Web Query Expansion and Refinement using Query - Level Clustering

Senthil Kumar N, Saravanakumar K
School of Information Technology &Engineering, VIT University, Vellore, India
Email: senthilkumar.n@vit.ac.in, ksaravanakumar@vit.ac.in

Abstract

The objectives raised in this paper are to pave the new dimension to Internet searching and bring the semantic core strategies to the forefront to add values to the search process. In precise, "the search must be what user wish, not what user types". To know the process of search intricacy, we observed the vocabulary contradiction and mismatch problem existence during retrieval can estimate the irrelevant document matching. Generally, a term or vocabulary mismatch can happens to the search iteration only if the terms not present in the fetched documents. Many techniques have been proposed such as library science, pseudo relevance feedback and later semantic indexing etc, where all the algorithms tend to find the objectives sustained but did not deal with alternate process. Hence we have proposed a technique which gives the sheer implications of all the pitfalls and device a new mechanism to support the mismatch problem. By bringing the semantics aspects of the sentences and word order of the sentence to the core part, we have emulated the proper solution to get rid of sentence or term mismatch problem.

Keywords: semantic similarity, query expansion, lexical database, semantic vector, word order vector.

1. Introduction

It is observed that searches conducted on search engines are purely for learning, entertainment or to carry business transactions. But many searches are having the real purpose and made some impact on to take important decision about life, health, major purchase of certain things or quenching the business community quest for an acquisition target. Although the search engines have been achieving remarkable success in recent years and reaching new heights in bringing the quality results to the users, but still poor at helping the people to find exactly what they want, and their needs, especially in the circumstances where the users don't have a clear idea of what they are actually looking for. Both the conventional and the modern search engines are simply attempt to find the best match between what users asks for and what is available in their indices. Search engines have not done a good job of assessing exactly what the user wants because they are lack in the sheer knowledge of the context that made the user to generate the poor search query. Besides, the ambiguities of language are an issue which is more difficult to understand the exact intent or absolute meaning of the user's query. Searching is a iterative process in which a users grab the intended web pages via trial and error query methods that work best for the issue to resolve. It might surprise most people to know that search engines only index a small percentage of the knowledge resources available. This occurs because many web pages are stored behind password protected sites, pages are dynamically created and disappear once they serve their purpose, and several types of information are in formats that are not useable by search engines.

Users search the web for the information with their needs and mostly their queries are explicit expression of their search needs. The information need in web search process can be termed as intent and that demands more productive fetching of web pages. Many times, the user query is not adequate to describe the intent which they actually aimed but it only contains few terms. This problem exists, because of the lack of domain knowledge or insufficient skills to express their intents. And also, the intent primarily resides in the mind of the user and thus difficult to observe. Despite all these hiccups, even if the user is obliged to reveal his actual intent, it's also a challenging task to describe the intent accurately. Hence, users can reformulate the initial query following the search results shown to them and their understanding would become more specific by extracting clues from search activities. Basically, the web users are categorically separated as: navigational, informational and transactional. The navigational query can be used to reach the specific web site or web pages where the users don't have the clear indication of it. The navigational queries can take the user to different web pages which are all relevant to one another. The information queries are very specific where it demands the relevant information about the given topic. The users want to learn or find the information which might scatter at various web pages or sites. The transactional queries are absolutely interactive and carry out a robust transaction with the websites like downloading music, carry out online shopping, playing online games etc.

In order to achieve the search process more productive, we need to extract the semantics from the questions which the user often posed in the web. The questions can be categorized in many ways like the queries which are only yes or no type, some queries are seeking the reasons of particular thing (like why type questions), few queries are asking the opinion of particular things, some queries wants to know the details of the particular

information (like how it happens or occurs), certain queries would enable interactive session to discuss and share, some queries needs the translation and so on. So finding the semantic relations from the given user question can fall in any of the above said categories and made easy to interpret the observations.

