Proceedings of I-MEDIA '07 and I-SEMANTICS '07
Graz, Austria, September 5-7, 2007

Improving Search on the Semantic Desktop using Associative Retrieval Techniques

Peter Scheir
(Graz University of Technology, Austria
peter.scheir@tugraz.at)

Chiara Ghidini
(Fondazione Bruno Kessler, Italy
ghidini@itc.it)

Stefanie N. Lindstaedt
(Know-Center Graz, Austria
slind@know-center.at)

Abstract: While it is agreed that semantic enrichment of resources would lead to better search results, at present the low coverage of resources on the web with semantic information presents a major hurdle in realizing the vision of search on the Semantic Web. To address this problem we investigate how to improve retrieval performance in a setting where resources are sparsely annotated with semantic information. We suggest employing techniques from associative information retrieval to find relevant material, which was not originally annotated with the concepts used in a query. We present an associative retrieval system for the Semantic Desktop and show how the use of associative retrieval increased retrieval performance.
Key Words: semantic desktop, associative information retrieval
Category: H.3.3, I.2.4, I.2.6, I.2.11

1 Introduction

On the one hand it is largely [6] [17] agreed that semantic enrichment of resources on the web or desktop provides for more information to be used during search and that this can lead to higher effectiveness of a retrieval system. One the other hand critics [10] as well as advocates [13] of the Semantic Web agree on the low coverage of resources on the current web with semantic information. The sparse annotation of resources with semantic information presents a major obstacle in realizing search applications for the Semantic Web or the Semantic Desktop, which operate on semantically enriched resources. For this reason we propose the use of techniques from associative information retrieval to find additional relevant material, even if no semantic information is provided for those resources.

We describe our approach to information retrieval on the Semantic Desktop and present a retrieval component developed during the first year of the APOSDLE[1] project. The rest of this paper is organized as follows: in section 2 we introduce the concept of associative information retrieval, in section 3 we present statistical information about the knowledge base used in APOSDLE to

[1] http://www.aposdle.org/

demonstrate why the use of associative retrieval techniques suits our needs. Section 4 describes our retrieval system and section 5 its evaluation. We present related work in section 6 and our conclusion in section 7.

2 Associative Information Retrieval

Crestani [5] understands *associative retrieval* as a form of information retrieval which tries to find relevant information by retrieving information that is by some means *associated* with information that is already known to be relevant. Information items which are associated can be documents, parts of documents, extracted terms, concepts, etc. The idea of associative retrieval dates back to the 1960s, when researches [14, 15] in the field of information retrieval tried to increase retrieval performance using associations between documents or index terms, which were determined in advance.

Association of information is frequently modeled as graph, which is referred to as *associative network* [5]. Nodes in this network represent information items such as documents, terms or concepts. Edges represent associations between information items and can be weighted and / or labeled, expressing the degree and type of association between two information items, respectively.

3 Statistics about the Knowledge Base used

We operate on the knowledge base created for the first prototype of the APOSDLE system. The goal of the present system is to help knowledge-workers understand the field of requirements engineering. Therefore this domain is modeled using an ontology. Documents containing learning material (definitions, examples, tutorials, etc.) about requirements engineering are partly annotated with concepts from the domain ontology. The ontology consists of 70 concepts, 21 concepts are used to annotate documents. The document base consists of 1016 documents, 496 documents are annotated with one or more concepts from the knowledge base. We experience a typical scenario here: only parts of the ontology are used for annotation and only parts of the documents are annotated. We see this setting corresponding to the coverage problematic presented in section 1 and employing associative retrieval techniques appropriate to finding relevant material that was not originally annotated with concepts from the domain ontology.

4 The Prototype

The system presented here is based on an associative network consisting of two interconnected layers, one for concepts and one for documents. Nodes in the concepts layer correspond to concepts in the domain ontology. Nodes in the document layer correspond to documents in the system. Concept nodes are associated by means of semantic similarity (cf. section 4.1)), document nodes are associated by means of textual similarity (cf. section 4.2). Concept nodes are associated with document nodes if the concept is used to annotate the document (cf. sections 4.3 and 4.4). The network is searched using a spreading activation algorithm (cf. section 4.5).

