International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:11, 2007

A Hybrid Ontology Based Approach for Ranking Documents

Sarah Motiee, Azadeh Nematzadeh, Mehrnoush Shamsfard

Abstract**—** Increasing growth of information volume in the internet causes an increasing need to develop new (semi)automatic methods for retrieval of documents and ranking them according to their relevance to the user query. In this paper, after a brief review on ranking models, a new ontology based approach for ranking HTML documents is proposed and evaluated in various circumstances. Our approach is a combination of conceptual, statistical and linguistic methods. This combination reserves the precision of ranking without loosing the speed. Our approach exploits natural language processing techniques to extract phrases from documents and the query and doing stemming on words. Then an ontology based conceptual method will be used to annotate documents and expand the query. To expand a query the spread activation algorithm is improved so that the expansion can be done flexible and in various aspects. The annotated documents and the expanded query will be processed to compute the relevance degree exploiting statistical methods. The outstanding features of our approach are (1) combining conceptual, statistical and linguistic features of documents, (2) expanding the query with its related concepts before comparing to documents, (3) extracting and using both words and phrases to compute relevance degree, (4) improving the spread activation algorithm to do the expansion based on weighted combination of different conceptual relationships and (5) allowing variable document vector dimensions. A ranking system called ORank is developed to implement and test the proposed model. The test results will be included at the end of the paper.

Keywords**—** Document ranking, Ontology, Spread activation algorithm, Annotation.

II. INTRODUCTION

As the Internet grows, finding documents that are relevant to user query becomes increasingly hard. The main reason is that the semantic of documents is not recognized correctly and users do not describe their information needs clearly. So we need efficient systems for responding the user information need in a short time and with high precision. Most available information retrieval systems provide the access to different kinds of information, but their precision is low.

Manuscript received January, 25, 2005.

Sarah Motiee has received her B.Sc. in computer engineering from Shahid Beheshti University and now is a M.Sc. student in AmirKabir University of technology, Tehran, Iran. (Corresponding author to provide phone: 98-21-88608835; fax: 98-21-44069921; e-mail: sr_mt79@yahoo.com).

Azadeh Nematzadeh has received her B.Sc. in computer engineering from Shahid Beheshti University and now is a M.Sc. student in AmirKabir University of technology, Tehran, Iran. (e-mail: azadeh_nematzadeh@yahoo.com).

M. Shamsfard is an assistant professor at Electrical and Computer Engineering Department, Shahid Beheshti University, Tehran, Iran. (e-mail: shams@sepehrs.com).

To answer a user query a long list of documents is usually generated. But the user examines the first ten to twenty of them. So an algorithm for ranking documents according to their relevance to user query is used. A ranking algorithm calculates the similarity degree of each document to user query. Then documents are sorted by this degree and will be presented to the user. Ranking algorithms exploit different information to estimate the similarity degree. According to the information these algorithms use, we classify document ranking models into four classes: Boolean, statistical and probabilistic, hyper link based and conceptual models.

Boolean Model [1] has been used in the simplest form of retrieving documents according to relevancy to the user query. In this Model user query is a weightless phrase and its evaluating consequence in each document only indicates their relevancy and the document's rank will not be computed. To make ranking possible Extended Boolean model [2] was introduced. In this model the weights were assigned to the document and query's words and extended Boolean operators compute similarity measure. One of the most popular extended Boolean models is p – norm [3].

Statistical model is one of the most common and oldest models for document ranking, which uses a list of terms for representing document and query. Principally representing methods in this model does not mention any conceptual relation among terms. This model exploits statistical information such as term's frequency, document length, etc to compute similarity degree of document and query. Vector space model [4] and an alternative form of it called LSI[1] [5], are two known models in this category. Probabilistic model [6] applies probability theory for ranking documents and uses variant methods for representing document and query. Relevance Models [7] and Inference Models [7] are two kinds of probabilistic model.

