. lower left corner in Figure 1, it is lemmatized and directed to the text search index as normal. On the semantic search side, an ontological concept matching is performed (in the same vein as when indexing the contents) and the resolved concepts are used as queries into the ontological concept indices created from the expanded documents. The result sets of all queries, based on text and semantic search, can then be combined in several different ways to produce different outcomes for the end user. Our research question is whether this can be done in a way that provides better search results than text search or semantic search alone. The choices and evaluation results of some strategies will be presented in the evaluation section (see 3.2).

2.3 Pattern Language

A crucial part of ontological document expansion is the pattern which defines the ontological relations that are to be followed when constructing a cluster around a given concept. A pattern is comprised of paths made up of hierarchical and associative relations in a given ontology. It is ontology-specific and should be tailored to a specific database in order to take full advantage of the proposed method as different domains place varying emphasis on different relations. Because of this patterns should be easy to construct when configuring the system for new applications. An XML-based pattern language was developed with this in mind.

The basic layout of a pattern is as follows:

- A pattern is comprised of one or several paths
- A path is comprised of one or several relations or steps
- Each path includes a weight which is applied to the resources at the end of the path

Each step of the path includes a relation and information on whether it should be traversed towards the object or the subject of the triplet. This has to be done because triplets are directed and not all relations have an inverse relation specified, but it can still be useful to traverse the relation in that direction. An example of this is the property rdfs:subClassOf, which is used to build the class hierarchy for ontologies. Its inverse, i.e. the superClassOf-relation, is not normally defined, yet it is often interesting to traverse the hierarchy towards subclasses, too.

Aside from these obligatory definitions, the pattern language includes a number of definitions for ease of use. First one is depth, which determines how many times a given step is to be performed until proceeding to the next step. Another is inclusiveness, which determines whether the weight is to be applied to every concept along the path or just to the final set at the end of the last step.

Relations of path	Depth	Weight	Is inclusive
subClassOf (s), subClassOf (o)	1,1	0.05	false
associativeRelation (s)	1	0.2	true
subClassOf (s)	1	0.05	true
subClassOf (0)	1	0.1	true

Table 1. The clustering pattern used for the evaluation

An example pattern is depicted in Table 1. Each row in the table describes one path. The first column shows the relations that make up the path with either (s) or (o) depending on whether the relation is to be followed starting from the subject or the object of the triplet. From the table we can see, for example in the last two paths, that a higher weight is given to the subclasses of a given concept than to its superclasses. Finally, an XML serialization of the pattern language was realized.

2.4 Searching vs. Recommending

An interesting question raised by document expansion is the relation between semantic search and *semantic recommendation* used as a key component in some semantic portals [13]. The idea of semantic recommendations is to provide the user with additional semantically related hits that are likely to be of interest to her, but that cannot be included in the search result. This is because the connection between the query and recommendations is not necessarily obvious, and the recommendations could look like wrong hits without further explanation. For example, by using our pattern language it is possibly to include in the result set of a query 'Finland' a document that is related to 'Sweden', a neighboring country, if the geospatial relation is considered important. The article may then not be connected with the query in terms of subject matter at all.

When expanding a query or a document semantically, the vague borderline between search hits and recommendations is easily crossed, and the actual search results get mixed with the recommendations. In our view, the distinction is useful and clarifying from an end-user's view point, as illustrated is systems such as [12, 14].

We therefore decided to investigate hybrid strategies between text search and semantic recommendation in our case study, too. For this purpose a recommendation scheme was devised that picks a number of the most relevant documents returned by the text search, for example ten. These documents are then searched for the concepts that occur in more than one document, and an intersection of the found concepts is used to form a new query into the concept index of the database. The intuition behind this scheme is that the shared concepts are likely to tell something semantically essential of the query and the underlying document set. Further constraints for recommendations are possible, too, based on metadata present in the original result set. For example a time window can be used so that the recommendation results must fit within a certain time interval based on the temporal metadata in the oldest and the newest document in the original result set.

