International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 4, June - 2012

QUERY CLASSIFICATION BASED ON SEMANTIC SIMILARITY

Indrajit Mukherjee,V. Bhattacharya
Department of Computer Science
Birla Institute of Technology
Mesra, India

SOURAV SINHA
Department of Computer Science
Birla Institute of Technology
Mesra, India

P. K. Mahanti
Department of Computer Science & Applied Statistics,
University of New Burnswick
Saint John, Canada

Abstract

The objective of this paper is to classify the query by using Query-Query Semantic Similarity algorithm (QQSSA). This can be achieved by term based expansion of the query, generating and identifying the strong concepts and use of information on type of relationships between the query and concept for the clustering of the query. This can used for the domain specific categorization of different queries and hence can achieve better information retrieval.

1. Introduction

Many search engine companies are interested in providing commercial services in response to user queries, including targeted advertisement, product reviews or any other value added services such as banking and transportation .The queries that user submit to search engine are usually short string composed of several words which are very short and inexplicit. It has become an important research topic to correctly identify the categories of user queries.

Retrieving documents in response to a user query is the most common text retrieval task. For this reason, most of the text similarity measures that have been developed take as input a query and retrieve matching documents [13]. However, a growing number of tasks, especially those related to web search technologies, rely on accurately computing the similarity between two very short segments of text. Example tasks include query reformulation (query-query similarity), sponsored search (query - keyword similarity), and image retrieval (query-image caption similarity), web search based document retrieval (query - document similarity).

If the query and document do not have any terms in common, then they receive a very low similarity score, regardless of how topically related they actually are. This is well known as the vocabulary mismatch problem. This problem is only exacerbated if we attempt to use these measures to compute the similarity of two short segments of text. For example, “USA” and “United States of America” are semantically equivalent, yet share no terms in common.

The rest of the paper is organized as follows. In Sect. 2, some of the existing techniques related to query recommendations and also focuses on different similarity measures like query-query similarity and query-document similarity. In Sect 3, we discuss about the different query expansion techniques and query classification techniques. In sec.4, we are proposing an algorithm named Query-Query Semantic Based Similarity Algorithm (QQSSA). This algorithm works on a new approach it filters out part of speech and breaks the long Query into small words and filters all possible preposition, conjunction, article, special characters and other sentence delimiters from the query. And then expand the query into logically similar word (same sense) to form the collection of similar words. Construct the Hyponym Tree for query1 and query2. And based upon some distance measure we classify the query. In Sect. 5, various experimental results are presented to demonstrate the domain specific query categorisation. Finally, Sect. 6 concludes the paper.

2. RELATED WORK

Most of the existing query-recommendation methods use the similarity measures obtained by mining. Some of the important characteristics which are used by different query recommendation techniques for measuring query similarity.

(i) the query terms
(ii) the clicked documents
(iii) the user sessions containing the queries.

(a) Query Recommendations using Clicked Documents

Baeza-Yates et al. propose a measure of query similarity and use it to build methods for query expansion [6]. Their technique is based on a term-weight vector representation of queries, obtained from the aggregation of the term-weight

Vectors of the URLs clicked after the query. Wen et al. also present a clustering method for query recommendation [7].

(b) Query Recommendations using Query Reformulations

Fonseca et al. propose a query recommendation system based on association rules [8]. Zhang and Nasraoui, attempts to extract information from the query log who represent each user session by a complete graph where consecutive queries are connected with an edge of a predefined weight d [9]. Non-consecutive queries are connected by an edge weighted with the product of the weights on the walk connecting them.

(c) Query Recommendations using Query Templates

Idan Szpektor et. al. introduces with concepts of rules between query templates and the query-template flow graph as an abstraction and a generalization approach for relations between queries[5]. Their novel approach is useful for addressing the long tail of rare or previously unseen queries in various search-related tasks.

Query expansion is a common technique used to convert an initial, typically short, query into a richer representation of the information need [1,2,3,4]. This is accomplished by adding terms that are likely to appear in relevant or pseudo relevant documents to the original query representation. With query expansion, the user is guided to formulate queries which enable useful results is obtained. The main aim of query expansion (also known as query augmentation) is add new meaningful terms to the initial query. This process of adding terms can either be manual, automatic or user-assisted. Manual query expansion relies on user expertise to make decisions on which terms to include in the new query.