Hyperlink Based models are another ranking models which use hyperlink structures. As linked based models use the content of other pages to rank the current page, it will not be interfered by the user. These models may be query-dependent such as HITS[2] [8] or query-independent such as PageRank [9] and WLRank[3] [10] or a combination of both such as SALSA[4] [11].

[1] Latent Semantic Indexing
[2] Hyperlink-Induced Topic Search
[3] Weighted Links Rank
[4] Stochastic Approach for Link Structure Analysis

Conceptual Models try to extract the concepts of the documents and the query to compare them. As the statistical algorithms do not consider the semantics of the document, they are not precise. Conceptual approaches map a set of words and phrases to concepts. They exploit conceptual structures for representing concepts. Ontology based model [12], Conceptual Dependency between Terms [13] and spread activation (SA) algorithm [14] are introduced in this category.
As mentioned above, there are variant of document ranking models and each of them has its benefits and awkward. It is clear that none of the above models can provide all the aspects of the user needs. Therefore, we introduce a hybrid ontology based approach by gaining the benefits of individual models and reducing their awkward that its main features are:

- Combination of conceptual, statistical and linguistic features of documents
- Improvement of the spread activation algorithm to do the expansion based on weighted combination of different conceptual relationships
- Expansion of query with its related concepts
- Considering the phrases of the query instead of its words
- Variable documents vector dimensions

In the next section our proposed model is introduced and the results of its evaluation are presented.

II. THE PROPOSED RANKING MODEL

Two main goals of information retrieval are precision and speed. Precision is related to the relevancy of retrieved documents according to the user query. Current statistical models reach the second goal but they are not precise enough because of ignoring linguistics features and semantic of query and document. On the other hand, the conceptual and language models are usually complicated. Although they are more precise than statistical models, they are not fast enough. The proposed ranking model trades off relevancy and speed by using a combination of conceptual, statistical, and language processing techniques to rank documents according to their relevancy to user's query.
Our Model is based on ontology and is evaluated by developing a ranking system called ORank. Figure (1) shows the structure of ORank and its basic components. As the figure shows, the main functional modules are (1) the document processor, (2) the ontology processor, (3) the query processor and (4) the ranker. They use a general ontology and a data base containing ontology and documents information.

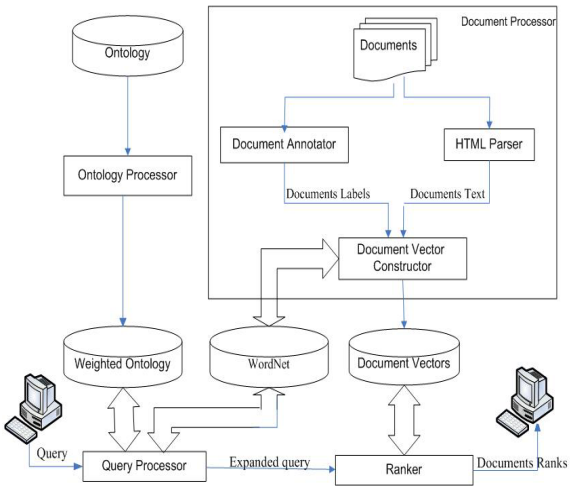

Fig.1 The structure of ORank system

The document processor creates a vector for each input HTML document. To do this, documents should be annotated in which their words and phrases are mapped to their corresponding conceptual instances using an ontology. The dimensions of the created vectors have variable sizes and correspond to labels produced by the annotation module. They are weighted by statistical methods considering higher priorities for phrases rather than single words.
The ontology processor weights the relations in the reference ontology. The weighted ontology will be used by the query processor.
The query processor, first extracts the query phrases, and then applies weighted ontology to expand these phrases with their related concepts. For this purpose, we have created improved SA algorithm in which the expansion can be done based on a weighted combination of some arbitrary conceptual relations. In addition, we use stemming technique in both document and query processing modules. After expansion, the query vector will be built in which each dimension corresponds to a query phrase or its expansion (instead of query words which are used by vector space model).
At last the ranker calculates the rank of each document according to its relevancy to the expanded user query.
In this section, we will describe the system's functionality in more details.

