A 3-Tuple Information Retrieval Query Interface with Ontology Based Ranking

Jinwoo Kim, Dennis McLeod
Computer Science Department
University of Southern California
jinwook@usc.edu, *mcleod@usc.edu*

Abstract

Currently keyword search is a prominent data retrieval method for the Web because the simple and efficient nature of the keyword processing allows it to process a large amount of information with fast response. However, keyword search approaches do not formally capture the clear meaning of a keyword query and fail to address the semantic relationships between keywords. As a result, its recall rate and precision rate are often unsatisfactory, and therefore its ranking algorithms fail to properly reflect the semantic relevance of keywords. This means that the accuracy (the precision and the recall rate) of search results is often low. Therefore, we proposed a new 3-tuple query interface and corresponding ranking algorithm. And we presented a comparison of the search accuracy of our new query interface and conventional search approaches.

Keywords: Information Retrieval; Ontology, Query Interface, Search Ranking Algorithm, Information Integration

1. Introduction

At present, keyword search is a dominant search method due to its efficiency. However, it does not systematically present a semantic understanding of keywords because it is difficult to identify the correct meaning of each keyword without considering their semantic relations or without considering their meanings in the context of a complete sentence. Consequently, its search result ranking is often disappointing. When a user searches information on the web through search engines, if the information he or she is trying to find is not included in the high ranking, the user normally takes the trouble to search again with a new query rather than flipping through the next pages [1, 2]. This trouble arises because current ranking algorithms are not able to properly map the semantic relevancy between the query and the web contents.

In this paper, we are introducing a new form of query interface which places one or more wildcards between keywords or at the beginning or at the end of a multi-word query will allow search engines to return what users are searching for more effectively. For example, if a user wonders what a shark did to a victim, he or she can place a query of shark, [wildcard], victim. This new query interface helps our model to calculate keywords, which more frequently occur in the position of wildcard as more relevant to what users are actually looking for. We expect that our new query interface will work better for the question form of a query.

Our research particularly focuses on increasing the accuracy (the precision and the recall rate) of search results for question-type, multi-word search. For that purpose, we propose a new query interface including wildcard and a statistical ontology-based semantic ranking algorithm based on sentence unit. First, we allocate higher-ranking scores to keywords located in the same sentence compared with keywords located in separate sentences. While existing statistical search algorithms such as N-gram [3] only consider sequences of adjacent keywords, our approach is able to calculate sequences of non-adjacent keywords as well as adjacent keywords. Second, we propose a slightly different type of query interface, which considers a wildcard as an independent unit of a search query to reflect what users are actually looking for by way of the function of query prediction based on not query data but actual web data. Unlike current information retrieval approaches such as proximity approaches, semantic and natural language assisted search approaches [19, 20], statistical language modeling, query prediction and query answering, our statistical ontology-based model synthesizes proximity concept and statistical approaches into a form of ontology. And the ontology helps to improve web information retrieval.

2. Related work

It is known that existing search engines adopt a number of factors to determine the ranking results of keyword search, such as title, anchor, URL, plain text large font, plain text small font, PageRank, within-document frequencies, inverse document frequencies, document lengths, etc. [4, 5, 6]. Among them, for multi-keyword search, the most important factors to determine their ranking results are frequency and proximity [4, 6, 7, 8, 9, 10, 11, 12]. One of the main problems with the current ranking algorithm of multi-word search arises from the fact that its methodology calculates the relevance of keywords only by their proximity without considering whether they exist in the same sentence or not.

Another problem is that current search algorithms fail to capture the semantic relevance of sequences of keywords when they are not situated adjacently due to an insertion of other words not included in the query such as adverbs or adjectives.

There is the other problem with current search algorithms. Users often encounter situations where they do not exactly know the information they are actually searching for, but where they only know keywords related to it. However, most current search engines are not able to handle this situation effectively because they produce search results mainly based on query inputs, which users only know. To solve this problem, our approach introduces a new form of interface including wildcard.

Our research does not try to replace existing search algorithms with our new algorithm model, but rather tries to improve the accuracy (the precision and the recall rate) of question-type, multi-word search results by solving aforementioned problems caused by current search algorithms. This means that existing search engines can adopt our method in addition to their existing search methods in order to help users get more semantically relevant search ranking results.