4.1 Calculating Semantic Similarity of Concepts

For calculating the similarity of two ontological concepts a symmetric semantic similarity measure is used. The method was presented in [18] and requires two concepts belonging to the same ontology as input. It calculates the semantic similarity between these two concepts according to equation 1. This similarity measure builds on the path length to the root node from the least common subsumer (lcs) of the two concepts, which is the most specific concept they share as an ancestor. This value is scaled by the sum of the path lengths from the individual concepts to the root.

$$sim(c_1, c_2) = \frac{2 \cdot lcs(c_1, c_2)}{depth(c_1) + depth(c_2)} \tag{1}$$

With:

- c_1 ... first concept
- c_2 ... second concept
- *lcs* ... least common subsumer of two concepts
- *depth* ... depth of concept in the class hierarchy

Depending on the features present in an ontology different similarity measures qualify to be applied. We chose the measure presented in [18], as a prominent feature of our ontology are taxonomic relations between concepts. An advantage of the used measure is that it tries to address one of the typical problems of taxonomy-based approaches to similarity: relations in the taxonomy do not always represent a uniform (semantic) distance. The more specific the hierarchy becomes, the more similar a child node is to its father node in the taxonomy.

4.2 Calculating Text-based Similarity of Documents

As similarity measure for text-documents we use an asymmetric measure based on the vector space model implemented in the open-source search-engine Lucene[2]. The similarity between two documents is calculated as shown in equation 2.

$$sim(d1, d2) = score(d1_{25}, d2) \tag{2}$$

With:

- $d1$ ... document vector of the first document
- $d2$ ... document vector of the second document
- $d1_{25}$ ... document vector of the first document with all term weights removed except the 25 highest terms weights

$d1_{25}$ is used as query vector for the *score*-measure of Lucene. For extracting the 25 terms with the highest weights, both the document content and the document title are taken into account. The calculation of Lucene's score is depicted in equation **??**. A detailed explanation of the various parameters that can be used to adapt the behavior of Lucene can be found in the Javadoc of the `org.apache.lucene.search.Similarity` class.

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

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t\_in\_q} (tf(t\_in\_d) \cdot idf(t)^2 \cdot t.getBoost() \cdot norm(t,d)) \quad (3)$$

With:

- q ... query vector
- d ... document vector
- $coord(q,d) = numberOfMatchingTerms/numberOfQueryTerms$
- $numberOfMatchingTerms$ ... number of terms in document matching query
- $numberOfQueryTerms$ ... number of terms in the query
- $queryNorm(q)$ ... normalization of the query vector, Lucene default used
- $tf(t\_in\_d)$ ... term frequency of current term in document, Lucene default used
- $idf(t)$ ... inverse document frequency of current term in the document collection, Lucene default used
- $t.getBoost() = tf(t\_in\_q) \cdot idf(t)$
- $tf(t\_in\_q)$ ... term frequency of current term in query
- $norm(t,d) = 1/sqrt(numberOfDocumentTerms)$
- $numberOfDocumentTerms$ ... number of terms in the current document

Finding similar documents to a document based on the vector space model is a well researched topic. Equally, Lucene is a frequently used text search engine. Therefore we are confident of the applicability of both, the similarity measure as well as the search engine to our scenario.

4.3 Semantic Annotation of Documents

Despite to other approaches, where fine-grained annotation of the words present in a document with concepts from the ontology is carried out (c.f. [3] or [8]), we follow a more pragmatic approach. We adopt the tagging metaphor and blend this approach with the controlled vocabulary of an ontology. This means that we tag whole documents with a set of concepts the document *deals with*.

We follow this approach two reasons: (1) Although the complete semantics of a sentence contained in a document are not recognized using this approach, the additional information added to the document still provides opportunities to be used at a later time in retrieving material [17] by a limited amount of human involvement. (2) We think that for the near future it makes sense to work on making the Semantic Web a reality, by focusing on bringing little semantics [7] into the current web and taking small steps. We follow this pragmatic approach and try to apply it to the Semantic Desktop in the context of our work.

In APOSDLE, annotation of documents is supported by a plug-in for the ontology editor Protégé[3] and supported by a classification algorithm. This means that we have implemented functionality that suggests a set of concepts for documents to be annotated based on the set of documents already annotated. A detailed description of our approach and an early realization of it can be found in [16].

[3] `http://protege.stanford.edu/`

4.4 Weighting the Annotations

In our (and other) approach(es) to semantic annotation, a document is either annotated with certain concepts or it is not. From a retrieval point of view this means that a document is either retrieved, if it is annotated with a concept present in the query, or it is not retrieved, if none of the concepts in the query are assigned to the document. Ranking the retrieved document set is impossible.