A. The Document Processor

The document processor receives the document set as input. It applies statistical and conceptual techniques for building the documents vectors. Conceptual document processing is accomplished by ontology-based annotation. The purpose of document annotation is using the semantic of the document in addition to its statistical characteristics. Ontology-based annotator uses information extraction techniques for mapping the document's words and phrases to ontology's concepts. ORank annotates document in a semi automatic way. It uses

an online tool, AeroSwarm [15] for automatic document annotation. As this annotation may not be sufficient for expressing the document concepts, the system administrator can add the required labels too. Computing relevancy degree between query and document is similar to vector space model, so it is necessary to build vector for each document. It is noticeable that in our model (in opposition to the vector space model), the length of document vectors and also their dimensions are different for different documents.

To build the vectors we should first extract the text part of the document, so we adapted the HTML parser [16] for analyzing the HTML document, removing HTML tags and extracting the text part of it. The output of the HTML parser will be fed to the stemming module for replacing text's words by their root forms. Document's labels could be phrases therefore the system considers document's phrases instead of just the words.

To extract appropriate phrases (labels) and count their frequency in the document the following algorithm is used. In this algorithm, the number of words in phrases of each document is equal to the maximum length of document labels i.e. length of a label which contains maximum number of words. Following algorithm is used for counting labels of document:

1. Variable i is set to be 1.
2. Variable l is set to be maximum length of document labels in the current document.
3. from word i to the end of the document do
    a. The words between word i to word i+l are considered.
    b. All phrases in this range are extracted. (All permutation of length one to l)
    c. Each phrase is searched in document labels.
    d. Frequency of the longest phrase which exists in document labels is increased by one.
    e. Variable i is increased by one.

In ORank each document's vector is stored in a table in the database. This table contains the labels and their weights. Label's weight indicates its relevancy degree to the document concept. These weights are calculated by statistical processing of the documents as shown in equation (1). In this calculation document's phrases have higher priority than single words.

$$W_{i,j} = \frac{freq_{i,j}}{\max_k freq_{k,j}} \times \log \frac{N}{n_i} \quad (1)$$

$freq_{i,j}$ is the frequency of each label (i) in document (j), N is the number of documents in the collection, n_i is the number of documents containing the label (i) and $\max_k freq_{k,j}$ is the frequency of the label with maximum frequency in document (j).

After weight calculation, vector length should be computed and stored in the database. Length of the vector is the square root of the sum of the squares of all its components.

B. The Query Processor

In order to compute the relevancy degree of a document to the user query, it is required to covert the query into a representation which is comparable with document representation. As it is said, in ORank each document is represented by a vector. Therefore a vector should be built for the query. Since the user query might not show his information needs obviously, a new approach is suggested for expanding query in this system. Two main parts of this approach are phrase extraction and flexile expansion of phrases based on various conceptual relationships in the ontology.

A. Phrases extraction- User query words are stemmed and all their possible phrases (all ordered combinations of input words) are generated by considering word order. Then phrases will be selected among others which (1) exist in the ontology or (2) occur in the document set labels. In this way the query will not be limited to a set of words and both words and phrases are used as input for expansion algorithm. But for gaining better precision higher weights are assigned to phrases.

B. Words and phrases expansion- In this step user query words and phrases are expanded with their related concepts using the improved version of SA Algorithm. In this new algorithm expansion is flexible and the expansion relation i.e. the relation that expansion is done through it, can be selected. In other words expanding concepts can be done in various dimensions. This expansion relation can be any of or a weighted combination of various individual relations such as hyponymy or hyperonymy (taxonomic relations in both directions), synonymy, mereonymy (part of), identity, etc. The weights which show the importance of relations can be defined by user or computed by the system by means of relevance feedback methods.

