International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.163.03

Exploratory Search for Retrieving Information from Web Pages using Synonym, Hyponym and Hypernym

Dr.R.Manjula
Associate Professor, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India

Abstract - As information retrieval in website is growing and it has very good interest in techniques to help efficiently locate deep web interfaces. The exact information is given to the user based on the curiosity. When users investigate unaware fields, they may want to learn more about a particular subject area to gain their knowledge rather than solve a specific problem. The users are satisfied with search results for the users right keywords in the search query. Highly relevant ones for a given topic to the query are ranked by the crawlers to prioritize websites. To get more accurate results, this work is searching the root word of synonym, hyponym and hypernym is identified after pre-processing the given query by user. To retrieve information about the user query which in depth manner by finding synonym, hypernym and hyponym of the query given by the user that is pre-processed. We use customized stop list and stemming to pre-process the user's query for removal of stop words. Then for those root words, term frequency and inverse document frequency are calculated and the cosine similarity is used for finding relevance between the terms in the query and documents. Thus the system used to find synonym, hypernym and hyponym for keywords to eliminate irrelevant information in the results. This system is more efficient to provide the relevant documents for the given query in very less computation time.

Keywords - Stopping keywords, Vector space model, Hypernym, Hyponym, Synonym

I. INTRODUCTION

This work developed a search engine, and which can be used for exploratory search for retrieving unaware fields for different kind of users query to satisfy their desires. The search engine existing now will provide good results only if the keywords are given correctly in search query. So we are going to develop a search engine which allows user to search even if they do not know exact keywords. This work is to provide an exploratory search for users to know about different fields using Hypernym, Hyponym, and Synonym of the keywords given in search query and to allow them to bookmark globally and locally by storing bookmarks in database [5].

The words ends with 'nyms' are used to describe different classes of words, and the relationships between words. A word which has similar meaning is called Synonym. For example, one synonym of beautiful is "pretty" however; this word carries quite a negative connotation. A word that has a more general meaning than another is called Hypernym. For example, the word color is a hypernym for red, green, blue, brown, etc.,. A word that has specific meaning is Hyponym. For example, it is the specific meaning of hyponym like red, green, blue, brown, etc.

The search engine existing now will provide good results only if the keywords are given correctly in search query. So we are going to develop a search engine which allows user to search even if they don't know exact keywords. The crawler performs site based searching in order to evade the visiting of more pages unnecessarily. The search results would be useless when the users are not giving the accurate and exact search word in the query. The information is ubiquitous and highly demand for searching the search engine has to be built for supporting sdearch behaviours beyond simple lookup.

II. RELATED WORKS

The problem in the existing system is that the users investigate unfamiliar fields the search would not be efficient and the system uses "smart crawler method" which avoids users from visiting large number of web pages.

Chang Jun Jiang has introduces an indexing network model for analyzing the relationship between various information service resources and considers the problems like heterogeneity, diversity and etc.. [2].

H.Chim and X.Deng proposed a phrase-based document similarity approach and it computes the pair wise similarities between the documents based on the Suffix Tree Document (STD) model. This work concludes that the phrase-based document similarity works better than the single-word TF-IDF similarity measure [3].

http://dx.doi.org/10.21172/ijiet.163.03

Adam L. Kaczmarek designed clustering-by-directions algorithm for interactive query expansion and it supports users of search engines in forming Web search queries. This approach is based on clustering by directions (CBD) algorithm and facilitates the users to form queries [1].

Ho Chung Wu a novel probabilistic retrieval model and it interprets the relevant decisions to TF-IDF term weights. This retrieval model is to establish about information retrieval as relevance decision making and an advanced TF-IDF related term weights for future elaborate retrieval models should be developed [8].

III. PROPOSED WORK

Our system designed for supporting the multi keyword ranked search as well as the synonym hyponym and hypernym based search and gives an efficient and flexible search. The multi keyword search and result ranking is implemented using a Vector Space Model (VSM) to build document index. In this model each document and query is expressed as a vector and dimension and weight is measured using Term Frequency (TF) of its corresponding keyword and the document index is measured using the Inverse Document Frequency (IDF) weight [7][8]. The Cosine similarity is used to find the similarity between the document and the query by calculating the angle between them[3].