After finding a result set of recommendations, those documents that are present in the original search result set should be removed, because recommendations are by definition complementary to direct search results. This recommendation method therefore provides an entirely separate additional set of documents that are strongly related to the original search query through concepts in the ontology and the actual metadata of the expanded documents.

3 Strategies for Hybrid Search and Recommendation

In order to test the architecture of Figure 1, an application using ontological concept clustering named Airo[4] was implemented using a dataset of 8000 articles of the newspaper Helsingin Sanomat[5]. Airo provides an implementation of the ontological concept clustering as well as text and semantic search capabilities based on it. Airo was coded in Java and it uses the Jena framework[6] for handling RDF(S) ontologies and Lucene[7] for search and indexing tasks. Automatic annotations of the data set were created using the tool Poka[8]. The General Finnish Ontology (YSO)[9] with over 20,000 concepts was used as the underlying ontology. YSO is a relatively simple ontology featuring associative, part of, and subclass -relations.

One design goal of Airo was to ensure its scalability to datasets comprised of millions of articles as is the case with electronic archives of newspapers. To this end the application needed to be fast, be able to adapt to material that is added daily, and be compatible with arbitrary ontologies. With the test configuration, the indexing of 8000 articles took about an hour, which means that indexing a million articles could be done

[4] http://www.seco.tkk.fi/tools/airo/

[5] http://www.hs.fi/

[6] http://jena.sourceforge.net/

[7] http://lucene.apache.org/

[8] http://www.seco.tkk.fi/tools/poka/

[9] http://www.yso.fi/onki/yso/

in less than a week. In reality the indexing would be done with more powerful computers which would cut down the time needed considerably. The process is also non-recurring and is needed to be done again only when the ontologies used for indexing are changed or new ones are added. Adding new articles to an existing index is very fast. Also, even though the size of the index is considerably larger, data storage space is not a concern these days when it comes to textual data. With modern search engines like Lucene, the system's effect on search time is negligible and practically independent of the size of the index.

3.1 Evaluating Search and Recommendation Strategies

In order evaluate different combinations of text search, semantic search and recommending, an evaluation test was first carried out. For this purpose, the test system of Cross Language Evaluation Forum (CLEF)[10] was used. The specific version used was ELRA-E00008 The CLEF Test Suite for the CLEF 2000-2003 Campaigns whose Finnish test set is comprised of articles from the newspaper Aamulehti and search tasks connected to these. The tests were done with all of the 60 search tasks of the year 2003.

The search tasks in the test suite are comprised of a title, which gives the topic of the task, a short description, which defines the task, and a longer narrative. The narrative describes the situation behind the task and the limitations on the kind of articles that are considered relevant to the query. Only the titles were used to construct the queries since Airo does not include the kind of natural language processing functions used for parsing search queries from narratives. The evaluation itself was done by comparing the articles given as a result to a search task by the system with a relevance file that lists the binary relevance of each article in the database for each query. It is worth noting that the database provided does not include any relevant documents for some of the search tasks. The pattern that was used for concept clustering is the one depicted in Table 1.

3.2 The Test and Results

Five different search strategies were used for each of the search tasks:

1. *Text search* refers to the traditional search where the lemmatized search terms were queried from the text index.
2. In *Concept search* the search terms were matched with ontological concepts of the YSO ontology and these were used to query the concept index.
3. *Text and concept search* combines the previous two queries through Lucene's Boolean should-operator which corresponds to a union.
4. *Recommendation* is comprised of the eleven most relevant articles gotten through the query expansion method described earlier.
5. *Smartly combined text search and recommendation* means that the fifteen most relevant text search results are listed first, after which the ten most relevant recommendation results are listed and followed by the rest of the text search results. The number fifteen was chosen here arbitrarily as a guess on how many topics a user

[10] http://www.clef-campaign.org/

might scan from the text search results before looking at the recommendations if both are shown at the same time next to each other. User tests would be needed to get an accurate number for this, but its effect is rather minimal as the CLEF Test Suite does not evaluate the order of the results.