The improved SA algorithm receives a set of concepts containing the user query words and phrases as input. It also receives the activation value of these concepts. Then it moves in a weighted ontology (its construction procedure is mentioned in 3-3) through the expansion relation(s) to extract the related concepts and update the activation values.

In ORank, extracted phrases are considered as an initial concept set which their activation values are set to be one. Activation values of other concepts in the ontology are zero.

Then, the ontology is traversed over the selected relationships to produce the related concepts to initial concept set. Ontology traversal can be done in multi levels. The activation value of these concepts is calculated by equation (2):

$$I_j = \sum_{\forall k \in R} I_{j,k} \times \alpha_k \quad (2)$$

In this equation α_k is the weight of relation k and $I_{j,k}$ is the activation value of concept j which is obtained through the relation k. This activation value is calculated by equation (3):

$$I_{j,k} = \sum_{i \in C} W_{i,j} \times I_{i,k} \quad (3)$$

In this equation, C is a concept set, $I_{i,k}$ is an activation value of concept i, $I_{j,k}$ is an activation value of j^{th} related concept to concept i and $W_{i,j}$ is the weight of relation between these two concepts.

To clarify the subject, the ontology traversal procedure and activation value calculation for hybrid relation *ISA* which is the combination of parent and child relations (hyperonymy and hyponymy) is shown as an example in the following:

I. for each traverse level, following steps are done:
   1. A concept is selected from concept set if it was not selected before. The concept set initially contains query words and extracted phrases, which expanded concepts will be added to it.
   2. Parents of current concept are obtained.
   3. the activation value of each parent will be calculated by formula (4)

$$I_{j,p} = I_{j,p} + W_{i,j} \times I_{i,p} \quad (4)$$

   In this formula $I_{i,p}$ is the activation value of current concept. $I_{j,p}$ is the activation value of its parent and $W_{i,j}$ is the weight of the relation between these two concepts.
   4. The parent and its activation value are added to the concept set. If this parent exists in the concept set, the parent with the maximum value is selected and will be added to the concept set.
   5. Step 1 is repeated.

II. Phase I is repeated for initial concept set children.

III. Two concept sets obtained in phases I and II are merged together. As these sets may have intersections, the common concepts activation values are calculated by formula (5).

$$I_j = \alpha_p \times I_{j,p} + \alpha_s \times I_{j,s} \quad (5)$$

In this formula $I_{j,p}$ and $I_{j,s}$ are the activation values obtained from parent and child relations. α_p and α_s are weights assigned to these relations.

The concept set which is obtained by above algorithm, consists the expanded query.

The query vector length is the square root of the sum of the squares of activation values of query concepts.

C. The Ontology Processor

Ontology processor is responsible to assign weight to ontology's links. This weighted ontology will be used in query processing.

In our proposed model ontology link's weight is computed by multiplication of similarity measure and specificity measure. Similarity measure indicates the similarity between two related concept instances *Cj* and *Ck* and is computed by formula (6). The idea behind this measure is that two concepts will be similar if they are related to same concepts.

$$W(c_j, c_k) = \frac{\sum_{i=1}^{n} n_{i,j,k}}{\sum_{i=1}^{n} n_{i,j}} \quad (6)$$

$\sum_{i=1}^{n} n_{i,j}$ is the number of related concepts to concept instance c_j and $\sum_{i=1}^{n} n_{i,j,k}$ is the number of related concepts to both concept instances c_j and c_k .

For instance if the chosen relation is hyponymy, in order to calculate the similarity measure of two related concept instances c_j and c_k, sum of their common fathers and common sons is divided to sum of c_j 's fathers and sons.

The specificity measure indicates how much a relation is specific. Formula (7) is used to calculate the specificity measure. A relation will be more specific if its destination concept is related to few concepts.

$$W(c_j, c_k) = \frac{1}{\sqrt{n_k}} \quad (7)$$

In this formula, n_k is the number of relations which their destination concept is c_k.

