World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:3, 2008

ORank: An Ontology Based System for Ranking Documents

Mehrnoush Shamsfard, Azadeh Nematzadeh, and Sarah Motiee

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 for extracting phrases and stemming 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 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.

I. INTRODUCTION

AS the Internet grows, finding documents that are relevant to the user query becomes increasingly hard. The main reason is that the semantic of documents is not recognized correctly and users do not express their information needs clearly. Therefore, we need efficient systems for responding the user information need in a short time and with high precision. An information retrieval system can be described as a collection of documents, a collection of queries and mechanisms for determining the relevance degree of documents with queries. Most available information retrieval systems provide the access to different kinds of information, but their precision is low.

To answer a user query usually a long list of documents is generated. However, the user examines the first ten to twenty of them. So an algorithm for ranking documents according to their relevance to user query is needed. 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. Most conventional algorithms are keyword based and use statistical information such as term frequency, document length, etc. to calculate the relevance degree. By the creation of the web, ranking algorithms apply hyperlink structures too. As the statistical algorithms do not consider the semantics of the document, they are not precise enough. This problem has caused the conceptual approaches to appear in the recent years. These approaches try to extract and compare the concepts of the documents and the query.

There are variants of document ranking models, which are introduced in the next section. Each of these models has its advantages and drawbacks. It is clear that none of them can fully respond all the aspects of the user needs. Therefore, we introduce a hybrid ontology based approach, which has the advantages of individual models while reducing their drawbacks. The main features of our approach 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
- Allowing variable document vector dimensions

In the next section, document ranking models are reviewed briefly. Then the proposed model is introduced and the results of its evaluation are presented.

II. APPROACHES TO RANK DOCUMENTS

There are different models for ranking documents. These models receive query and a collection of documents as input and convert them to a non-textual representation. The procedure of this conversion for documents and query can be the same or different. By comparing these two representations,

Manuscript received April, 10, 2006.

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

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. (e-mail: sr_mt79@yahoo.com).

the documents that are more similar to the query gain a higher rank. According to the information that ranking models use, we classify them into four classes: Boolean, statistical and probabilistic, hyperlink based and conceptual models.

Boolean Model [1] is considered as the simplest form of retrieving documents according to relevancy to the user query. In this Model user query is a weightless phrase and evaluation of documents only indicates whether they are relevant to the query or not 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 both the document and query's words and Extended Boolean operators are used to 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 documents and queries. Principally representing methods in this model do not mention any conceptual relation among terms. This model exploits statistical information such as term frequency, document length, etc to compute similarity degree of document and the query. Vector space model [4] is a well-known model in this category. This model (presumably) after removing stop words and stemming, computes the term's weight by the $tf \times idf$ [1] formula. These weighted terms build the document vector and the query vector, which will be normalized then. Afterward, similarity degree of the document and the query is computed by using methods such as calculating the cosine of the angle between the two vectors, distance functions, Jaccard's coefficient and so on. Vector space model does not distinguish homonyms (similar words with different meanings) and synonyms (different words by similar meanings). An alternative form of vector space model is LSI[2] Model [5], which eliminates the drawback of vector space model by using statistical properties to extract term's conceptual relations. LSI reduces the vector dimensions into k-dimension space by using a matrix decomposition method called Singular Value Decomposition (SVD). Each dimension in this space is an extracted concept from the document collection, independent from the other concepts. The similarity degree is calculated in a way similar to vector space model.

Probabilistic model [6] applies probability theory for ranking documents and uses variant methods for representing the document and the query. Relevance Models [7], which are a kind of probabilistic model, apply Training Data Set to calculate terms weights. One of the simplest and most common Relevance Models is Binary Independence Retrieval model [8]. The other kind of probabilistic models are Inference Models [7], which use AI and logic concepts without any need to training data sets. Inference Models are classified into two main categories. The first one is based on the non-classical logic [9], and the second one is based on Bayesian inferences [10]. Inference Models use domain knowledge, user profile, user feedback about the relevancy degree, etc to compute similarity measures.