2.1. Proximity search

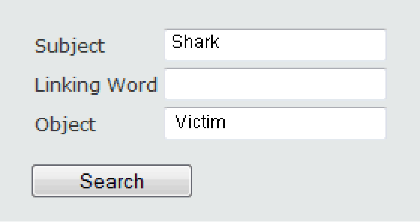
Proximity search is based on the idea that more semantically relevant keywords are placed closely. As several researches on proximity search show [6, 7, 8, 9, 10, 11, 12], this method has improved information retrieval a great deal. To calculate proximity, the proximity search considers distance between keywords. However, the difference between our statistical ontology-based approach and the current proximity search is that while the proximity approach calculates proximity only by considering distance between keywords, our approach additionally calculates distance between keywords and wildcard in addition to calculating distance between keywords. The main purpose of calculating distance between wildcard and keywords is that it allows our model to calculate proximity between the user's query and what the user is actually looking for when the user does not exactly know wanted answers. Our method is especially useful when users only know keywords related to the answer, but when they do not exactly know what the answer is.

In addition, among many proximity algorithms, our approach adopts the method of calculating proximity based on sentence unit. This method calculates the proximity of keywords by presenting the links of keywords based on sentence unit and assigning the same value to keywords group within the same sentence. Unlike proximity search, which only considers distance between keywords, this method allows us to calculate how frequently the keywords given occur by sentence unit, thereby creating statistical ontology based on proximity and frequency of keywords. The merit of considering both proximity and frequency of keywords helps our model to produce more semantically accurate search results than proximity search because human language mostly expresses different meanings by sentence unit. This means that our method of categorizing human language by sentence unit can offer a more semantically relevant way of calculating proximity values of keywords.

Another benefit of our approach over current proximity approaches is that when other words such as modifiers are inserted between query keywords in the same sentence, the proximity value of the query keywords increases in current proximity approaches and as a result their semantic relevancy is recognized as low. However, in the same situation, our approach calculates the proximity value of keywords with in-between modifiers as the same as that of the keywords without in-between modifiers. This way, our approach is able to calculate more semantically relevant search results by ignoring elements such as modifiers, which are not semantically relevant to what the user is actually looking for.

3. New query interface

Our statistical ontology-based search model adopts a new form of query interface, which we have developed for our approach as shown in Fig. 1 The query interface window has three input boxes. If users need to enter a two-word query, in our query interface users enter two keywords into any two input boxes out of three. This means that one box is always left blank, which can be used as a wildcard to predict what users are actually looking for. For example, if a user wonders what a shark did to a victim, he or she can place a query of [shark], [wildcard], [victim] or if a user wonders what shark attacked, he or she can place a query of [shark], [attack], [wildcard].



Subject: Shark
Linking Word:
Object: Victim

Figure 1. New query interface

One of the main functions of our new query interface is to allow users to express what they are looking for in the form of the wildcard if they do not exactly know what it is. Users often encounter situations where they do not exactly know the information they are actually searching for, but where they only know keywords related to it. However, most current search engines are not able to handle this salutation effectively because they produce search results mainly based on query inputs, which users only know. By contrast, our query interface is able to tackle this problem by means of the wildcard.

In addition, our new query system allows users to predict words between the query keywords, whereas the N-gram only predicts words following or preceding the query keywords. This new query interface using wildcard significantly reduces the amount of data to process for building ontology. Moreover, our ontology is able to return the most frequently used keywords in the location of wildcard from the actual web data. This method can also work as a query prediction system.