The work based on Term Based Query Expansion chooses expansion terms from past user queries directly, rather than using them to construct sets of full text documents from which terms are then selected. The method consists of three phases: ranking the original query against the collection of documents; extracting additional query terms from the highly ranked items; then ranking the new query against the collection. Another suggested method for finding relations between queries and phrases of documents based on query logs. They use the hypothesis that click through information available on search engine logs represents an evidence of relation between queries and documents chosen to be visited by users. This evidence is called cross-reference of documents. Based on this evidence, the authors establish relationships between queries and phrases that occur in the documents chosen. These relationships are then used to expand the initial query or to give query suggestions. This approach can also be used to cluster queries extracted from log files. These clusters are used in question answering systems to find similar queries.

Dou Shen et.al proposes a novel solution for classifying Web queries into a set of target categories, where the queries are very short and there are no training data [12]. In our solution, an intermediate taxonomy is used to train classifiers bridging the queries and target categories so that there is no needto collect the training data.

Mingxia Gao et.al proposes a concept weight vector algorithm which takes into consideration the semantic structure of Ontology information [13]. The algorithm parses input information and preliminary results are based on keywords into set of concepts and it then builds weight vector according to influence of set of concepts on whole Ontology semantic. At last it deals with corresponding weight vectors as resultant vectors according to concepts matching. The resultant vector's sum will be seen as measure value in order to filter irrelevance Ontologies and order remainder Ontologies.

3. SOME DEFINITION

3.1. QUERY EXPANSION

Query expansion (QE) is thus the process of reformulating a seed query to improve retrieval performance in information retrieval operations. Query expansion involves techniques such as:

- Finding synonyms of words, and searching for synonyms
- Finding all the various morphological forms of words by stemming each word in the search query
- Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results
- Re-weighting the terms in the original query

The most popular types of Query Expansion are classified in the following broad categories as follows:

3.1.1. CONCEPT BASED QUERY EXPANSION

In concept based query expansion, concepts are extracted by analyzing and locating cycles in a special type of query relations graph. This is a directed graph built from query relations mined using association rules [3,4]. The concepts related to the current query are then shown to the user who selects the one concept that he interprets is most related to his query. This concept is used to expand the original query and the expanded query is being processed instead of the original query. The use of association rules to mine query relations, the building of a query relations graph for identifying strong concepts (or entities), the use of information on the type of relation between a query and a concept selected to improve retrieval, and the fact that all feedback the user has to provide is simple and intuitive. Once we have identified a set of concepts related to the user query, it is important to determine which concept better satisfies the user information need. It is also important for the user to select the concept that better suites his information need at the moment.

3.1.2. TERM BASED QUERY EXPANSION

In Term Based Query Expansion is done by selecting those terms that are related to the query terms and help in increasing retrieval efficiency [1]. Such terms might be synonyms, stemming variations, terms that co-occur with query terms or terms which are close to query terms on text. The process involves the following steps as follows:

(1) Term Selection
(2) Construction of Thesaurus
(3) Term Weightage

Some of the important criteria for term selection include: Simple use of co-occurrence data, Document classification, and syntactic context and Relevance feedback.

3.1.3. ONTOLOGY BASED QUERY EXPANSION

In Ontology Based Query Expansion, we obtain term suggestions from the knowledge model in contrast to the relevance feedback model which rely on having a reasonable set of relevant documents from which to suggest suitable terms [2]. With the ontology-based query expansion approach, sample size is not needed. The newly suggested terms still need to have weightings attached for the ranking algorithm.

Ontology Based Query Expansion is based on getting information from a community of users who share the same interest as the searcher and an ontology is usually a 'collective' representation of a domain which has been derived from a community of domain-specialist users.

3.2. QUERY CLASSIFICATION

The aim of query classification is to classify a user query Q_i into a list of n categories $c_{i1,}$ c_{i2}, …c_{in},where c_{ij} selected from set of N categories {c_{i1}, c_{i2},….c_{in}} [10]. Among the output c_{i1} is ranked higher than c_{i2} and c_{i2} higher than c_{i3} and so on. The queries are collected from real search engines submitted by Web users. The meaning and intension of the queries are subjective.