Hyperlink Based models use hyperlink structures for ranking. As the link based models use the content of other pages to rank the current page, they will not be interfered by users. These models may be query-dependent or query-independent. Query-dependent models such as HITS[3][11] build a query-specific graph called neighborhood graph by analyzing hyperlinks. They rank documents according to in-degree value of the graph nodes. Query-Independent models such as PageRank algorithm [12], assign a score to each page only once, independent of a specific user query, to measure the intrinsic quality of each page. At the query time, this score is applied to rank all documents matching the query. PageRank ranks documents by assigning weight to hyperlinks according to the quality of the page containing the hyperlink. WLRank[4] [13] is a variant of PageRank, which considers new attributes to give more weights to some links. SALSA[5] algorithm [14] uses the combination of ideas from both HITS and PageRank.

According to the idea of link based models, a model is introduced for ranking the Semantic-Linked Network [15], which contains different kinds of links. The rank of the documents will be an average of the ranks of all links.

Conceptual Models try to extract the concepts of the documents and the query to compare them. They map a set of words and phrases to concepts and exploit conceptual structures for representing them. Ontology based model [16] is one of the conceptual models which maps document's phrases into conceptual instances by using annotation. Then it assigns weights to these instances (for example according to a $tf \times idf$ formula. The weights indicate the relevancy degree of conceptual instances to document meaning. In ontology-based model, user query is converted to an internal representation such as RDQL[6], and executed on the knowledge base. This execution returns a list of relevant instances, such as conceptual tuples, that satisfy the query. Then vector space model can be used for computing documents ranks.

In another model based on Spread Activation (SA) algorithm [17], documents and their concepts are represented in a semantic network. For each network link, a weight is computed according to a measure such as similarity or specificity measure. In this model, an initial set of relevant documents to user query are expanded by executing spread activation (SA) algorithm. SA algorithm expands a set of initial components to their relevant components by exploring the semantic network.

Models based on the semantic network may represent the document terms in the form of semantic network. In these

[1] Inverted Document Frequency × Term Frequency
[2] Latent Semantic Indexing

[3] Hyperlink-Induced Topic Search
[4] Weighted Links Rank
[5] Stochastic Approach for Link Structure Analysis
[6] Resource Description Query Language

models [18], one of the R-distance [19] or K-distance [20] methods could compute semantic distance of the document and the query terms.

In another conceptual model, called Conceptual Dependency between Terms [21], the document and the query are represented by a covering over a conceptual hierarchy. Then a weight is assigned to each term of these coverings, which indicates the specificity degree of that term. In order to calculate term specificity, one of the following approaches has been taken: Absolute Specificity, Totally Relative Specificity, and Partially Relative Specificity. To compute covering ranks, the covering specificity, which is the sum of weights of covering terms, is used. These ranks are applied for document ranking.

There are other models for document ranking. Some of them such as language model [22] and relaxation algorithm [23] use natural language processing techniques. These models consider syntactic and semantic structure and morphological form of terms. Some other document ranking models such as Context Based Model [24] use information about the user.

Neural Networks [25], Genetic Algorithms [26], Fuzzy Sets [18], Relevance Feedback Models [27] could be used for increasing the performance of ranking models.

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

III. THE PROPOSED RANKING MODEL

Although precision and recall are two main goals of information retrieval, speed is important too. Current statistical models are fast but ignoring linguistics features and semantic of the query and document decreases their precision. 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 makes a trade off between the relevancy and speed by using a combination of conceptual, statistical, and language processing techniques to rank documents according to their relevancy to the user 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) document processor, (2) ontology processor, (3) query processor and (4) ranker. They use a general ontology and a database containing ontology and documents information.

The document processor creates a vector for each input HTML document. For this purpose, documents are annotated, which means that their words and phrases are mapped to their corresponding conceptual instances in ontology. The dimensions of the created vectors are variable and equal to the number of labels produced by the annotation module. The dimensions are weighted by statistical methods considering higher weights for phrases rather than single words.