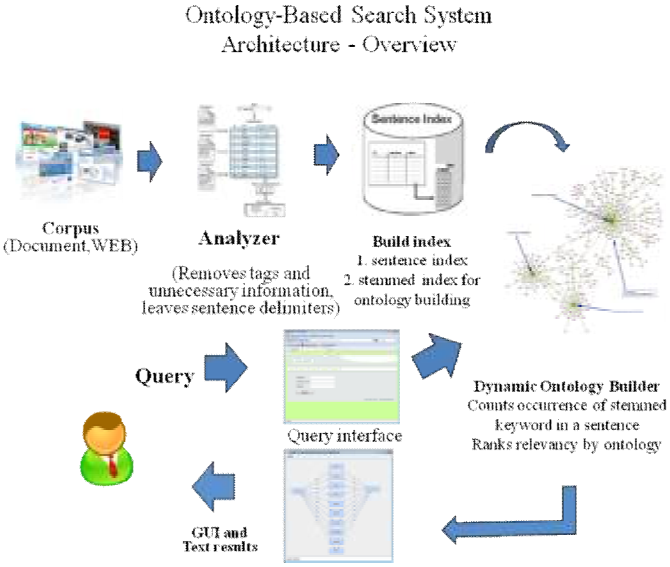
Our new query system calculates the frequency of keywords, which occur in the location of wildcard within our ontology, thereby creating a statistical language model in the location of wildcard. Based on the statistical language model, we build ontology and increase ranking value by referring to the ontology.

4. Approach

In order to improve information retrieval, we have adopted a method of developing a statistical ontology and a new query interface. This approach is based on the idea that in order to improve search results, search engines should understand the semantic relevancy of keywords by sentence unit because human languages tend to express different meanings by sentence unit. Our ontology-based search model better reflects the semantic relevancy of keywords because our ontology has been built based on data extracted by sentence unit. We have also developed a new query interface to handle the situation where users do not exactly know what the answer is to their questions, thereby expressing what they are looking for in the incomplete form of a keyword query. This method also helps us to build our statistical ontology more efficiently.

4.1. Overview and general architecture

In order to test our hypothesis that referring to the ontology [13, 14, 15] and adopting a new query interface produce more semantically relevant search rankings, we have developed a statistical ontology-based semantic search model. Using the model, we have built the ontology, have created a new query interface, and have generated and re-ranked a query-relevant subset of the corpus of English News Text from the TREC [16] with a list of questions and answers for each query. As shown in Fig. 2, our statistical ontology-based search system consists of new query interface, ontology builder, analyzer, indexer, user interface and corpus. The detailed process of each step below is described in the following sections.



Ontology-Based Search System
Architecture - Overview
Corpus
(Document, WEB)
Analyzer
(Removes tags and unnecessary information, leaves sentence delimiters)
Sentence Index
Build index
1. sentence index
2. stemmed index for ontology building
Query
Query interface
Dynamic Ontology Builder
Counts occurrence of stemmed keyword in a sentence
Ranks relevancy by ontology
GUI and
Text results

Figure 2. Ontology-based search system architecture

4.2. Corpus and test data

Evaluating search results in a quantitative way is a difficult task. For this reason, to show the effectiveness of our statistical ontology-based ranking algorithms, we used 2007 Question Answering Data from TREC (Text REtrieval Conference) [16, 17]. This data is called AQUAINT 2, which contains about a million news articles developed by the TREC, whose main goal is to help research within the information retrieval community by providing the infrastructure necessary for large-scale evaluation of text retrieval methodologies. For this goal, the AQUAINT 2 supplies a set of questions and answers to help evaluate search performance in a quantitative way. Using the AQUAINT 2, we evaluated our search model in a quantified way.

4.3. Analyzer and indexer

As Fig. 2 shows well, our statistical ontology-based semantic search has been built referring to the index structure of search engines. To perform the indexing of our data, we used and modified the open source Apache Lucene indexer (version 2.9.1) [18] and pointed it to look at all of our individual TREC documents. We began by using the built-in stop words analyzer while modifying the white space tokenizer and filter. Whereas existing search engines remove sentence delimiters while indexing so that they are not able to process data by sentence unit, our approach allows the tokenizer to discard all symbols, other than sentence delimiters such as periods as all items are one character in length. In addition to removing tags, whitespace, and "stop words" such as "and", "the", and "to", we added "www", "http", "copyright", and other words that appear frequently in the footers of web pages and that are believed to be unnecessary when looking at the domain of our corpus. Users can modify a text file to add/remove stop words or specify ones in addition to the defaults in the command line. In this way, users can change the list of stop words for various types of corpus..