2. Related Work

The objectives proposed are concerned with estimating a searcher's intent and improves the search process. The peculiar problem faced by most of the users in searching the web is not getting the required information which they are seeking. This is due to the fact that most of the web users are just enter their queries which consist of only two or three terms and inadvertently that are often too ambiguous to fetch the required sources of pages. According to the paper [4], the study pinpoints the statistics that from 7% to 23% of online web users have entered queries less than three terms. To understand the short queries, many researchers have been carried out in the field of query disambiguation and term extraction. In the paper[7][8], the basic idea lingering beneath is that to replace user query terms with more accurate and precise terms and narrow down the search to the context users actually have in the mind. To quench this satisfactory process, it demands the approaches which exploit some external knowledge such as WordNet, taxonomies, or user models to normalize the meaning of the search terms in distinct context. Though the results are satisfactory, still lack in absolute results set in those approaches due to the expansion of search terms with absolute words. Other approaches[11][13] are based on query log analysis which deals the query for long periods of time on the server side and used to clarify the user intention by the sheer assumption of short query. The extensive process was under taken to evaluate the similarity pertaining to different users and compared the corresponding query results. And these strategies heavily succumbed to the rare queries and some time no availability of sources. The term generation for the given query can achieve by either co-occurrence analysis or dynamic clustering of query results. To entirely contrast the previous approaches[6][12], the goal is to scrutinize the iterative process of query reformulation and supply the domain specific disambiguate queries which matches to the documents focused on the retrieve topic of interest. And then top ranked results are displayed to the users based on document diversification. This diversification has been performed based on the comparison of documents one against another, assuming that similar documents deal with similar topics. Eventually by providing the similar reformulations of the original query, the user can get the web pages which they actually intended in the first positions. The novelty metric [8] [14] is related to cluster the documents and estimates the amount of new documents contained in a cluster with respect to documents seen before. Clusters can be ranked based on some query similarity measures that guarantees the content relevance of results.

In the paper [3], they have applied the conventional machine learning techniques to automatically classify the user queries based on their locality. And also they were emphasized on the query categorization and manipulate the methods which improve the query result standards. According to paper [1] [2], they were analyzed various retrieval strategies building their progress on query type and attain the best quality results against stipulated categorization. However if we go with manual classification of topic, then it will be a time consuming process and very expensive in nature which demands more human labor endeavors and also the sense of the user queries keep changing over time. In the paper [7], they were proposed a strategy to take the discriminating preferences based on the method which attempt to categorize the association exists between query terms and yield the better query classification against the some association rules. But barely sticking to the association rule and query classification brings bad consequences and often it's a cumbersome task to supply sufficient query logs where the query must be gained from the taxonomy. Recently, in the paper [9], it has also mentioned that query classification can afford the results with high relevancy and precision of the query results can reach the threshold against the previous query results. By utilizing these solutions, categorizing the web query is computationally infeasible and most of the cases, it is prohibitive for a crawler to accomplish this huge task. This is due to the fact that amount of query enlisted in the query log go exponentially and demands mammoth query log storage. The statistics report highlighted that the query log volume increased in millions per day.

3. Extracting the Query Suggestions from sources

Generally the terms that are shown to the web users can actually fetched either from well structured knowledge entropy like thesauri or from the documents pre-defined in the search results. In the due process, if the key terms of the documents are not indexed or the searchers not availing the key terms from the indexed documents, then the probability of reaching the relevancy and precision of the results would be very low. Therefore it has been widely suggested not to pick the search terms which the user actually not intended and give high importance on how the user reacts to someone query. It has followed two approaches to build the effective measure of query type and enact the basic model to produce the alternate query type from the web source.

The first approach deals with global co-location evaluation which accumulates the related terms extracted from diverse document gatherings. The second approach extracts the terms which are subset of the retrieved documents. The term suggestion can be highlighted only if the query suggestion attempted to blend suggested terms with other manipulated terms and offer the alternate new query to the user.