Typically, Web users submit a short Web query consisting of a few words to search engines. Because these queries are short and ambiguous, how to interpret the queries in terms of a set of target categories[2] has become a major research issue. The query classification problem is not as well-formed as other classification problem such as text classification. The difficulties include short and ambiguous queries and lack of appropriate training data

3.2.1. QUERY CLASSIFICATION BY CATEGORY

The query classification problem can be converted to a traditional text classification problem by finding some training data online for each category in the target taxonomy. Our method of collecting the training data is by finding documents in certain intermediate taxonomies that are found online. To do so, they need to construct mapping functions between the intermediate categories and the target categories. Given a certain category in an intermediate taxonomy,

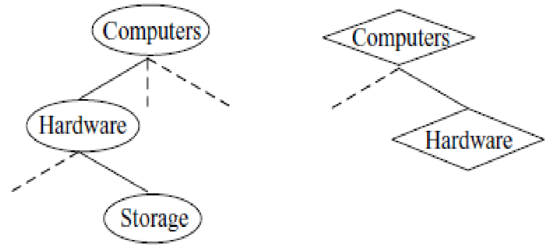
We say that it is directly mapped to a target category if and only if the following condition is satisfied: one or more terms in each node along the path in the target category appear along the path corresponding to the matched intermediate category. For example, the intermediate category "Computers\Hardware /Storage" is directly mapped to the target category "Computers\Hardware" since the words "Computers" and "Hardware" both appear along the path *Computers → Hardware → Storage* as shown in Figure . They call this matching method *direct matching*.

After constructing the above mapping functions by exact word matching, we may still miss a large number of mappings. To obtain a more complete mapping function, we expand the words in the labels of the target taxonomy through a thesaurus such as the WordNet [14].

Some solutions require human intervention in the mapping process [15]. For example, the keyword "Hardware" is extended to "Hardware & Devices & Equipments". Then an intermediate category such as

ISSN: 2278-0181
Vol. 1 Issue 4, June - 2012

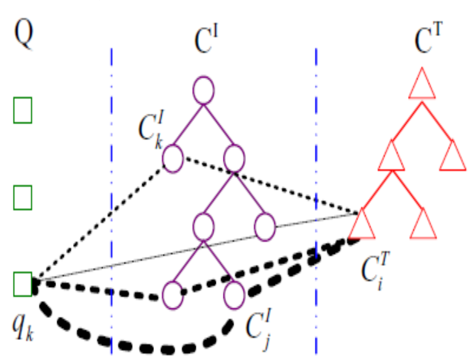
"Computers\Devices" can now be mapped to "Computers\Hardware". This matching method is called *extended matching.*



(1) Intermediate Taxonomy (2) Target Taxonomy

3.2.2. QUERY CLASSIFICATION BY BRIDGES

Taxonomy-Bridging Algorithm new QC approach called taxonomy bridging classifier, or bridging classifier in short, by which we connect the target taxonomy and queries by taking an intermediate taxonomy as a bridge [11]. The idea is illustrated in Figure , where two vertical lines separate the space into three parts. The square in the left part denotes the queries to be classified; the tree in the right part represents the target taxonomy; the tree in the middle part is an existing intermediate taxonomy. The thickness of the dotted lines reflects the similarly relationship between two nodes. For example, we can see that the relationship between $C^T{}_i$ and $C^I{}_j$ is much stronger than that between $C^T{}_i$ and $C^I{}_k$. Given a category $C^T{}_i$ in the target taxonomy and a query to be classified q_k, they can judge the similarity between them by the distributions of their relationship



to the categories in the intermediate taxonomy.

Figure 3: Illustrate of the Bridging Classifier

In order to improve search precision by semantic analysis the paper proposes concepts-weights vectors matching algorithm (CWVMA). The algorithm firstly parses input messages and preliminary keywords based results into concepts sets. Then it decides matching rule according to influence of concepts set on Ontology semantic and creates weight vector by matching rule. At last it obtains result vector as foundation of measure similarity between input messages and preliminary results. .

3.2.3. QUERY CLASSIFICATION BY CONCEPT WEIGHT VECTORS

Concepts--weight vectors matching algorithm [12]

Definition 1 (weight vector): An Ontology is made up of multiple terms (which are called concepts) that are related and constrained by various structural frameworks. Ontology of n concepts is mapped into the vector ($r_1, r_2, \ldots, r_n$) by matching rule, $r_i \in [0, 1]$. The value of r^i denotes the influence of the i^{th} concept on whole Ontology semantic and is decided by matching rule. The vector $(r_1, r_2, \ldots, r_n)$ is called weight vector.
Definition 2 (concepts--weights vectors): Concepts set $(C_1, C_2, \ldots, C_n)$, where C_i is the i[th] concept of the Ontology, and corresponding weight vector ($r_1, r_2, \ldots, r_n$) are named concepts--- weights vectors together.
Definition 3 (benchmark Ontology): Given an Ontology, it is a standard by which other Ontology can be measured or judged.
Definition 4 (evaluating Ontology): An Ontology that needs to compare with benchmark Ontology.
The basic idea of the algorithm is to estimate semantic similarity between the input message and preliminary keywords based results (They are regarded as benchmark Ontology and evaluating Ontologies respectively). The algorithm is the following.