A maximum of 1000 documents were considered when evaluating the result sets. The recall and precision of the five different search setups are depicted in Figure 3.

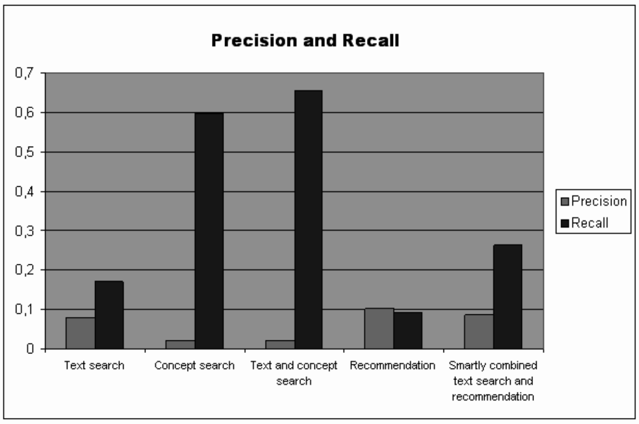


Fig. 3. The precision and recall of different search setups

The values of both precision and recall are between one and zero. The scores of text search should be regarded as the base level against which the others are compared to. From the figure it can be seen that the recall of both concept search and text and concept search combined are high but the precision of both is low. This is to be expected because concept search retrieves a much higher amount of documents than traditional text search and therefore returns also a large number of the relevant documents.

In recommendation precision is slightly higher and recall somewhat lower than in text search, the latter of which occurs because the maximum number of returned documents was set to eleven, which is lower than the number of articles listed as relevant in the case of some search tasks. A feature worth noting here is that due to the algorithm used, the result set is completely different from the result set that was gotten for the traditional text search. This can be seen in effect in the next setup, smartly combined text research and recommendation where the recall is simply the sum of the recall of text search and recommendation. Precision on the other hand is the average of the precision of the two component methods.

Straight comparison between the setups including all the results returned will not give an accurate idea of the qualities of the setups in actual intended usage of the system. An end user is not typically interested in hundreds of documents but rather scans the first few dozen results at maximum. Owing to this, precision with a certain maximum size result set is a meaningful measure and CLEF Test Suite produces this automatically. In practice this measure is calculated just like precision above, but taking into account only the n most relevant results. If the number of documents returned is less than n, the missing results are presumed wrong, which means that it is impossible to achieve perfect precision if n is larger than the total number of relevant documents for a given query in the database. When an average of the precision over all search tasks is calculated, comparing the different setups with different maximum number of returned documents is easy. This is depicted in Figure 4.