Query suggestions [7] therefore can support new viewpoint and assisting explore unfamiliar domain areas. We implemented three basic suggestion services with different popular search giants for our evaluation: Google, Bing and Yahoo. Additionally, as a novel approach, we have created a combined term suggestion service which combines the thesaurus and a search term recommender service. The terms are chosen by matching the input term against a ranked list of the user terms. The user terms are ordered by the frequency count of their usage.

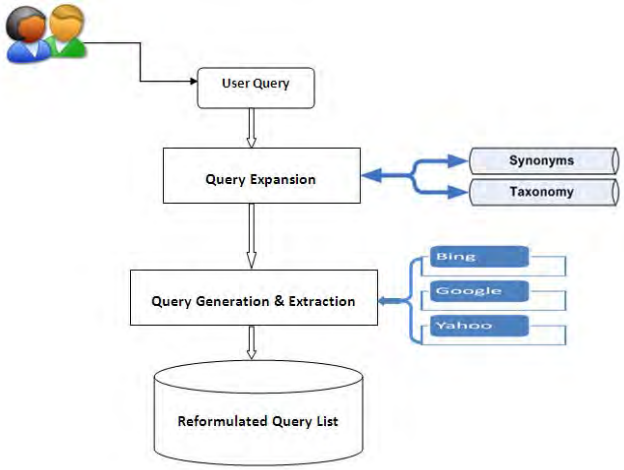

Fig 1: Building the Reformulated Query List

4. Process of Semantic Similarity Measure

Semantic similarity measures play a pivotal role in various applications of natural language processing such as word-sense disambiguation, language modeling, synonym extraction, and automatic thesauri extraction. Although the semantic similarity has prominently used in many applications, finding the semantic similarity between two words would be the hard task. This is due to the fact that some new words are emerged inadvertently and also some new senses are attributed to the existing words. Holding the thesauri [6] for these new emerging words and their senses are observed very costly and maintaining the index would become a challenging task if not possible. In query expansion [3] [8], to hit the relevancy of search, we ought to bring the appropriate synonymous words to the user query and improve the search process. The best way to get the candidate query for the user query is to compare the previous user queries using semantic similarity measures. Whenever the query is semantically matched to the previous queries, then it can be known both the user as well as to the search engine query logs for future reference. The best way to measure the similarity between two words is to calculate the shortest path distance which has a link in the taxonomy. For instance, if the user query tends to be polysemous, then there would be high chance of getting multiple paths between the two words. In this circumstance, similarity justification can be done only through the finding the shortest path exists between two words among the collection of possible senses. But this strategy has encountered many flaws when most of the links generated in the taxonomy are returns unique distance values.

To derive the sentence similarity for each given sentence, the raw vector is generated with the link of a lexical database and the respective word order vector is produced for each sentence. The sentence can consists of words which can yields different sense to the whole sentence and the weight of the words can be obtained from the corpus. The importance of word is weighted by using information content and derived from the corpus. The purpose of these tasks [5] [9] is to generate the semantic vector for each of the two sentences by combining the raw semantic vector with information content from the corpus. The semantic similarity can be computed with reference to the semantic vectors. Similarly the order similarity between two sentences is calculated using the two order vectors. Eventually, the absolute sentence similarity between two sentences will be derived by combining semantic similarity and order similarity.

In the lexical knowledge base[1][6], it has been widely observed that words present in the upper node of the hierarchy will give more general semantics of the sentence and it implicates the less similarity to the sentence context. And the words at lower node can give more concrete semantics of the sentence and resemble more similarity. From the statements, the depth of word in the hierarchy matters a lot and it should be taken in the account. In brief, the similarity between words is determined not only by path lengths but also by depth.