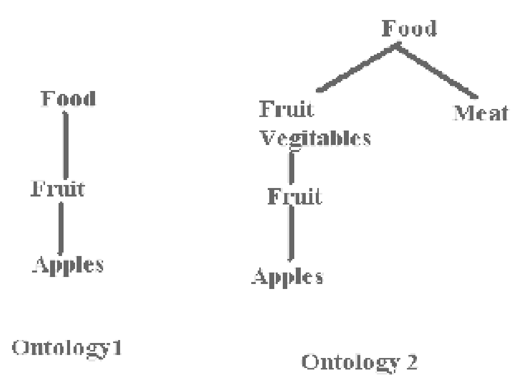
<u>**Concepts-weights vectors matching algorithm**</u>

Input: benchmark Ontology:Ontology1,evaluating Ontology:Ontology2
Output: result vector ($R_1, R_2, \ldots, R_m$)
parse Ontology1 and Ontology2 into ($I_1, I_2, \ldots, I_m$) and ($C_1, C_2, \ldots, C_n$) ;
create corresponding weight vectors ($t_1, t_2, \ldots, t_m$) 1 2 m and ($r_1, r_2, \ldots, r_n$) by matching rules;

for all I_i in ($I_1, I_2, \ldots, I_m$) do
compare I_i with C_j in ($C_1, C_2, \ldots, C_n$) ;
if $I_i = C_j$ then $R_i = t_i \times r_j$;
else $R_i = 0$;
end if
end for
output result vector ($R_1, R_2, \ldots, R_m$) .

R_i in result vector (R_1 ,R_2,...,R_m) can express semantic similarity between some concept of evaluating Ontology and the i[th] concept in benchmark Ontology.
Sum of result vector (R_1 ,R_2,...,R_m) , that is m $\sum_m Ri$, can denote semantic similarity between evaluating Ontology and benchmark Ontology. So it can be seen as value of similarity between evaluating Ontology and benchmark Ontology. For example: Two Ontology snippets are the following: They are parsed into concepts sets (food, fruit, apples) and (food, fruit-Vegetables, meats, fruit, apples). Weight vectors acquired by the distance-root rule are (1,0.75,0.5)and(1,0.75,0.75,0.5,0.25)respectively.
We account result vector (1, 0.75*0.5, 0.5*0.25) by the above algorithm. The result indicates similarity between "food" in Ontology 2 (evaluating Ontology)

For example: Two Ontology snippets are the following



4. PROPOSED WORK

Our Proposed work is to develop an algorithm for query-query similarity by expansion of the query for all possible senses. We learn our system by using of machine learning techniques for set of queries and use folding and naive Bayesian classifier for the pre-processing and training of the system for set of queries. In our algorithm, we use semantic distance measures of Wordnet to compute the query similarity. We name this algorithm as Query-Query Semantic Based Similarity Algorithm (QQSSA) .

Pseudo code of the algorithm is as follows:
Input - two user defined query Q1and Q2

Output - Similarity Measure based on semantic distance.

Assumption:
(1) pos - parts of speech
(2) dis – semantic distance measures
(3) BestPos – Best possible parts of speech
(4) AvgDis – Average semantic distance measures
(5) T_1,T_2,T_3.........T_n – n number of conceptual terms
(6) t_1,t_2,t_3....t_n - n number of conceptual terms.
(7) S_1,S_2,S_3..S_n – n number of expanded terms forming Synset of Query Q1
(8) s_1,s_2,s_3..s_n – n number of expanded terms forming synset of Query Q2

Function:
1. getpos(): returns best possible parts of speech in the above case returns common noun
2. getdis(): returns the semantic distance value based on taxonomical shortest path
3. getBestPos():Finds the most-common part-of-speech for the word, according to its polysemy count, returning the pos for the version of the word with the most different senses. single-char String for the most common part of speech ("a"=adjective, "n" = noun, "r" = adverb, "v" = verb), or null if not found.

4. getCommonPararent(): Returns String[] of Common Parents for 1st senses of words with specific 'pos' or null if not found.