To allow for ranking the result set and increase the performance of our system we weight the annotations between documents and concepts using a tf-idf-based weighting scheme. This is a standard instrument in information retrieval to improve retrieval results [11]. Our weighing approach is related to the one presented by [3], who are also weighting semantic annotations using a tf-idf-based measure.

$$weight(c,d) = tf(c,d) \cdot idf(c) = tf(c,d) \cdot \log \frac{D}{a(c)} \tag{4}$$

With:

- c ... a concept
- d ... a document
- $tf(c,d)$ ... 1 if d is annotated with c, 0 otherwise
- $idf(c)$ ... inverse document frequency of concept c
- D ... total number of documents
- $a(c)$ ... number of documents annotated with concept c

4.5 Searching the Network

The network structure underlying the system is searched by spreading activation. Starting from a set of initially activated nodes in the network, activation spreads over the network and activates nodes associated with the initial set of nodes. Originally stemming from the field of cognitive psychology, where it serves as a model for operations in the human mind, spreading activation found its way over applications in both neural and semantic networks to information retrieval [5]. It is comparable to other retrieval techniques regarding its performance [9].

Besides systems that use spreading activation for finding similarities between text documents or search terms and text documents, approaches exist, which employ spreading activation for finding similar concepts in knowledge representations [1] [12]. The novelty of our approach lies in combing spreading activation search in a document collection with spreading activation search in a knowledge representation. The formula we use to calculate the spread of activation in our network is depicted in equation 5.

$$A(n_j) = \frac{\sum_{t=1}^{t} A(n_i) \cdot w_{i,j}}{\sum_{t=1}^{t} w_{i,j}} \tag{5}$$

With:

- $A(n_j)$ ... activation of node n_j
- $A(n_i)$ ... activation of node n_i
- t ... number of nodes adjacent to node n_j
- $w_{i,j}$ ... weight of edge between node n_i and node n_j

Search in our network is performed as follows:

1. Search starts with a set of concepts, representing the information need of the knowledge-worker. The concept nodes representing these concepts are activated.
[2. *Optionally*, activation spreads from the set of initially activated concepts over the edges created by semantic similarity to other concepts nodes in the network.]
3. Activation spreads from the currently activated set of concept nodes to the document nodes over the edges created by semantic annotation to find documents that deal with the concepts representing the information need.
[4. *Optionally*, activation spreads from the documents nodes currently activated to document nodes that are related by means of textual similarity and are therefore associated with the document nodes.]
5. Those documents corresponding to the finally activated set of document nodes are returned as search result to the user.

5 Evaluation

The present approach to retrieval on the Semantic Desktop is different from current attempts to retrieval in a desktop environment: (1) the semantic information present in an ontology is taken into account for retrieval purpose; (2) the query to the retrieval system is formulated by a set of concepts stemming from an ontology as opposed to a set of terms (words) as typically used in the context of desktop search. As we are not aware of any standard test corpora for the evaluation of an information retrieval system for the Semantic Desktop we have created our own evaluation environment.

We have evaluated six different configurations of our system using a set of 22 queries. For every query we relevance-judged the first 30 search results. Afterwards we calculated precision at rank 10 (P(10) cf. [2]), precision at rank 20 (P(20))[4] and inferred average precision (infAP[5]). All three evaluation measures rank the tested system configurations in the same order.

Table 5 shows the ranking of the different system configurations. The columns *SemSim*, *TxtSim* indicate whether semantic similarity or text-based similarity was used for the search. The last line (configuration 6) of table 5 is the baseline configuration of our system. The results delivered by this configuration are comparable to the use of a query language as SPARQL combined with an idf-based ranking and no associative retrieval techniques used. Exactly those documents are retrieved that are annotated with the concepts present in the query. All associative search approaches employing semantic similarity (configurations 1, 2, and 5), text-based similarity (configurations 1, 2 and 3) or both (configurations 1, 2, 3, 4 and 5) increase retrieval performance compared to the baseline config-

[4] As we judged 30 documents for every query it would also have been possible the calculate P(30) but as we are aiming on presenting our search results using a sidebar-based interface and we have limited space for our search results there, we are not considering to present 30 results.

[5] infAP was proposed by [19] and performs a random sampling approach to all *judged* results (relevant and not relevant) for a query. The measures P(10) or P(20) only consider judged, relevant results. As infAP takes more information into account than P(10) or P(20) it is considered as a more stable measure.

uration. Additional relevant documents are found, which are not annotated with the concepts used to query the system.

Configuration	SemSim	TxtSim	P(10)	P(20)	infAP
1	Yes (> 0.5)	Yes	0.7	0.6523	0.5728
2	Yes (> 0.7)	Yes	0.6909	0.6477	0.5706
3	No	Yes	0.6636	0.6227	0.5431
4	Yes (> 0.5)	No	0.6545	0.5818	0.4971
5	Yes (> 0.7)	No	0.6364	0.5727	0.46
6	No	No	0.6045	0.5545	0.4176

Table 1: Ranking of system configurations using P(10), P(20) and infAP

For calculating the evaluation scores we have used the trec_eval[6] package, which origins from the Text REtrieval Conference (TREC) and allows for calculating a large number of standard measures for information retrieval system evaluation.