5. Estimating the Path Length

The semantic similarity between two words can be measured by assigning the values and the values can be assigned by underpin assumptions [2] [8]. If the given two words convey exactly the same sense, then the value

will be set to 1. For instance, the word set w1 and w2 are fall in the same synset, then the literal meaning is that both the words imply the same meaning. Therefore, the weight for the matched word set will be assigned the value 0. On contrary to this effect, if the two words find no similarity with one another and get total contradiction, then the value will be set to 1. Suppose w1 and w2 are different synset, but the synset for w1 and w2 share one or more common words, which indirectly indicates that w1 and w2 partially contribute the same features. Therefore we assign the semantic path length between w1and w2 to 1. In some case, the similarity between two words is more or less conveying the some relevant sense, and then the value will be of the interval between 0 and 1. If both the words (w1, w2) are not finds its ways in same synset or don't share any common words, then it has been decided to count the total path between two words w1 and w2. For example, when the path length in the lexical knowledge source step down towards 0, then it would be observed that the similarity of the sentence is increasing towards the limit 1 and the path length increases constantly, then the similarity values can decreases to the limit 0.

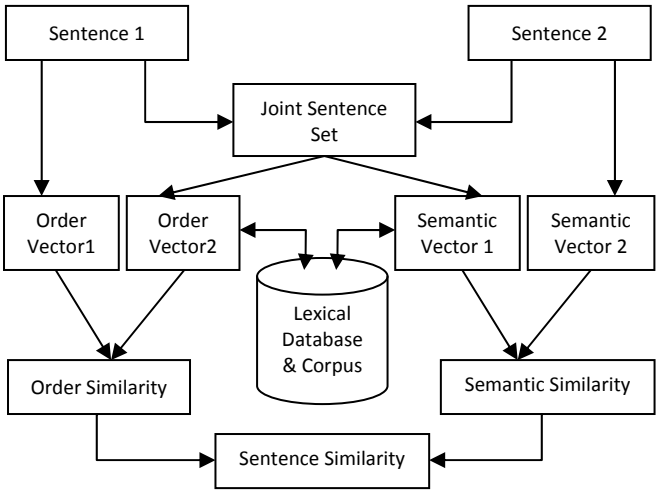

Fig 2: Calculating the semantic similarity between sentences

6. Calculating the Semantic Similarity Between Two Sentences

When two sentences [s1, s2] are given, the aim is to quantify the similarity of s1 and s2 which resulting in a similarity score [estimate the similarity between the two sentences]. It has been determined by weighing the highest score of the two sentences [9] [12]. "The maximum similarity between two sentences are adjudged by obtaining the high score levels". The sentence similarity can be derived from congruent similar values among collective word pairs. However each word in the sentence can yield different meaning to the same sentence but the seminal role of each word can be retrieved from the corpus where it has been assigned suitable weights to corresponding words. When two sentences (T1, T2) are given, a joint word set is denoted as

T : T1 U T2.

T1: A glass of cider.

T2: A full cup of apple juice.

T: A glass of cider full cup apple juice.

The joint word set T includes unique words from T1 and T2. Though the joint word set is purely retrieved from the compared sentences, it is absolutely condense with no redundant information. The joint word set, T, can be observed as the semantic information for the compared sentences. The vector retrieved from the joint word set is termed as the lexical semantic vector, referred as **ŝ**. Each occurrence of semantic vector with respect to a word in the joint word set is proportionally equals the total number of words in the joint word set.

The essential step for obtaining the semantic vector of T1 and T2 can be formulated from T and corpus statistics. At the foremost task, we have to set up the semantic vector for T1 in tandem with T and build the matrix in the given sentence order. To build the essential semantic vector for T1, the following cases should be noted:

1. If w_i presents in the sentence, then **ŝ** assigned 1.
2. If w_i is not occurs in T1, then the semantic similarity score is calculated from w_i to each word in the sentence T1.

Hence, the similarity existence for a word in T1 to w_i is getting the highest similarity score [4]. The information content of a word is derived from its probability in a corpus.

Finally, the value of an entry of the semantic vector is:

$$S_i = s.I(w_i).I(\widehat{w_i})$$

where w_i is a word in the joint word set, $\hat{w}_i$ is its associated word in the sentence. The purpose of $I(w_i)$ and $I(\hat{w}_i)$ gives the high regards to the two words to contribute the similarity which has its base on its every information contents.