In ORank input degree of each destination concept instance (n_k) will be computed and its inverse square will be assigned to relation between destination concept instance (e.g. father) and source concept instances (e.g. child).

All of ontology's relations and their weights are stored in system data base. So the ontology is ready to be used in query processor.

D. Ranker

The relevance degree between a query and a document defines the document's rank in the list of retrieved documents for this query. In order to calculate this relevancy degree, ranker divides the internal product of document and query vectors to the product of these vector lengths. Then documents are sorted according to their ranks and will be presented to the user.

III. Experimental Results

In order to test and evaluate ORank we applied precision and recall measures. First we prepared a collection of two hundred HTML documents with various topics, a collection of ten queries, a collection of relevant documents for each query and two ontologies. The ontologies can be selected via an interface. The selected ontologies to test ORank were Cyc and WordNet. The queries were chosen in a way that comprises both worst and best cases.

The documents were processed once and their vectors were stored in a system data base while it was possible to add new documents to this collection. Therefore while the user query is entered, the only required processes are query processing and computing its similarity degree with each document.

We performed many tests, in order to evaluate how our new ranking model's features effect on relevancy degree of retrieved documents. Test results are shown in the diagrams

shown in figures 2-4. The diagrams have been drawn for average cases.

A. Improved SA algorithm

To analyze the efficiency of using improved SA algorithm instead of keyword based search, we tested ORank in two modes: 1- In our improved algorithm, the query's keywords and phrases are spread using hyponym relation of CYC ontology. 2- The documents are ranked using keyword search and no expansion is done. As figure2 shows applying improved SA algorithm has increased the precision of retrieved documents relative to their recall.

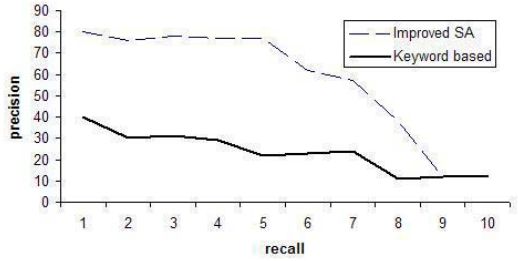

Fig. 2 Comparing improved SA with keyword search

B. Ontology

As we mentioned earlier, ontology based models are so dependent on the ontology they use. To show this fact, we expanded the user's query using hyponym relation of WordNet and CYC ontologies separately. We gain better results with CYC ontology as shown in figure3.

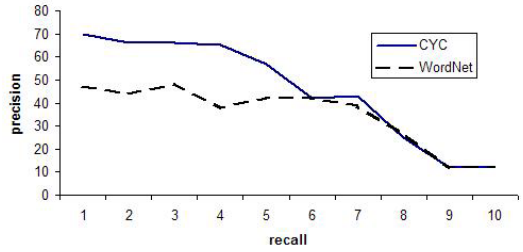

Fig. 3 Comparing CYC with WordNet

C. Hybrid Relation

We claimed that by considering different ontology's relation for expanding user query, better results would be obtained. To prove this claim, we expanded the query using hyponym, synonym and both on WordNet Ontology. The results are shown in figure4.

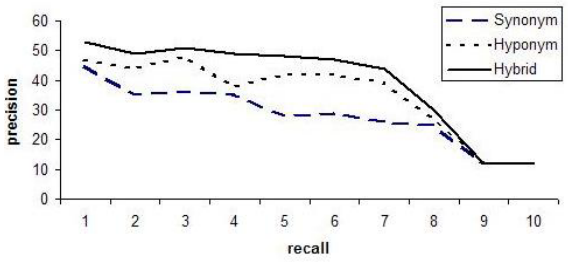

Fig. 4 Comparing hybrid relation with synonym and hyponym relations

IV. Conclusion

In this paper, we introduced an ontology-based model for ranking documents according to relevancy to user's query. In this model the precision of existing statistical models is improved using concept instances instead of words in document and query's vectors. Another salient point in this method is extracting the query and document's phrases instead of their single words which are stemmed too. In addition, we improved SA algorithm to expand query's keywords and phrases to their related concepts instance using various relation types of an arbitrary ontology. So the relevance degree of retrieved document would be increased.