5. Experiment

5.1. Experiment approach

This research attempts to prove the effectiveness of our methodology by several experiments with the Text Retrieval Conference document collection. We have adopted two-word queries as our experiment's object, as a two-word query is the most common query form. After applying our statistical ontology-based algorithm to the Nutch search engine, we compare the result with results of original Nutch search and Google Desktop Search. The result demonstrates that our methodology has improved by 28% accuracy (the precision and the recall rate) of search results against original Nutch search approach.

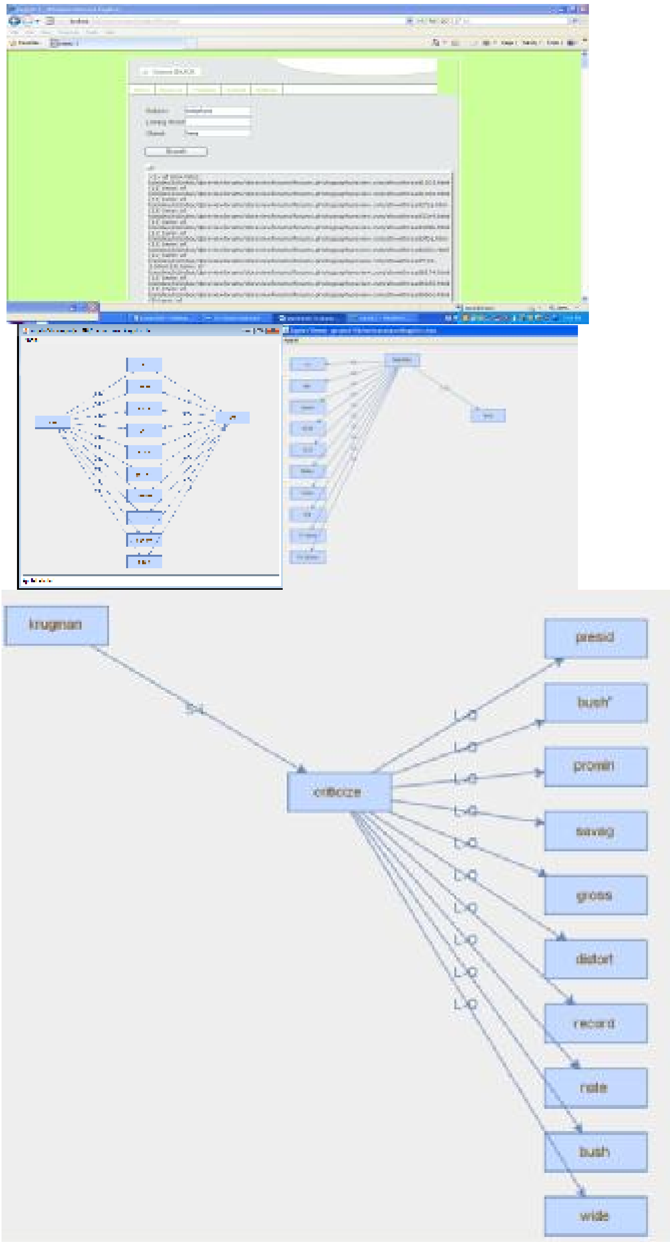


Figure 3. Ontology-based search system's output (Input query: Krugman criticized [blank])

5.2. Experiment result

In order to test both recall rates and precision rates of each search approach, we have used questions offered by the TREC data, which have multiple correct answers. A total of twenty-two queries are used after excluding queries which have 0 results for all three search approaches. Fig. 4 below shows a result comparing the precision rates of each search approach. The X axis shows the TREC's query ID, and the Y axis is the precision rate.