.

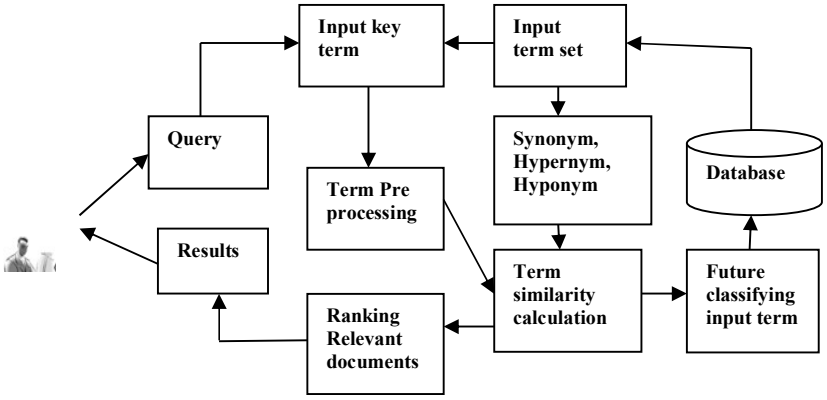

Figure.1 Architecture of proposed system

System Design is a process through which the requirements are translated into a representation of the software. The work flow of the proposed system is given in Figure1.The user has given a query through interface to the system. The input key terms of query are found and term preprocessing is done. The synonym, hypernym and hyponym of a given term are identified for both the documents and the query terms. Then the similarity calculation of terms is performed to find the relevant documents. Then according to the ranking the relevant documents are given as a result to the user

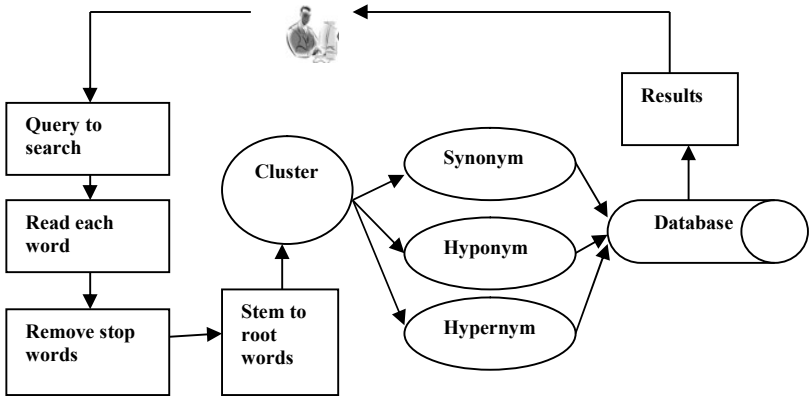

Figure.2 processing of synonym, hypernym and hyponym

To retrieve information about the query which is given by the user in depth manner by finding synonym, hypernym and hyponym of the query given by the user that is pre-processed is given Figure 2. We use customized stop list to pre-process the user's query for removal of stop words [1]. Further, successor count root words are found by using stemming algorithm for eliminating the suffix of the given query. For the obtained

http://dx.doi.org/10.21172/ijiet.163.03

root word, we find hypernym, hyponym and synonym using Natural Language Processing. Then for those root words, term frequency and inverse document frequency are calculated and based on that calculations, we have stored website links. These links are retrieved when the user selects the word to know in detail. Images are also stored for perceptual view. To know in-depth details about certain keywords and its meaning, the user gives query in the search engine where the interface acts as an intermediate between the user and the system.

A. TERM PROCESSING

For the given query, pre processing is done to obtain the root word from the given query. So that it can retrieve the data related to that root word. Time efficiency is achieved by using this technique. For the given query by user, stop word is filtered. Stop words are terms with less weighted words like 'who', 'what', 'how', 'a', 'is', 'the', etc which is removed during pre-processing. It is controlled and implemented using customized stop list. After stopping process is done, another process of stemming is implemented. Stemming of word is the removal of suffix to get the root word. Stemmer contains table which represents the relations between root forms and normal forms. To stem a word, the table is queried to find a matching word and the associated root form is returned. Eg: A stemming algorithm reduces the words "walks" and "walking" to the root word, "walk".