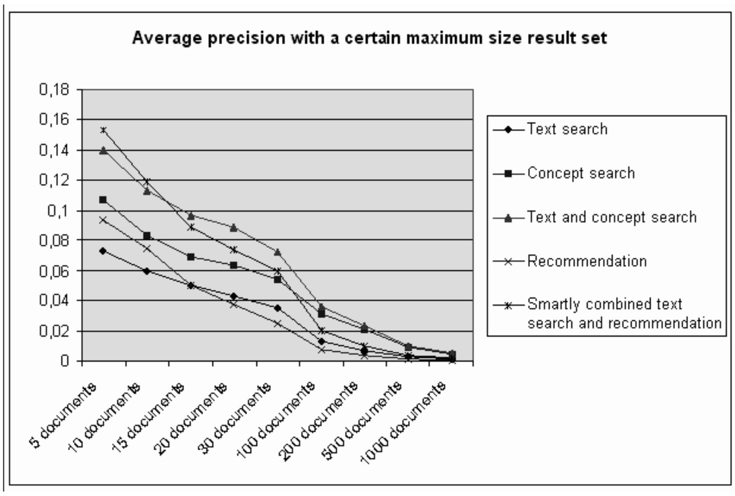
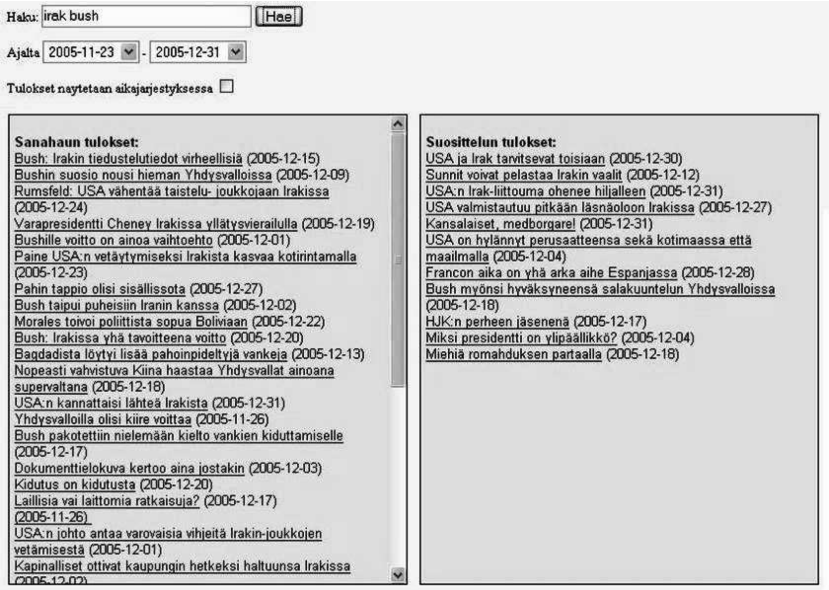


Fig. 4. Average precision with a certain maximum size result set

Traditional text search and recommendation have the lowest precision when viewed in this way while their combination has the highest with a low number of documents. With 15 documents or more, the text search combined with concept search is best. The aforementioned method of calculation where missing documents are considered false does skew the results especially with high maximum number of documents. When the maximum is low, though, the measure accurately simulates a real use case where the end user scans the first 10-30 results offered. This means that the simple combination of the text search and the concept search, though severely lacking in precision when considered over the whole result set, might still work in real life situations where the user is interested in only a few of the best ranked results. More tests are needed to draw definite conclusions.

3.3 Airo Application

Based on the evaluation the user interface depicted in Figure 5 was implemented to use the recommendation system detailed before to accompany the traditional text search.



Haku: irak bush

Ajalta 2005-11-23 - 2005-12-31

Tulokset naytetaan aikajarjestyksessa □

Sanahaun tulokset:
Bush: Irakin tiedustelutiedot virheellisiä (2005-12-15)
Bushin suosio nousi hieman Yhdysvalloissa (2005-12-09)
Rumsfeld: USA vähentää taistelu- joukkojaan Irakissa (2005-12-24)
Varapresidentti Cheney Irakissa yllätysvierailulla (2005-12-19)
Bushille voitto on ainoa vaihtoehto (2005-12-01)
Paine USA:n vetäytymiseksi Irakista kasvaa kotirintamalla (2005-12-23)
Pahin tappio olisi sisällissota (2005-12-27)
Bush taipui puheisiin Iranin kanssa (2005-12-02)
Morales toivoi poliittista sopua Boliviaan (2005-12-22)
Bush: Irakissa yhä tavoitteena voitto (2005-12-20)
Bagdadista löytyi lisää pahoinpideltyjä vankeja (2005-12-13)
Nopeasti vahvistuva Kiina haastaa Yhdysvallat ainoana supervaltana (2005-12-18)
USA:n kannattaisi lähteä Irakista (2005-12-31)
Yhdysvalloilla olisi kiire voittaa (2005-11-26)
Bush pakotettiin nielemään kielto vankien kiduttamiselle (2005-12-17)
Dokumenttielokuva kertoo aina jostakin (2005-12-03)
Kidutus on kidutusta (2005-12-20)
Laillisia vai laittomia ratkaisuja? (2005-12-17)
(2005-11-26)
USA:n johto antaa varovaisia vihjeitä Irakin-joukkojen vetämisestä (2005-12-01)
Kapinalliset ottivat kaupungin hetkeksi haltuunsa Irakissa (2005-12-02)