Information content of w in the corpus is defined as

$$I(w) = 1 - \frac{\log(n+1)}{\log(N+1)}$$

where N is the total number of words in the corpus, n is the frequency of the word in the corpus.

The semantic similarity between two sentences can be well defined by taking the cosine coefficient between two standard vectors:

$$Ss = \frac{s1.s2}{|s1|.|s2|}$$

7. Computing the Word Order Similarity

The word order vector is the best way to represent the structural characteristics of any two sentences. The sheer goal of word order is to estimate the exact similarity exists between in two sentences. The actual word order similarity can be well scrutinized and determined by the normalized variation of word order [8] [15]. For example, let us take a pair of sentences (T1, T2), in which assign a unique index number for each word in T1 and T2. The index numbers are set to the words the way in which it has present in the sentence.

In computing the word order similarity, a word order vector r, is formed for T1 and T2 respectively, based on the joint word set T. For taking T1, for each word w_i in T, we try to find the same or the most similar word in T1 as follows:

1. Suppose, the duplicated word is well occupied in T1, then it has to be filling the word in r1 according to the index number from T1. Else, get the suitable equivalent sense of the word from T1.
2. Whereas the similarity exists among the word pair ($w_i, \hat{w}_i$) is larger than the threshold value, then for each w_i in r1 has to be mapped with corresponding index number of $\hat{w}_i$ in T1.
3. Otherwise fill the value 0 to every entry of w_i in r1(if both cases get fails).

The word order vectors are derived as :

r1 : { 1 2 3 4 2 2 4 4}

r2 : { 0 3 0 6 2 3 5 6}

After getting the relevant word order vectors, we need to measure the word order similarity of two sentences. The equation for measuring the word order similarity of two sentences as:

$$Ss = 1 - \frac{|r1 - r2|}{|r1 + r2|}$$

The metric applied for obtaining word order vector for two sentences indicate that computational method for sentence similarity should also gives the greater impact of word order. The derived word order similarity measures the number of different words as well as the number of word pairs in a different order.

8. Sentence Similarity Construction

Eventually, to estimate the best suitable method to measure the absolute similarity between sentences demands three parameters: an initial set value for deriving the semantic vector, a value for forming the word order vector and a normal factor δ for weighting the significance between semantic and syntactic information[10][17]. Hence both semantic and syntactic information play a role in conveying the meaning of sentences. The complete sentence similarity is determined by combination of semantic similarity and word order similarity:

$$S(T1, T2) = \delta S_s + (1 - \delta) S_r$$

9. Crawling the web pages

A web crawler can take the initial step by choosing the set of well-selected sentences which are all holding the semantic similarity score above 0.7 and maintain the frontier to search for more relevant web pages. The web crawler starts its process from the given sentences and recursively search for the linked web pages. The frontier maintains the list of all hyperlinks from downloaded pages for all the well-selected sentences which satisfies the threshold semantic similarity value 0.7 and above. The performance of the crawler strongly depends on the crawling strategy, i.e., the way the frontier is ordered.

The major purpose of post-processing results in the web crawler is to adapt them for the task of retrieving relevant websites. Now, the task is to cluster the hyperlinks which are all shared for more than one query and bring them to the forefront to prepare for the ranking of web pages. The clustering [6] [19] can be performed by which solely mapping the similar hyperlinks exist between different queries. After identify the similarity between hyperlinks, ranking can be performed on the occurrences of hyperlinks pertained to the query. Taking all the aspects in consideration, sorting the hyperlinks to make the process highly relevant and yield the results which are absolutely user interested.