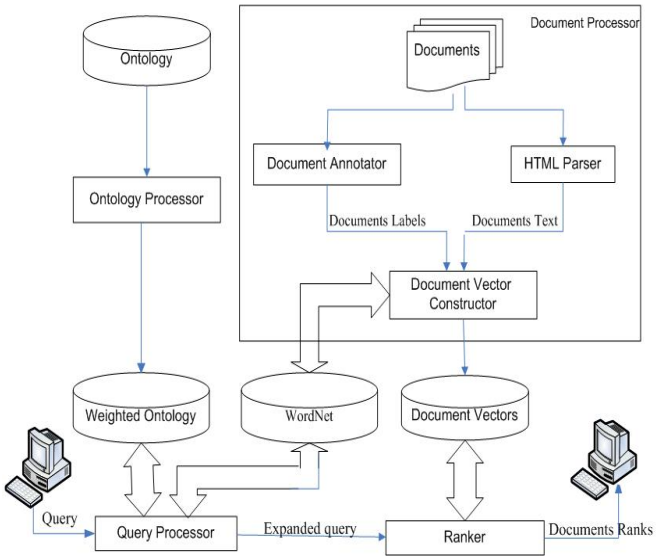

Fig.1 The structure of ORank system

The ontology processor assigns weights to 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 techniques 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 expanded concepts (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 a set of documents 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. The ontology-based annotator uses information extraction techniques for mapping the document's words and phrases to ontology's concepts. ORank annotates documents in a semi automatic way. It uses an online tool, AeroSwarm [28] for automatic document annotation. As this annotation may be not sufficient for expressing the document's concepts, the system administrator can add the required labels too. As computing the relevance degree between the query and the document is similar to the vector space model, it is necessary to build a 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 varied for different documents.

To build the vectors we should first extract the text part of the document, so we adapted the HTML parser [29] for analyzing the HTML documents, removing HTML tags and extracting the text part of them. The output of the HTML parser will be fed to the stemming module to replace text's words with their root forms. The stemmed text part of each document will be passed through the annotation module to extract the documents labels. The labels may be single words or phrases (based on what exists in the ontology).

To extract appropriate phrases (labels) and count their frequency in the document, the following algorithm is used. In this algorithm, the maximum number of words in phrases of each document is equal to the maximum length of document labels.

1. Set variable i to 1.
2. Set variable l to the maximum length of document labels in the current document.
3. From word i to the end of the document do
   a. Consider the words between word i to word i+l.
   b. Extract all phrases in this range. (All permutations of length one to l)
   c. Search each phrase in the document labels.
   d. Increase the frequency of the longest phrase in the document labels, by one.
   e. Increase variable i by one.

In ORank each document vector is stored in a table in the system database. This table contains the labels and their weights. A label's weight indicates its relevancy degree to the document concept. This weight is calculated by statistical processing of the documents as shown in equation (1). In this calculation, document phrases have higher weight 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 convert the query into a representation, which is comparable with the document representation. As it is said, in ORank each document is represented by a vector. Therefore a vector should be built for the query too. Since the keywords in the query might not show the user's information needs obviously, ORank suggests a new approach for expanding the query. Two main parts of this approach are phrase extraction and flexible expansion of phrases based on various conceptual relationships in the ontology.

A. Phrase extraction- As we need both single words and phrases of the query, so the system will extract every possible phrases from the query. To do this, it first does stemming on all words to find their roots. Then it extracts every possible combinations of words reserving the order. The phrases that exist in the ontology or occur in the document set labels are selected as query phrases. 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, is selective. In other words, expansion 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. These concepts form an initial concept set and their initial activation value is set to one. Activation values of other concepts in the ontology are zero. After creating the initial concept set, ORank moves in a weighted ontology through the expansion relation(s) to extract the related concepts and update the activation values. To do this, the ontology is traversed over the selected relationships to produce the concepts that are related to the initial concept set. Ontology traversal can be done in multi levels. The activation value of related 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 the activation value of concept i, $I_{j,k}$ is the 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 parenthood and childhood relations (hyperonymy and hyponymy) is shown in the following example:

I. For each traverse level, do:
   1. Select an unselected concept from the initial concept set (called current concept). The concept set initially contains query words and extracted phrases. The concepts that are the results of expansion should be added to this set later.
   2. Obtain the parents of the current concept.
   3. Calculate the activation value of each parent by equation (4)

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

   In this equation, $I_{i,p}$ 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. Add the parent and its activation value to the concept set. If the parent already exists in the concept set, update its value to the maximum obtained value.
   5. Goto step 1.