Suosittelun tulokset:
USA ja Irak tarvitsevat toisiaan (2005-12-30)
Sunnit voivat pelastaa Irakin vaalit (2005-12-12)
USA:n Irak-liittouma ohenee hiljalleen (2005-12-31)
USA valmistautuu pitkään läsnäoloon Irakissa (2005-12-27)
Kansalaiset, medborgare! (2005-12-31)
USA on hylännyt perusaatteensa sekä kotimaassa että maailmalla (2005-12-04)
Francon aika on yhä arka aihe Espanjassa (2005-12-28)
Bush myönsi hyväksyneensä salakuuntelun Yhdysvalloissa (2005-12-18)
HJK:n perheen jäsenenä (2005-12-17)
Miksi presidentti on ylipäällikkö? (2005-12-04)
Miehiä romahduksen partaalla (2005-12-18)

Fig. 5. Airo user interface

Recommendation was chosen as it showed improvements to the traditional text search in all of the scales used and was easy to add to the interface in an unobtrusive way that still leaves the text search in place.

In Figure 5 the results of the text search are shown on the left and on the right are the eleven best results that were gotten from the recommendation algorithm. The query has been for ”Iraq Bush” from the time period of November 23rd to December 31st in 2005. The text search results include many at least seemingly relevant titles of articles, but also some less immediately clear ones like ”President Morales hopes for a political peace for Bolivia”. Recommendation also holds seemingly relevant titles, especially at the top, but also less relevant ones like ”Franco’s time is still a sore subject in Spain”.

The number of recommendation results shown is a purely arbitrary number that would be simple to change, but finding the ideal would take some user testing and might depend on the dataset as well as on the ontology. The relatively low amount of recommendation articles shown hopefully keeps the user from being overwhelmed and showing them separately lets the user easily ignore them if they so wish.

4 Discussion

A noticeable thing about the results is the low precision score of all the test setups. This is caused by the fact that only the titles of the search tasks were used when creating the queries as the use of search task descriptions and narratives would have required the use of high level natural language processors, which were not available. On the other hand the use of only the title simulates somewhat accurately a real use case where the end user generates the first query quickly and refines it later based on the results gotten. This problem also affects both the baseline as well as the test methods so the comparison between them is still valid.

Perhaps the most crucial question when considering the evaluation results presented above is how great a problem is returning a number of documents even when none of them are relevant. This can be seen as a negative trait as the end user wastes time going over irrelevant documents while it would be better to formulate a new query. The recommendation system can be seen as bypassing this problem somewhat in that the results can be presented separately from the actual results and so the end user can read them or ignore them as they wish. A better way would be to devise a ranking algorithm that would allow blending both the text search and hybrid method results into a single, properly ranked result set, but finding out the right way of combining the results would require further research.

Regardless of the problem outlined above, combining the recommendation system with the traditional text search yielded better results than using the text search alone. Recall is much better without it adversely affecting precision. Concept search on its own might not be suitable for replacing text search, but as a component in a search engine it can produce additional value.

The result that was obtained represents the minimum this system is capable of in a sense. The ontology that was used had not been made specifically for the news domain and only one pattern was tested for the expansion. Using a more domain specific ontology and a less arbitrary pattern would yield better results as the expansion would then favour connections and concepts that are of interest to the specific use case of news articles. Even though the study is restricted to only one data set and domain, the fact that the system was not optimized for either of these has to be borne in mind. Generalizing from the fact that this evaluation ended up with positive results seems to indicate that ontological concept clustering holds promise.

Aside from the direct benefit to the Airo search application, recognizing the ontological concepts from the text and expanding them to include other, closely related concepts forms a good basis for further refinement and ties the documents into the Web of Data.