5. getDistance() - Returns the min distance between any two senses for the 2 words in the wordnet tree(result normalized to 0-1 with specific pos, or 1.0 if if either is not found.

Algorithm-
(1) Obtain the user defined query from the input.
(2) Break the given user query into smaller possible set of term of words Q1(T1,T2,T3,T4) and Q2(t1,t2,t3,t4).

(3) Filter the terms by removing all possible preposition, conjunction, article, special characters and other sentence delimiters.
(4) Expand the query into logically similar word (same sense) to form the synset (collection of similar words).
(5) Obtain length of synsets (similar words) of the two user defined query.
(6) Construct the Hypernym Tree for query1 and query2.

(7) for all terms in synsets of Query 1
    for all terms in synsets of Query 2
      pos = getBestpos(query1)
      dis = getDistance(query1, query2,pos)
      parent = getCommonParent(query1, query2
            ,pos)
      if(parents is not null)
        output the parents.
      endif
    end for
  end for

(8) Construct the matrix for the relationship between the elements

Elements of synset of Q1(S1,S2,S3).
Elements of synset of Q2(s1,s2)
Semantic Distance is stored in matrix.

$$\begin{array}{cc} & \text{Query1} \\ \text{Query2} & \begin{matrix} dis(S1,s1) & dis(S2,s1) & dis(S3,s1) \\ dis(S1,s2) & dis(S2.s2) & dis(S3,s2) \end{matrix} \end{array}$$

(9) Compute the average distance measure of synsets in query 1 with respect to synsets in query2. For instance, average distance between query1 and query2
avgDist=

$$\frac{\min\{(S1,s1),(S2,s1),(S3,s1)\}+\{(S1,s2),(S2,s2),(S3,s2)\}}{Number\ of\ elements\ in\ synset\ of\ Query2}$$

(10) If avgDist is less than threshold and avgDist is not zero
    Output Query are nearly similar.
  Else if avgDist is zero
    Output Query are similar.
  Else
    Ouput Query are not similar.

  EndIf

5. RESULTS AND DISCUSSIONS

We applied the above algorithm for 15000 queries, where each of the queries contains nearly average of 3-4 terms respectively. We had conducted simulation in order to evaluate our Query-Query Semantic Similarity Algorithm (QQSSA). After several simulation runs, it was found that for different queries which were relatively similar to each other had their semantic distance between the range of 0.8 to 1 and based on the semantic distance, we construct the domain specific ontology for each of the query terms. Few results of our approach are tabulated in the below table

Table 1: Query Classification and Clustering based on Semantic Distance.

Domain	Query	Semantic Distance
Animal	Animal	1.0
	Dog	0.7905864
	Cat	0.7482143
	Pig	0.6732027
	Goat	0.7591037
Bird	Bird	1.0
	Parrot	0.9166667
	Hen	0.9333334
	Owl	0.8461539
	Pigeon	0.8201059
Education	Education	1.0
	Examination	0.6558296
	School	0.8777345
	Course	0.8834657
	Class	0.8465459
History	History	1.0
	Etymology	0.909091
	Past	0.875
	Life	0.8717825
	Recital	0.9

The query terms with semantic distance of 1.0 indicates that they are similar and those with semantic distance of 0.0 indicates that they are dissimilar.

Table 2: Domain Categorization based on Semantic Distance using our proposed algorithm & Latent Semantic Analysis (LSA) for query "Biography"

User-query	Domain	Proposed Algorithm	LSA
Biography	Animal	0.1661	0.00
	Bird	0.1024	0.00
	Education	0.4238	0.03
	History	0.8999	0.32

Suppose the user fires a new query to the system say for instance "Biography" for which the domain is unknown. After applying our algorithm, we compute the semantic distance for determining the domain of the query. The greater is the semantic distance, greater is the semantic similarity. Hence after applying our algorithm, we find semantic distance of "Biography" with respect to domain "History" is 0.8999999 whereas LSA gives similarity result of 0.32 and thus we conclude that "Biography" belongs to the domain "History".

Table 3: Domain Categorization based on Semantic Distance using our proposed algorithm & Latent Semantic Analysis (LSA) for query "Predator"

User-query	Domain	Proposed Algorithm	LSA
Predator	Animal	0.9166	0.49
	Bird	0.7236	0.03
	Education	0.1739	0.01
	History	0.2666	0.06

Hence after applying our algorithm, we find semantic distance of "Predator" with respect to domain "Animal" is 0.9166 whereas LSA gives similarity result of 0.49 and thus we conclude that "Predator" belongs to the domain "Animal".