II. Repeat Phase I for the children (instead of parents) of initial concept set.

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

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

In this equation $I_{j,p}$ 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, obtained by the above algorithm, makes 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

The ontology processor is responsible for assigning weight to ontology's links. This weighted ontology is used in the query processing.

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

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

In this equation $n_{i,j}$ is the number of related concepts to concept c_j by relation i (the sum of in-degree and out-degree of c_j according to relation i) and *m* is the number of selected relations. So $\sum_{i=1}^{m} n_{i,j}$ is the total number of related concepts to concept instance c_j and $\sum_{i=1}^{m} n_{i,j,k}$ is the total number of related concepts to both concept instances c_j and c_k .

For instance if the selected relation is hyponymy, to calculate the similarity measure of two related concepts c_j and c_k, the number of their common fathers and common children is divided to the number of c_j 's fathers and children.

The specificity measure indicates how much a relation is specific. Equation (7) is used to calculate the specificity measure of the relation from c_j to c_k. 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 equation, n_k is the number of relations which their destination concept is c_k.

In ORank, the 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 ontology relations and their weights are stored in the system database. Therefore, the ontology is ready to be used in the query processor.

D. The 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 vectors' lengths. Then documents are sorted according to their ranks and will be presented to the user.

IV. 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 a user interface. The ontologies that were used in ORank evaluation were Cyc and WordNet. The queries were chosen in a way that comprises both worst and best cases.

Then, ORank processed the documents to compute their vectors and stored the vectors in the system database. This process is done once and there is no need to repeat it for the processed documents anymore. Of course it is always possible to add new documents to this collection. Therefore, when the user query is presented to ORank, 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 figures 2-4.

A. *Improved SA algorithm*

We tested ORank in two modes in order to show how using improved SA algorithm increases the relevancy degree of retrieved documents in comparison to keyword-based search:

1- In the first case study, we applied our improved SA algorithm, in which the query's keywords and phrases were spread using hyponym relation of CYC ontology.

2- In the second case study, the documents were ranked using keyword search and no expansion was done.

As figure 2 shows, applying improved SA algorithm has increased the precision of retrieved documents relative to their recall.

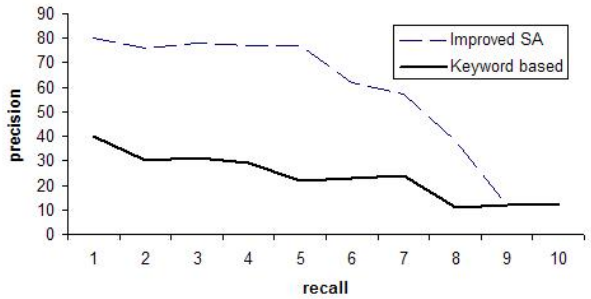

Fig. 2 Comparing improved SA with keyword search

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.

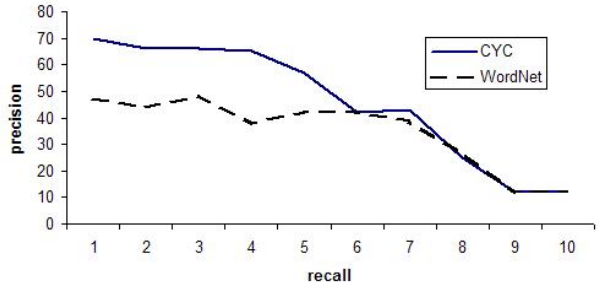

Fig. 3 Comparing CYC with WordNet

C. *Hybrid Relation*

We claimed that by considering different ontology's relations 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 figure 4.