5 Related Work

In [18] the authors showed that document expansion using terms that are similar to the concept of the query, rather than the query terms, results in a notable improvement in retrieval effectiveness. An automatically constructed similarity thesaurus based on a specific index was used instead of a manually constructed ontology, which makes the process simpler as expensive ontology construction is avoided but also precludes the usage of relations and information beyond what can be automatically found in the corpus that is being indexed. They also used a probabilistic query expansion model, which similarly to our method, weighs the expanded query terms based on their similarity to the original query's concepts.

TAP [7] is a semantic search system, which tries to identify the concept of a search query and then show relevant data pulled from the Semantic Web to the user. In this their scope is the whole of the web and not a restricted dataset such as a news paper archive. Their approach is based on the user manually disambiguating the concept and the system then sorting the results accordingly without it necessarily performing actual query expansion.

Neptuno [2] aims to apply the techniques of semantic web to news paper archives. The semantic search system of Neptuno uses a specifically created news domain ontology whose concepts can be used in lieu of free query terms and the results can show specific parts of articles that have been annotated with the query concepts. The system also includes a separate visualization ontology which is a simplified version of the news domain ontology intended for making the navigation easier for end users. The greatest difference between Airo and Neptuno is that in the latter all the annotations are done manually while Airo aims for automation in order to make the indexing of existing news archives less labor-intensive. Also, the ontology in Neptuno is more aimed at broader classification than providing machine-understandable framework for the documents that are being indexed.

NEWS [6] has an automatic annotation component, which produces IPTC NewsCodes[11] classifications for news articles. It also recognizes persons, organizations and places based on linguistic as well as statistical properties. Unlike in Airo, annotations are based on a fairly limited number of classes, which are extensively instantiated and again the focus is not on fully annotating natural language terms into their ontological concept counterparts. Disambiguation in NEWS is done according to two principles: semantic coherence, which is somewhat similar to concept clustering, and news trends which takes into account the annotations in other news articles of the same period. Semantic coherence differs from concept clustering in that it is strongly based on previous articles and their annotations as opposed to ontological information.

KIM [17] is another semantic indexing, annotation and search system, whose central functionality is recognition of named entities instantiated from ontological classes. It also includes rules-based methods of recognizing and creating new instances from text. Disambiguation in KIM is done through clues based on wordlists, but disambiguation between entities with the same name is not discussed. Compared to the concept-based Airo, KIM is more focused on instances like individual places and people as opposed to document expansion using the ontological hierarchy.

6 Future Work

Much of the future work pertaining to Airo has to do with improving the extrinsic factors like the quality of the patterns and the ontologies used.

The chief problem in the evaluation was the limited amount of configuration that was done. The ontology used was not specifically designed for news domain and is therefore not tailored for the data or the use case that were used in the evaluation. As an example of this, YSO includes a number of two-way associative relations that are

[11] http://www.iptc.org/NewsCodes/

essentially one-way relations in the news domain. For example, the YSO concept of children as family relationship has an associative relation to incest. In most practical situations the concept of incest has an associative relation to children but not vice versa. Another example is the lack of many relations that are obvious to humans. For example the concepts of ice hockey and ice hockey players have no relation between them and they are in widely different places in the class hierarchy. Though in reality these two concepts are highly correlated, Airo could not make this connection. The only way to fix this problem is to use an ontology that fits the domain of the database better.

Also the pattern designed for the evaluation was the only one tested and it was not based on any deeper analysis of the ontology that was being used. One future interest is in creating a learning system which constructs optimized patterns based on training data. The simplest way of accomplishing this would be to create a set of paths based on the relations in the ontology and then varying the parameters on those paths until an optimal score in recall and precision was achieved.

One crucial component of the system is the original matching of terms found in the text to their respective ontological concepts. For the system to behave optimally, the concepts must be disambiguated properly. If the ontology is large enough with a relatively dense network of relations so that most of the terms in the documents that are being indexed can be found there, concept clustering could be used as a disambiguation tool. By making clusters for the concepts that were derived from unambiguous terms it is likely that these clusters give different weights to different possible concepts of the ambiguous terms. Testing this fully would again require a more comprehensive ontology than was available, but it is of future interest.