B. VECTOR SPACE MODEL

In vector space model the word or term is considered as vector of terms. Since the vector space has large dimension the words in the vocabulary becomes an independent dimension. The vector based systems are operating in positive quadrant of vector space, the terms are assigned with positive value. The query is a collection of terms and that is also converted into vector. Our system finds the similarity between the document and the query vectors and the numeric score is assigned to a document for a given query [4]. The angle between two vectors is measured and the difference between the vectors, and cosine of the angle is used as the numeric similarity. It is 1.0 for identical vectors and 0.0 for orthogonal vectors. Otherwise the inner-product of two vectors is often used as a similarity measure. The deviation of angle between each document vector, original query vector is compared and shown in Figure 3. Angle is calculated using "cosine" given in Eq.1.

$$cos\theta - \frac{doc}{||doc_2} \qquad \text{Eq.1}$$

Where doc_2 .q is an intersection and $|| doc_2||$ is norm, which is calculated as $|| doc2|| = \sqrt{\sum_{i=1}^{n}}$

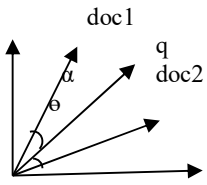

Figure 3.Cosine similarity calculation

If cosine value is equal to 0 then the query is not matched with any document. Each dimension has separate term and a term occurs in document, then the value is non zero. For ranking websites we are using vector space model as it is efficient and involves cosine similarity function to find similarity between keywords.

1. Term Frequency measures the frequency of terms that occurs in a document. Since the documents are in different size, the term would appear many times in big documents than small ones. So the term frequency is normalized by using

   TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

2. Inverse Document Frequency (IDF) is used to measures the importance of a term t in document d. For Example the terms like "is", "of", and "that", may appear many times in the document but have less importance. The inverse document frequency is calculated using

   IDF(t)=log(number of documents / Number of documents containing term t**)**

http://dx.doi.org/10.21172/ijiet.163.03

Compute cosine similarity for TF and IDF. Consider a file that contains 3 documents such as doc1,doc2 and doc3.Assume the document contains the terms "computer science ",the document doc2 has "electrical and electronics" and doc3 consist of " electronics and communication ".Here the total number of documents, N=3. Now the user has given a query "electrical and electronics" and the cosine similarity calculation is given below. The calculated term frequency and multiplication of TF-IDF for the above mentioned terms are given in Table 1 and Table 2 and 3. Here the stop words in the documents and query are removed at the time of preprocessing. The IDF value for the given terms are calculates as:

Table 1. Inverse Document Frequency (IDF)

Terms	IDF value
computer	$\log_2(3/1) = 1.584$
science	$\log_2(3/1) = 1.584$
electrical	$\log_2(3/1) = 1.584$
electronics	$\log_2(3/2) = 0.584$
communication	$\log_2(3/1) = 1.584$

Table 2. Term Frequency (TF)

Documents	communication	computer	electrical	electronics	science
doc1	0	1	0	0	1
doc2	0	0	1	1	0
doc3	1	0	0	1	0

Table 3. TF-IDF for term calculation

Documents	communication	computer	electrical	electronics	science
doc1	0	1.584	0	0	1.584
doc2	0	0	1.584	0.584	0
doc3	1.584	0	0	0.584	0

Table 4. TF-IDF for Query calculation

Query	communication	computer	electrical	electronics	science
q	0	0	1/1*1.584 =1.584	½*0.584 =0.292	0

The cosine similarity measure of each document in the collection is computed as ,

Length of doc1 = sqrt($1.584^2 + 1.584^2$) =2.240
Length of doc2 = sqrt($1.584^2 + 0.584^2$)=1.688
Length of doc3 = sqrt($0.584^2 + 1.584^2$)=1.688
Length of q = sqrt($1.584^2 + 0.292^2$)=1.610

The Similarity values of terms in the query and the document are calculated and given below.