Table 4: Domain Categorization based on Semantic Distance using our proposed algorithm & Latent Semantic Analysis (LSA) for query "Assignment"

User-query	Domain	Proposed Algorithm	LSA
Assignment	Animal	0.4166	0.02
	Bird	0.5365	0.00
	Education	0.6636	0.01
	History	0.5894	0.11

Hence after applying our algorithm, we find semantic distance of "Assignment" with respect to domain "Education" is 0.6636 whereas LSA gives similarity result of 0.01 and thus we conclude that "Assignment" belongs to the domain "Education".

Table 5. Domain Categorization based on Semantic Distance using our proposed algorithm & Latent Semantic Analysis (LSA) for query "Poultry"

User-query	Domain	Proposed Algorithm	LSA
Poultry	Animal	0.738	0.1
	Bird	0.9069	0.09
	Education	0.5312	0.01
	History	0.4716	0.03

After applying our algorithm, we find semantic distance of "Poultry" with respect to domain "Bird" is 0.9069 whereas LSA gives similarity result of 0.09 and thus we conclude that "Poultry" belongs to the domain "Bird".

Hence this approach can be very useful in the construction of a domain specific ontology and hence improve the information retrieval from large datasets.

6. CONCLUSIONS AND FUTURE WORK

From series of simulation, we conclude that the queries with semantic distance between the range of 0.8 to 1.0 can be clustered together to form a sub-domain of the given domain. In case the domain is unavailable, the seed query can be used to create the domain. Subsequent queries with semantic distance between the range of 0.8 to 1.0 can be again clustered together to form a new sub-domain. Our proposed algorithm can be used in the process of query expansion to compute query-query similarity and grouping the similar queries into a cluster based on user log based analysis for domain or ontology specific query categorisation. Hence it can be used by search engine for generating nearly accurate results.

REFERENCES

[1] H.Imranand A.Sharan,"THESAURUS AND QUERY EXPANSION", *International Journal of Computer science & Information Technology, vol1 No 2,2009*,pp:89-97

[2] J. Bhogal,A. Macfarlane, P. Smith, "A review of ontology based query expansion", *Information Processing and Management 43,Science Direct*,2007, pp :866–886

[3] Yonggang Qiu, H.P. Frei, "Concept Based Query Expansion" ,*Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval,ACM,1993*

[4]Bruno M. Fonseca , Paulo Golgher,Bruno Pôssas , Berthier Ribeiro-Neto, Nivio Ziviani, "Concept-Based Interactive Query Expansion" *Proceedings of the 14th ACM international conference on Information and knowledge management,ACM,2005*

[5] Idan Szpektor , Aristides Gionis , Yoelle Maarek , "Improving Recommendation for Long-tail Queries via Templates" , *WWW '11 Proceedings of the 20th international conference on World wide web*,2011

[6] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza, "Query recommendation using query logs in search engines", In EDBT Workshops, 2004.

[7] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Clustering user queries of a search engine", *In WWW*, 2001

[8] B. M. Fonseca, P. B. Golgher, E. S. De Moura, and N. Ziviani , "Using association rules to discover search engines related queries", *In LA-WEB*, 2003

[9]Z. Zhang and O. Nasraoui , "Mining search engine query logs for query recommendation " , *In WWW*, 2006.

[10]Yu, J. and Ye, N. : "Automatic Web Query Classification Using Large Unlabeled Web Pages", *Proceedings of Web-Age Information Management IEEE Computer Society Washington*, DC, USA2008.

[11]Shen, D. and Sun, J. T., "Building Bridges for Web Query Classification" *Proceedings of SIGIR Information Retrieval, Seattle Washington* Aug 2006.

[12] Ga, M and Liu, C..and Chen, F., "An ontology search engine based on semantic analysis ", *Proceedings of Information Technology and Applications*, 2005.

[13] Donald Metzler,Susan Dumais, Christopher Meek, "Similarity Measures for Short Segments of Text" *Proceeding ECIR'07 Proceedings of the 29th European conference on IR research,* Springer-Verlag Berlin, Heidelberg ©2007

[14]G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to wordnet: an online lexical database. International Journal of Lexicography,3(4) pp:23–244, 1990.

[15]D. Vogel, S. Bickel, P. Haider, R. Schimpfky,P.Siemen, S. Bridges, and T. Scheffer. Classifying search engine queries using the web as background knowledge. SIGKDD Explor. Newsl., 7(2):117–122,2005.