Further evaluation could also be done comparing the method to other advanced search methods such as LSI [3]. The advantage of semantic document expansion over LSI is that it is not corpus dependent and that the links between concepts mapped in an ontology have been done by domain experts and can therefore be taken as correct. Using this existing information in some capacity to aid in searching seems beneficial so a possibility of a hybrid approach combining LSI with semantic document expansion could also be considered.

7 Acknowledgements

This research was part of the National Finnish Ontology Project (FinnONTO) 2003-2007, funded mainly by The National Technology Agency (Tekes) and a consortium of 36 companies and public organisations. The work continues in FinnONTO 2.0 (2008-2009) project.

References

1. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, New York, 1999.
2. P. Castells, F. Perdrix, E. Pulido, M. Rico, R. Benjamins, J. Contreras, and J. Lores. Neptuno: Semantic web technologies for a digital newspaper archive. In *The Semantic Web: Research and Applications*, pages 445–458, 2004.

3. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. L, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.
4. E. Efthimiadis. Query Expansion. *Annual Review of Information Science and Technology*, 31:121–187, 1996.
5. D. Fensel. *Ontologies: Silver bullet for knowledge management and electronic commerce (2nd Edition)*. Springer-Verlag, 2004.
6. Norberto Fernandez, Jose M. Blazquez, Jesus A. Fisteus, Luis Sanchez, Michael Sintek, Ansgar Bernardi, Manuel Fuentes, Angelo Marrara, and Zohar Ben-Asher. News: Bringing semantic web technologies into news agencies. In *The Semantic Web - ISWC 2006*, pages 778–791, 2006.
7. R. Guha, Rob McCool, and Eric Miller. Semantic search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 700–709, New York, NY, USA, 2003. ACM.
8. M. Hildebrand, J.R. van Ossenbruggen, and L. Hardman. An analysis of search-based user interaction on the semantic web. Technical report, Amsterdam, 2007. http://db.cwi.nl/rapporten/abstract.php?abstractnr=2098.
9. Markus Holi and Eero Hyvönen. A method for modeling uncertainty in semantic web taxonomies. In *Proceedings of WWW2004, New York, Alternate Track Papers and Posters*, May 2004.
10. Markus Holi, Eero Hyvönen, and Petri Lindgren. Integrating TF-IDF weighting with fuzzy view-based search. In *Proceedings of the ECAI Workshop on Text-Based Information Retrieval (TIR-06)*, Aug 2006.
11. Laura Hollink, Guus Schreiber, and Bob Wielinga. Patterns of semantic relations to improve image content search. *Journal of Web Semantics*, 5:195–203, 2007.
12. E. Hyvönen, E. Mäkela, M. Salminen, A. Valo, K. Viljanen, S. Saarela, M. Junnila, and S. Kettula. MuseumFinland—Finnish museums on the semantic web. *Journal of Web Semantics*, 3(2):224–241, 2005.
13. Eero Hyvönen. Semantic portals for cultural heritage. In *Handbook of Ontologies, 2. edition*. Springer-Verlag, Forth-coming, 2008.
14. Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio, and Katariina Nyberg. Culturesampo – a collective memory of finnish cultural heritage on the semantic web 2.0. In *Semantic Computing Research Group, Helsinki University of Technology and University of Helsinki*, Sept 29 2008. Submitted paper, http://www.seco.tkk.fi/publications/.
15. K. Järvelin, J. Kekäläinen, and T. Niemi. ExpansionTool: Concept-based query expansion and construction. *Information Retrieval*, 4(3/4):231–255, 2001.
16. Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
17. B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov, and M. Goranov. Towards Semantic Web Information Extraction. *proceedings of ISWC (Sundial Resort, Florida, USA, October, 2003)*, pages 1–23, 2003.
18. Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, New York, NY, USA, 1993. ACM.
19. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical report, Ithaca, NY, USA, 1987.
20. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. 1983.
21. S. Staab and R. Studer. *Handbook on ontologies (2nd Edition)*. Springer-Verlag, 2008.