6 Related Work

Beagle++ [4] is a search engine for the Semantic Desktop and indexes RDF-metadata together with document content. Both [3] and [8] present an extension of the vector space model. Together with document content they index semantic annotations of documents and use this information for search. All three are very promising approaches that extend the vector space model using semantic information. None of them employs measures of semantic association.

[12] present a hybrid approach for searching the (semantic) web, they combine keyword based search and spreading activation search in an ontology for search on websites. Ontocopi [1] identifies communities of practice in an ontology using spreading activation based clustering. Both are prospective approaches employing ontology-based measures of association and evaluating them using spreading activation. They do not integrate text-based measures of association into their systems.

7 Conclusions and Future Work

Our experiments encourage us, that the application of associative retrieval techniques to information retrieval on the Semantic Desktop is an adequate strategy. Following recent works [2] [19] in information retrieval system evaluation we are confirmed that the amount of relevance judgments we have used should be increased to have a higher confidence in our retrieval system evaluation. We tend to conclude that text-based methods for associative retrieval result in a higher increase in retrieval performance, therefore we want to explore the approach of attaching a set of terms to every concepts in our domain ontology during

[6] `http://trec.nist.gov/trec_eval/`

modeling time to provide search results even for concept that are not used for annotation. In addition we want to extend our research towards evaluating different semantic similarity measures.

Acknowledgments

We thank the anonymous reviewers of our submission for their constructive feedback.

This work has been partially funded under grant 027023 in the IST work programme of the European Community. The Know-Center is funded by the Austrian Competence Center program Kplus under the auspices of the Austrian Ministry of Transport, Innovation and Technology (www.ffg.at/index.php?cid=95) and by the State of Styria.

References

1. H. Alani, S. Dasmahapatra, K. O'Hara, and N. Shadbolt. Identifying communities of practice through ontology network analysis. *IEEE Intell. Syst.*, 18(2):18–25, 2003.
2. C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004.
3. P. Castells, M. Fernndez, and D. Vallet. An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Trans. Knowl. Data Eng.*, 19:261–272, 2007.
4. P.-A. Chirita, S. Costache, W. Nejdl, and R. Paiu. Beagle++: Semantically enhanced searching and ranking on the desktop. In *The Semantic Web: Research and Applications*, 2006.
5. F. Crestani. Application of spreading activation techniques in information retrieval. *Artif. Intell. Rev.*, 11:453–482, 1997.
6. J. Heflin and J. Hendler. Searching the web with SHOE. In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01.*, 2000.
7. J. Hendler. The dark side of the semantic web. *IEEE Intell. Syst.*, 22:2–4, 2007.
8. A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Journal of Web Semantics*, 2:49–79, 2004.
9. T. Mandl. *Tolerantes Information Retrieval. Neuronale Netze zur Erhöhung der Adaptivität und Flexibilität bei der Informationssuche.* PhD thesis, University Of Hildesheim, 2001.
10. R. McCool. Rethinking the semantic web, part 1. *IEEE Internet Comput.*, 9(6):88–87, 2005.
11. S.E. Robertson and K. Spärck Jones. Simple, proven approaches to text retrieval. Technical report, University of Cambridge, Computer Laboratory, 1994.
12. C. Rocha, D. Schwabe, and M. P. de Aragão. A hybrid approach for searching in the semantic web. In *Proceedings of the 13th international conference on World Wide Web, WWW 2004*, 2004.
13. M. Sabou, M. d'Aquin, and E. Motta. Using the semantic web as background knowledge for ontology mapping. In *International Workshop on Ontology Matching (OM-2006)*, 2006.
14. G. Salton. Associative document retrieval techniques using bibliographic information. *JACM*, 10:440–457, 1963.
15. G. Salton. *Automatic Information Organization and Retrieval.* McGraw Hill, 1968.
16. P. Scheir, P. Hofmair, M. Granitzer, and S. N. Lindstaedt. The ontologymapper plug-in: Supporting semantic annotation of text-documents by classification. In *Proceedings of the SEMANTICS 2006*, 2006.
17. K. Spärck Jones. What's new about the semantic web?: some questions. *SIGIR Forum*, 38:18–23, 2004.
18. Z. Wu and M. S. Palmer. Verb semantics and lexical selection. In *Meeting of the Association for Computational Linguistics (ACL)*, pages 133–138, 1994.
19. E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgments. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, 2006.