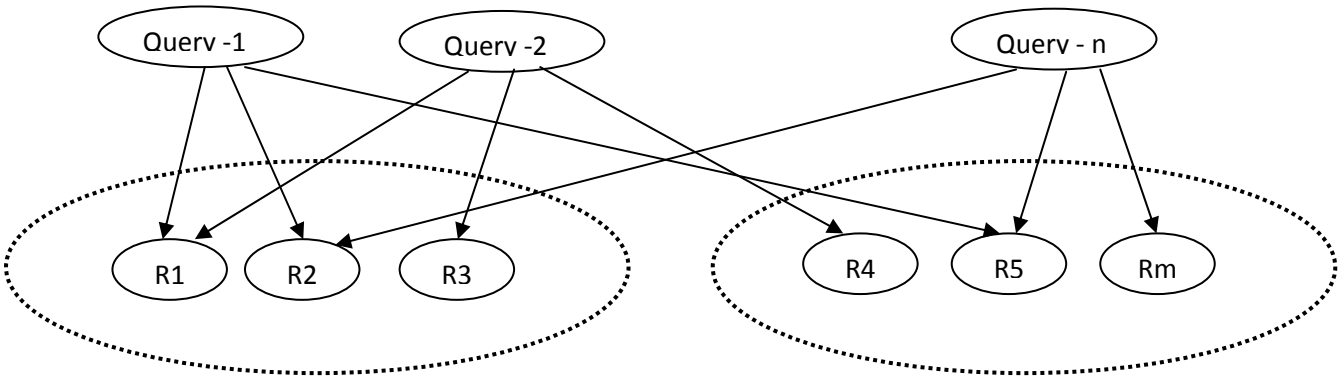

Fig 3: Clustering the web pages on the sentence level

10. The empirical analysis against conventional query expansion methods

The proposed solution has widely been compared with conventional query expansion techniques and marked the considerable improvement in the result sets. We have implemented a prototype for query expansion which uses the domain terms to extend the queries and the ontological information is also applied to augments the whole process. The expanded queries can run independently and yields the semantically matched sentences with appropriate weight and suit with proper sense. To test the performance of our proposed work, we have chosen a TREC and other standard resources for quantitative comparison. For the given query, the system has to accumulate the equivalent query set which have all extracted from popular search giants. Upon heap up of query collections, three phases of the validation method were applied on the candidate query sets in sequential manner. The similarity score were computed for all the query sentences which has obtained the scores higher than threshold similarity score (0.5). These sentences were passed as input to the second phase of validation where entity types of the candidate query sets were matched with the expected entity type of the query.

Finally, the proposed method has displayed the tallied results sets with the answers judged as correct by the human assessors. Precision and recall have been widely used parameters for measuring the performance of an information retrieval system. Higher recall indicates that the system is able to retrieve higher number of relevant documents (or sentences here) from the knowledge base while higher precision reflects the better accuracy of informational retrieval process.

For evaluation, we have compared our proposed results against the most popular search giants and shown the comparison chart in Figure 5. In our experiments, we have chosen the candidate query which has absolutely satisfies the average threshold limit (0.5) and sort the candidate query list for further categorization. To test the efficiency and precision reached for our model, we have compared the results with other search engines results and reached the consensus that our model has gained better precision than others. In the Figure 5, we have given the correlation exists between scores generated by our method and other popular search giants like Google, Bind and Yahoo. As the grant chart clearly depicts the better precision obtained for our method than the other search engine results. The grant chart has also illustrates the same sense that as we observe the curve, it has always been very consistent and more than that, it's above the average threshold (0.5). Whereas other search engine results are slump down to the least score and precision can't be retained for the search results. Hence, our method generates the query results which are most suitable and very appropriate to the user intent.