Cosine sim(doc1,q) =0
Cosine sim(doc2,q) = 1.245
Cosine sim(doc3,q) = 0.062

The document doc1 is not having the terms in the given query so the similarity measure is zero. So that will not be appeared in the final list. The documents are listed and presented based on the similarity value. Here the document doc2 is having higher similarity value than the doc3, so ranking is given as doc2 then doc3 .The keywords and synonym, hypernym and hyponym for keywords are stored in database and the Fig.4 shows the result of synonym, hypernym and hyponym of a given term.

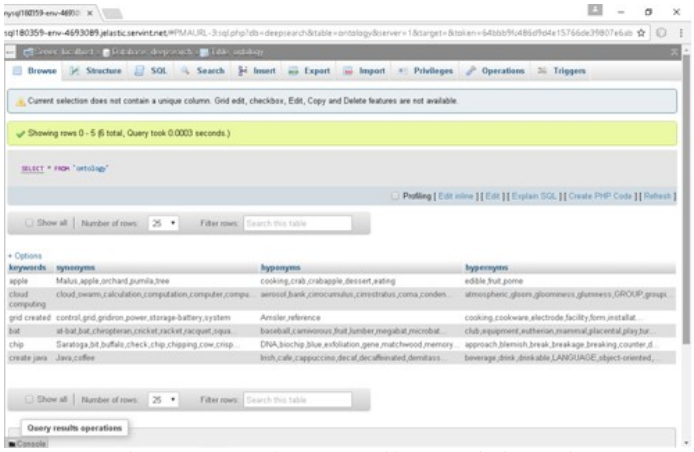

Figure.4 synonym, hypernym and hyponym for keywords

IV. CONCLUSION

Thus, this work hereby concludes, that this method helps to search about unaware fields in an efficient manner. This system is feasible for all kinds of user. Finding synonym, hypernym and hyponym for keywords by natural language processing is a challenging task as we have to eliminate irrelevant words and find only for similar useful words. Further to make stemming process more effective we defined a new combined strategy using existing strategies for stemming algorithm which is somewhat complex but gives high performance in retrieving relevant links. Information retrieval through images additional to keywords can be done in order to make searching more interactive and user friendly. We can make information retrieval more secured by displaying only websites which is of high security like https protocol websites alone.

REFERENCES

[1] Adam L. Kaczmarek, (2011). Interactive Query Expansion with the Use of Clustering-by-Directions Algorithm. IEEE Transactions on Industrial Electronics, 58(8): 3168 – 3173, **DOI:** 10.1109/TIE.2010.2045315.
[2] C. J. Jiang et al, (2013), An indexing network model for information services and its applications. Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference. 290–297.
[3] Hung Chim, Xiaotie Deng., (2008), Efficient Phrase-Based Document Similarity for Clustering. IEEE Transactions on Knowledge and Data Engineering, 20(9):1217 – 1229, DOI: 10.1109/TKDE.2008.50
[4] HoChung Wu., (2008), Interpreting tf-idf term weights as making relevance decisions. ACM Transactions on Information Systems (TOIS) ,26(3):1-37.
[5] HeikoPaulheim., (2013) ,Exploiting Linked Open Data as Background Knowledge in Data Mining, ceur-ws.org, 902:1-10.
[6] L. Kaczmarek., (2011), Interactive query expansion with the use of clustering-by-directions algorithm. IEEE Transactions on Ind. Electron., 58(8):3168–3173.
[7] M. T. Mills, N. G. Bourbakis., (2014). Graph-based methods for natural language processing and understanding ,A survey and analysis. IEEE Transactions on Systems, Man, and Cybernetics: Systems,44(1): 59-74.
[8] R.K. Roul, O. R. Devanand, S.K. Sahay, (2014),Web document clustering and ranking using tf-idf based apriori approach.IJCA Proceedings on ICACEA,2: 74-78.
[9] Xue, X, Zhou, Z. 2009.Distributional Features for Text Categorization, IEEE Transactions on Knowledge and Data Engineering.21(3):428-442.
[10] Zhu, H, Chen, E, Xiong, (2014), Ranking user authority with relevant knowledge categories for expert finding. World Wide Web, 17(5):1081–1107, https://doi.org/10.1007/s11280-013-0217-5.