To complete this effort following improvements are proposed as further works:

- Ontology's classes usually have comment property. During query expansion, it is possible to search query's phrases and words in comment property. If they were found, the query would have expanded by that class.
- Document annotation and query expansion accomplish by applying ontology. So using special purpose ontology would have great effect in test results.
- Appropriateness of annotation algorithm of automatic annotating tool increases the relevancy of retrieved documents.
- More precise approximation of increasing degree of phrase's weight coefficient relative to word's requires more tests.
- For computing weigh coefficient of relation in improved SA algorithm, it is possible to implement a system based on relevance feedback.

References

[1] E. Greengrass, "Information Retrieval: A survey". DOD Technical Report TR-R52-008-001, November 2000.
[2] G. Salton, E. A.Fox, H. Wu, "Extended boolean information retrieval", Communications of the ACM, Volume 26, No. 11, 1983, Pages: 1022-1036.
[3] J.H. Lee, "Properties of extended boolean models in information retrieval". Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1994, Pages: 182-190.
[4] D. L. Lee, H. Chuang, K. Seamons, "Document ranking and the Vector-Space model". IEEE Software, Volume 14, Issue 2, March 1997, Pages: 67 – 75.

[5] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, "Indexing by latent semantic analysis", Journal of the American Society for Information Science, Volume 41, Issue 6, 1990, Pages: 391-407.
[6] M. E. Maron, J. L. Kuhns, "On relevance, probabilistic indexing and retrieval".Journal of the ACM, Volume 7, 1960, Pages: 216-244.
[7] F. Crestani, M. Lalmas, J. van Rijsbergen, L. Campbell, "Is this document relevant? …probably. A survey of probabilistic models in information retrieval". ACM Computing Surveys, Volume 30, Issue 4, December 1998, Pages: 528 – 552.
[8] M. R. Henzinger, "Hyperlink analysis for the web". IEEE Internet Computing, Volume 5, Issue 1, January 2001, Pages: 45 – 50.
[9] S. Brin, L. Page, "The anatomy of a Large-Scale Hyper-textual web search engine". Proceedings of the Seventh International World Wide Web Conference, Elsevier Science, New York, 1998, Pages: 107-117.
[10] R. Baeza-Yates, E. Davis, "Web page ranking using link attributes". International World Wide Web Conference, Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, New York, NY, USA, 2004, Pages: 328 – 329.
[11] R. Lempel, S. Moran. "The stochastic approach for link-structure analysis (SALSA) and the TKC e®ect". In The Ninth International WWW Conference, May 2000.
[12] D. Vallet, M. Fernández, P. Castells, "An Ontology-Based information retrieval model". 2nd European Semantic Web Conference (ESWC 2005). Heraklion, Greece, May 2005. Springer Verlag Lecture Notes in Computer Science, Volume 3532. Gómez-Pérez,A.;Euzenat,J.(Eds.),2005, Pages:455-470.
[13] M. Nakashima,Y. Kaneko, T. Ito, "Ranking of documents by measures considering conceptual dependence between terms". Systems and Computers in Japan, Volume 34, Issue 5 , 2003, Pages: 81 – 91.
[14] C. Rocha, D. Schwabe, M. Poggi de Aragão, "A hybrid approach for searching in the semantic web". International World Wide Web Conference, Proceedings of the 13th international conference on World Wide Web, 2004, Pages: 374 - 383.
[15] Aeroswarm,http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html
[16] LCNetTools,http://itlang/vb.net/archivio.asp?subMenu=Tutte&FullTexton&TypeRi=AND&keyword=LCNettools