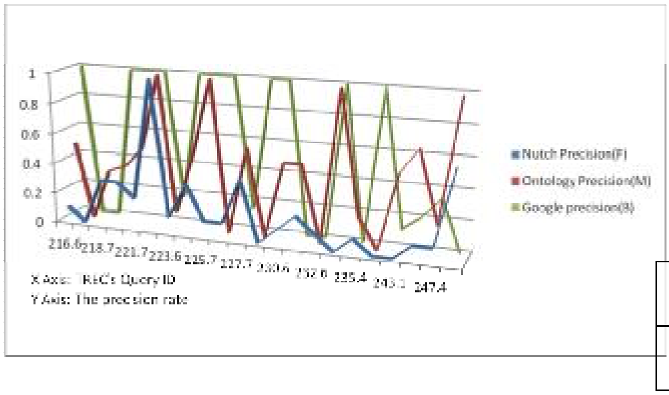


Figure 4. Precision comparison result

Table 1. Average precision of each query

Search Approaches	Nutch	Ontology	Google
Precision Avg. of 22 queries	0.1924242	0.4293831	0.5433621

As Table 1 shows, our statistical ontology-based search improved the precision rate by about 123% over original Nutch without ontology. Our approach considering whether or not keywords are placed in the same sentence, adds one more constraint to the search conditions of the previous search algorithms, thereby filtering more irrelevant search results than original Nutch can. For this reason, our approach produces more correct search results so that its precision rate is expected to increase more than original Nutch. Meanwhile, Google Desktop Search showed a very high precision rate because Google Desktop Search is sensitive to verb tenses or conjugation during the search process. Hence, we came to know that Google Desktop Search is more focused on producing precise search results than retrieving a wide range of target corpus.

The following Fig. 5 shows a result of recall rates of each search approach. The X axis shows the TREC's query ID, and the Y axis is recall rate.

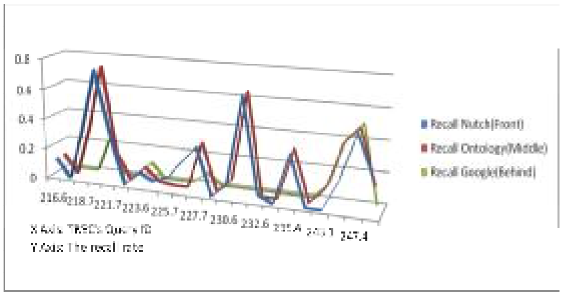


Figure 5. Recall rate comparison result

Table 2. Average recall rate of each search approach

Search Approaches	Nutch	Ontology	Google
Recall Rate Avg. of 22 queries	0.1976639	0.1834666	0.064860

As we expected, calculating whether keywords are placed in the sentence or not helped our ontology-based model to return a lesser number of search results than just calculating frequency and distance. When the number of search results decreases, its recall rate is expected to decrease. In our experiment, our statistical ontology-based search's recall rate also has decreased by about 7% over original Nutch's recall rate.

Meanwhile, Google Desktop Search shows a lower recall rate compared with our model and Nutch. This result can be easily expected because Google Desktop Search showed a higher precision rate in the previous experiment. This result demonstrates that Google Desktop Search considers more constraints during the search process in order to acquire a higher precision search result than our model and Nutch's.

In order to properly evaluate search engines, both precision and recall rate are generally considered all together. Our experiment also has evaluated search results this way. In order to evaluate three search approaches in a quantized way and to consider both precision and recall rate equally, we have adopted a standard, which the TREC suggests as shown below to aggregate recall rates and precision of each search approach.

IR = # instances judged correct & distinct/|final answer set|

IP = # instances judged correct & distinct/# instances returned

F = (2*IP*IR)/(IP+IR)

The following Fig. 6 shows the evaluation value of each search approach. The X axis shows the TREC's query ID and the Y axis is the TREC's search evaluation value.

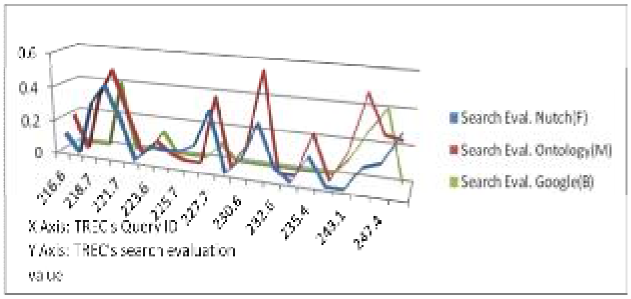


Figure 6. Average of TREC's evaluation value comparison result

Table 3. Average evaluation value of each query

Search Approaches	Nutch	Ontology	Google
Avg. of Eval of All queries	0.1417273	0.1817460	0.063496

We have calculated the evaluation value for each query and the average of all evaluation value is used to compare each search approach.