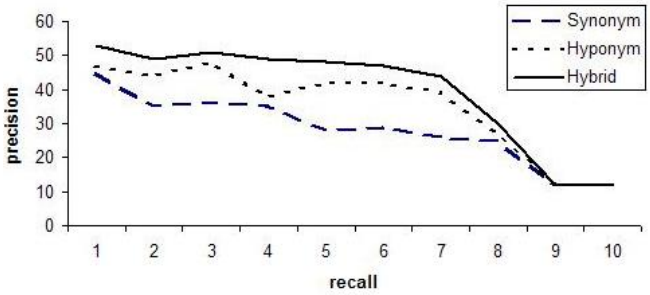

Fig. 4 Comparing hybrid relation with synonym and hyponym relations

V. CONCLUSION

In this paper, we introduced an ontology-based model for ranking documents according to their relevancy to the user's query. The proposed model improves the precision of existing statistical models using concept instances in the document and query's vectors instead of words. Another salient point in this method is extracting the query and document's phrases and stemming them. In addition, we improved SA algorithm to expand query's keywords and phrases to their related concept instances using various relation types of an arbitrary ontology. Therefore, the relevancy degree of the retrieved document is 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 the comment property. If they were found, the query would have been expanded by that class.
- Document annotation and query expansion accomplish by applying ontology. So using special purpose ontologies would have great effect in test results.
- As our model depends on annotation, designing an appropriate annotation algorithm increases the relevancy of retrieved documents.
- More precise approximation of phrases' weight coefficient in comparison to words' requires more tests.
- For computing weight coefficient of relations 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] W.M Shaw, "Term-Relevance computations and perfect retrieval performance". Information Processing& Management, Volume 31, No. 4, 1995, Pages: 491 - 498.
[9] G. Amati, S. Kerpedjiev, "An information retrieval logical model: implementation and experiments". Technical Report Rel 5B04892, Fondazione Ugo Bordoni, Roma, Italy, March 1992.
[10] H. Turtle, W.B. Croft, "Evaluation of an inference network-based retrieval model". ACM Transactions on Information Systems, Volume 9, No. 3, 1991.
[11] M. R. Henzinger, "Hyperlink analysis for the web". IEEE Internet Computing, Volume 5, Issue 1, January 2001, Pages: 45 - 50.
[12] 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.
[13] 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.
[14] 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.
[15] H. Zhuge, L. Zheng, "Ranking Semantic-Linked network". WWW (Posters), 2003.
[16] 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.
[17] 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.
[18] D.A. Grossman, O. Frieder. "Information retrieval algorithms and heuristics". Second ed. . Springer. 2004.
[19] R. Rada, H. Mili, E. Bicknell, M Blettner, "Development and application of a metric on semantic nets". IEEE Transactions on System, man, and Cybernetics, Volume 19, No. 1, Pages: 17 - 30.
[20] Y.W. Kim, J.H. Kim, "A model of knowledge based information retrieval with hierarchical concept graph". Journal of Documentation, Volume 46, No. 2, 1998, Pages: 113 - 136.
[21] 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.
[22] J.M. Ponte, W.B. Croft, "A language modeling approach to information retrieval". In Proceedings of the 21st ACM SIGIR Conf. on Research and Development in Information Retrieval, Pages: 275 - 281.
[23] W. A. Woods, L. A. Bookman, A. Houston, R. J. Kuhns, P. Martin, S. Green, "Linguistic knowledge can improve information retrieval". Applied Natural Language Conferences, Proceedings of the Sixth Conference on Applied Natural Language Processing, 2000, Pages: 262 - 267.
[24] H. Rode, D. Hiemstra, "Conceptual language models for Context-Aware text retrieval". Proceedings of the 13th Text Retrieval Conference (TREC), NIST Special Publications, 2005.
[25] R. Belew, "Adaptive information retrieval". In Proceeding of the Twelfth Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval, 1989, Pages: 11 - 20.
[26] H.Chen, "Machine learning for IR: Neural networks, symbolic learning, and genetic algorithims". Journal of the American Society for Information Science, Volume 46, No. 3, Pages: 194 - 216.
[27] Rocchio, "The SMART retrieval system experiments in automatic document processing". Relevance Feedback in Information Retrieval, Prentice Hall, 1971, Pages: 313 - 323.
[28] Aeroswarm,http://ubot.lockheedmartin.com/ubot/hotdaml/aeroswarm.html
[29] LCNetTools,http://itlang/vb.net/archivio.asp?subMenu=Tutte&FullTexton&TypeRi=AND&keyword=LCNettools