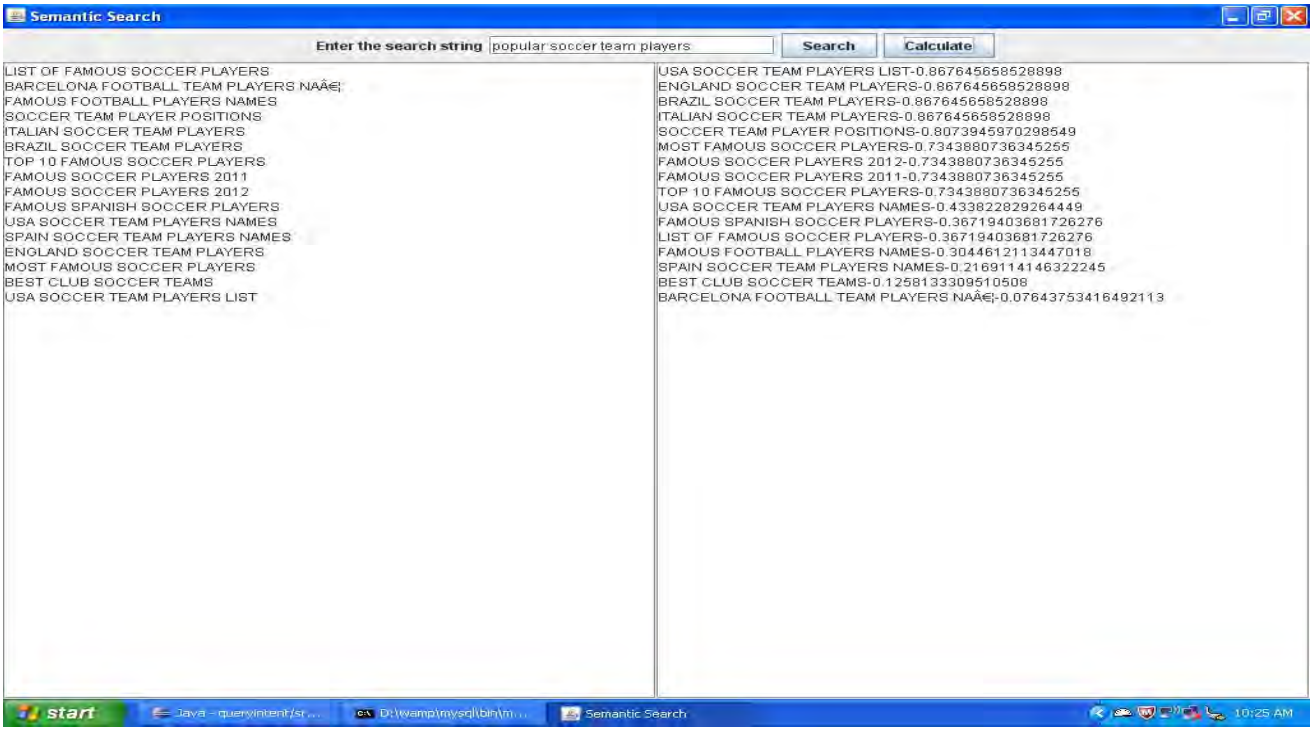

Fig 4: Reformulated query and refined query list

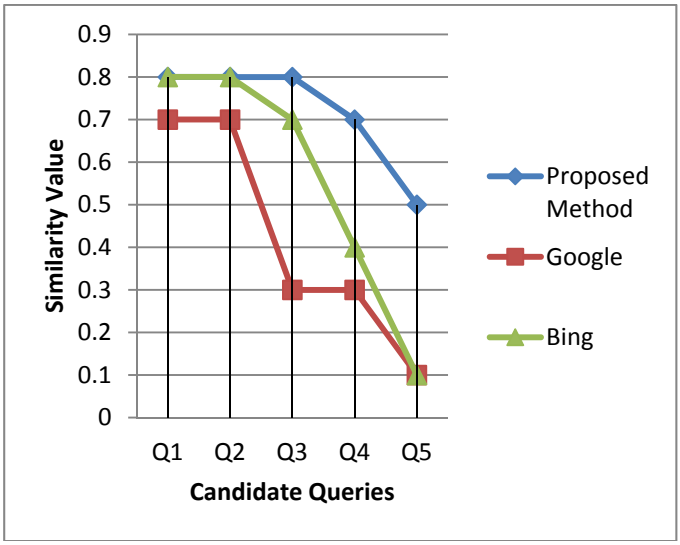

Fig 5: Comprehensive Candidate Query Analyses

11. Conclusion

We have tested our model at different levels and found that results were quite competitive when compared to empirical results obtained by task-specific methods. Our experimental results are well supported the hypothesis, as the performance of the process gets higher when both were used together rather than separately. The major advantage of this approach is that the training aspect is not computationally demanded which is due to the fact that all the computation is done at query level. Therefore, the update of the data set and related network does not imply an additional cost of re-computation. The only requirements for the method are that a meaningful distance measure be definable over nodes in the hierarchy, and that for any two profiles being compared, the sum of their scores is equal.