Table 4. Summary of experiment results

Search Approaches	Nutch	Ontology	Google
Recall Ratio Avg of all queries	0.197663	0.183466	0.06486
Precision Avg. of all queries	0.192424	0.429383	0.54336
Evaluation value Avg. of all queries	0.141727	0.181746	0.06349

In general, recall rate is in inverse proportion to precision. Our experiment results also show the same result. Our statistical ontology-based model applying the sentence-based filter to original Nutch served to lower the coverage of documents, and as a result, the recall rate went down. However, in the case of our experiment, precision degree has increased much more than the recall rate has decreased.

Accordingly, our algorithms and query interface have improved the final evaluation value by 28% compared with Nutch without our methods. The Google Desktop Search has a number of search algorithms. One of them makes it too sensitive to different forms of words, and as a result, although the precision degree was relatively high, the recall rate was too low so that the TREC's final search evaluation value was lower than our methodology.

In order to properly evaluate search engines, both precision and recall rate are considered. However, in order to test more queries and acquire more credibility, our methodology used one hundred factoid-type questions with only one single answer thereby being able to evaluate precision only. The following result shows that the precision increment rate is similar to that of list-type questions.

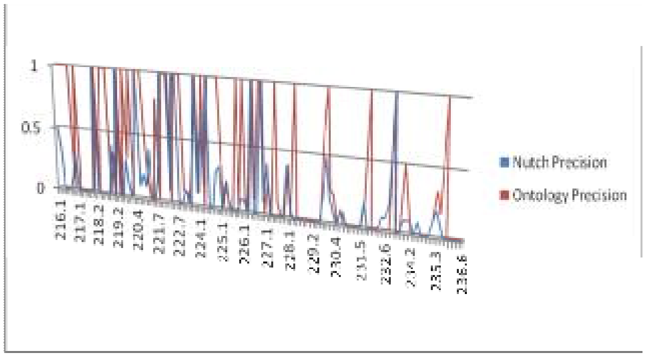


Figure 7. Average precision of each factoid-type query

Table 5. Average precision of each query

Search Approaches	Nutch	Ontology
Precision Avg. of 121 queries	0.176752906	0.357627866

Google Desktop Search has been excluded in this experiment since it produces many search results without any output as Figure 6 shows. Therefore, the precision of Google Desktop Search is ranked high as in the previous experiments. However, the recall rate cannot be evaluated with factoid-type questions because there is only one correct answer. We concluded that a high precision rate calculated from search results without any output is not valid. 102 Factoid questions and 19 list-type of questions are used for this experiment and the precision has increased by 102%.

5.3. Experiment conclusion

We have verified our hypothesis that allocating higher-ranking scores to keywords in the same sentence for multi-word search is able to produce more semantically relevant search rankings in the top-ranked documents than Nutch and Google Desktop Search. We also have proved the hypothesis that placing wildcards between keywords or at the beginning or at the end of a multi-word query helps to indicate user's information demand more clearly. As a result, with our model, more precise and efficient information retrieval is possible. Furthermore, our statistical ontology-based model, which adopts a statistical language model for multi-keyword search, helps to generate semantically more relevant information retrieval results.

6. Conclusion

In order to overcome the aforementioned problems with information retrieval approaches, our approach considers several elements together such as whether keywords are located in the same sentence or not, how frequently keywords appear in the location of wildcard, and distance between keywords and wildcard. Moreover, our approach creates statistical language models based on the order and frequency of keywords found in the target web documents. We calculate the relevancy of the query and the document collections by referring to our ontology, which actually works as a statistical language model. This way, our approach helps to offer more semantically precise values of the relationships between the inserted query and the document collection, thereby placing more semantically relevant web pages higher in the rankings.