12. References

[1] KANG Wen-ning, YANG Zhi-iang. Research and Application of Sentence Similarity Measurement in Intelligent Answering System [J]. *Computer Technology and Development*. 2010(02):71-74.

[2] Bernard J. Jansen, Danielle L. Booth, Amanda Spink, "Determining the informational, navigational, and transactional intent of web queries", Information Processing and Management, Sept 2007.

[3] Emily Pitler, Ken Church, "Using Word Sense Disambiguation Methods to Classify Web queries by Intent", Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, pages 1428–1436, Singapore, 6-7 August 2009.
[4] ZHOU Fa-guo, YANG Bing-ru. New method for sentence similarity computing and its application in question answering system [J]. *Computer Engineering and Applications*. 2008(01):165-178.
[5] Zhang peiying. Model for sentence similarity computing based on multi-features combination [J]. Computer engineering and applications. 2010,46(26):136-138.
[6] Eui-Kyu Park, Dong-Yul Ra, and Myung-Gil Jang. Techniques for improving web retrieval effectiveness. Information processing & management, 41(5): 2005, 1207– 1223.
[7] A. Budanitsky and G. Hirst. Evaluating WordNet-based measures of semantic distance. Computational Linguistics, 2006, 32(1).
[8] Y. Li, D. McLean, Z. Bandar, J. O'Shea, K. and Crockett. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowledge and Data Eng. 2006, 18:8.
[9] E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Pasca, and A. Soroa, "A Study on Similarity and Relatedness Using Distributional and WordNet-Based Approaches," Proc. Human Language Technologies: The 2009 Ann. Conf. North Am. Chapter of the Assoc. for Computational Linguistics (NAACL-HLT '09), 2009.
[10] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-Based Explicit Semantic Analysis," Proc. Int'l Joint Conf. Artificial Intelligence (IJCAI '07), pp. 1606-1611, 2007.
[11] Turney, P. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. In De Raedt, Luc and Flach, Peter (Eds.). *Proceedings of the Twelfth European Conference on Machine Learning (ECML-2001)*, 491-502, Freiburg, Germany.
[12] Baeza-Yates, R., Ciaramita, M., Mika, P., Zaragoza, H.: Towards semantic search. Natural Language and Information Systems pp. 4, 11, 2008.
[13] Baroni, M., Zamparelli, R.: Nouns are vectors, adjectives are matrices: representing adjective-noun constructions in semantic space. In: Proceedings of EMNLP 2010. pp. 1183{1193. EMNLP '10, Stroudsburg, PA, USA, 2010.
[14] Turney, P.D., Pantel, P.: From frequency to meaning: Vector space models of semantics. Journal of arti_cial intelligence research 37, 141, 2010.
[15] O. Egozi, S. Markovitch, E. Gabrilovich, Concept-based information retrieval using explicit semantic analysis, ACM Transactions on Information Systems (TOIS) 29, 2011, article n. 8.
[16] S. Hassan, R. Mihalcea, Semantic relatedness using salient semantic analysis, in: Proceedings of AAAI 2011 (25th AAAI Conference on Artificial Intelligence), San Francisco, CA, pp. 884–889.
[17] M. A. Alvarez, S. J. Lim, A graph modeling of semantic similarity between words, in: Proceedings of ICSC 2007 (1st International Conference on Semantic Computing), Irvine, CA, pp. 355–362.
[18] F. Suchanek, G. Kasneci, G. Weikum, Yago: A large ontology from Wikipedia and WordNet, Journal of Web Semantics 6 (2008) 203–217.
[19] S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In Proceedings of IJCAI 2003