7. Contributions

A primary contribution of this research is to introduce a more semantic understanding of web documents by adopting a method of retrieving web data by sentence unit. Our sentence-based ranking algorithms were able to improve by 28% the semantic relevancy of original Nutch search ranking by better reflecting semantic relevance of keywords. The other new method has also contributed to the good result. By marking what a user is searching for as a wildcard, we predict an answer to the wildcard based on our ontology we have developed with actual web data, not with a query log. This means that we create a query sequence including a wildcard, which allows us to more easily find an answer to what a user is searching for because the ontology enables us to find words which appear most frequently in the place of a wildcard on the web data. While displaying such words with this method, based on actual web data we can provide a query prediction, a query answering, and automated categorization. The difference between a query prediction and our methodology is that while current search engines' query prediction based on query log only helps complete a query, our algorithms help users to find more relevant data to what they are searching for on the web, thereby improving the rankings of search results. Our statistical ontology-based algorithms also help to save a great deal of users' time while they are searching information on the web by improving semantic relevance of rankings of search results.

8. References

[1] Bernard Jansen , Major Bernard , J. Jansen , Amanda Spink , Tefko Saracevic , Real Life, Real Users, and Real Needs: A Study and Analysis of User Queries on the Web (2000)

[2] A. Spink, D. Wolfram, B.J. Jansen, and T. Saracevic. Searching the web: the public and their queries. J. American Society for Information Science and Technology, 52(3):226–234, 2001.

[3] Ronald Rosenfeld, TWO DECADES OF STATISTICAL LANGUAGE MODELING *IEEE* 2000.

[4] Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, *WWW7 Proceedings of the seventh international conference on World Wide Web 7.* 1998

[5] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Manuscript in progress. [Page 98] 1999.

[6] Tao Tao, ChengXiang Zhai, An Exploration of Proximity Measures in Information Retrieval 2007.

[7] Benny Kimelfeld. Yehoshua Sagiv, Efficient Engines for Keyword Proximity Search. 2006.

[8] E. M. Keen. The use of term position devices in ranked output experiments. *The Journal of Documentation*, 47(1):1–22, 1991.

[9] E. M. Keen. Some aspects of proximity searching in text retrieval systems. *Journal of Information Science*, (18):89–98, 1992.

[10] D. Hawking and P. Thistlewaite. Proximity operators – so near and yet so far. *In Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 131–143, 1995.

[11] Y. Rasolofo and J. Savoy. Term proximity scoring for keyword-based retrieval systems. *In Proceedings of the 25th European Conference on IR Research (ECIR 2003)*, pages 207–218, 2003.

[12] S. Buttcher, C. Clarke, and B. Lushman. Term proximity scoring for ad-hoc retrieval on very large text collections. *In SIGIR '03: Proceedings of the 26nd annual international ACM SIGIR conference on Research and development in information retrieval*, 2006.

[13] S. Chung, J. Jun, and D. McLeod. *A web-based novel term similarity framework for ontology learning. In OTM Conferences (1),* pages 1092–1109, 2006.

[14] Nicola Guarino,National Research Council, LADSEB-CNR, Corso Stati Uniti 4, I-35127 Padova, *Italy, Formal Ontology and Information Systems* 1998.

[15] Boris Wyssusek, Queensland, University of Technology, On Ontological Foundations of Conceptual Modeling 2005.

[16] Ellen M. Voorhees, Overview of the TREC-9 Question Answering Track 2001.

[17] EllenM. Voorhees and Donna K. Harman, National Institute of Standards and Technology, *TREC: Experiment and Evaluation in Information Retrieval 2005*.

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

[19] Fernandez, Miriam; Zhang, Ziqui; Lopez, Vanessa; Uren, Victoria and Motta, Enrico (2011). Ontology augmentation: combining semantic web and text resources. *In: 6th International Conference on Knowledge Capture (K-CAP 2011), 25-29 Jun 2011, Banff, Alberta, Canada.*

[20] E. Alfonseca, P. Castells and M. Ruiz-Casado, Automatising the Learning of Lexical Patterns: an Application to the Enrichment of WordNet by Extracting Semantic Relationships from Wikipedia. *In the journal of Data and Knowledge Engineering vol. 61 (3), pp. 484-499, Elsevier*