ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

A Hybridized Model for Efficient Query-Dependent Ranking and Information Retrieval in Large Databases

Edward E. Ogheneovo*, Bunakiye R. Japheth

Senior Lecturer, Dept. of Computer Science, University of Port Harcourt, Port Harcourt, Nigeria. *

Lecturer, Dept. of Mathematics/Computer Science, Niger-Delta University, Yenagoa, Nigeria

ABSTRACT: Information Retrieval (IR) has become a topic of great interest with the advent of text search engines on the Internet. Information Retrieval (IR) is the area of study concerned with searching for documents, for information within documents, and for metadata about documents in Internet. This paper proposed a hybridized approach using KNN and VSM tools for retrieving information on the Web. The approach used standardized data to test this technique. Using the recall values of 0.1, 0.2, …, 1.0, the precision for KNN and VSM for each recall value is computed. Having obtained these values, the corresponding KNN/SVM hybridized values are then computed for ADINUL, CRANFIELD, CRN$NUL, MEDNUL, MEDLARS respectively. The performance improvement for each database collection was also computed. ADINUL is 36.4%, CRANFIELD is 47.8%, CRN4NUL is 18.4%, MEDNUL is 15.7%, and MEDLARS is 23.8%. Based on these results, it was discovered that the combined KNN/VSM retrieval models outperforms that of KNN or VSM when used separately. That is, this technique is able to retrieve information faster with significant lesser time. Thus we conclude that the hybridized KNN/VSM model is better in ranking and retrieving relevant documents than the previous techniques.

KEYWORDS: Information retrieval, query ranking, relational databases, vector spacing model, k-nearest neighbor

I. INTRODUCTION

Information Retrieval (IR) has become a very important topic with the advent of text search engines on the Internet [1]. Information Retrieval is concerned with searching for documents, for information within documents, and for metadata about documents, as well as that of searching relational databases and the Internet [2]. A text is composed of two fundamental parts; the document (book, journal paper, chapters, sections, paragraphs, Web pages, computer source code, etc.) and text terms (word, word-pair, and phrase within a document). At inception of Information retrieval, text queries and documents are both represented in a unified manner, as sets of terms, to compute the distances between queries and documents thus providing a framework to directly implement simple text retrieval algorithms [3] [4]. However, these days, information retrieval has been applied in a number of areas.

Information retrieval started with scientific publications and library records, but these days; information retrieval has spread to other forms of content, particularly those of information professionals, such as journalists, lawyers, and doctors [5]. However, there are various problems associated with providing access to information. With these challenges, the field of Information Retrieval evolved to provide various approaches for searching various forms of contents. Much of the scientific research on Information Retrieval has occurred in these contexts, and much of the continued practice of Information Retrieval deals with providing access to unstructured information in various corporate and governmental domains. These days, many universities and public libraries use Information Retrieval systems to provide access to books, journals and other documents. Web search engines are the most visible Information Retrieval applications. Therefore, information retrieval is aimed at retrieving documents relevant to the user's information needs [6].

Automated information retrieval (IR) systems [7] [8] were originally developed to help manage the huge scientific literature that has developed since the 1940s. Many universities, corporate bodies, and public libraries now use IR systems to provide access to books, journals, and other documents. Commercial information retrieval systems [9] offer databases containing millions of documents in various subject areas. Dictionary and encyclopedia databases are now widely available for personal computers (PCs). Thus information retrieval has been found useful in such disparate areas as office automation and software engineering. Therefore, any discipline that relies on documents to do its work could potentially use and benefit from information retrieval [10] [11].

An IR system matches user queries--formal statements of information needs--to documents stored in a database. A document is a data object, usually textual, though it may also contain other types of data such as photographs, graphs, etc. Often, the documents themselves are not stored directly in the IR system, but are represented in the system by document surrogates [12]. A document can be stored in its entirety in an IR database. However, it is also possible to create a document surrogate for it consists of the title, author, and abstract [13]. This is typically done for efficiency, that is, to reduce the size of the database and searching time. Document surrogates are also called documents, and in the rest of the work we will use document to denote both documents and document surrogates.

II. RELATED WORK

Wong et al. [14] model information by using vector space model. The authors proposed the Generalized Vector Space Model (GVSM). Considering the limitations associated with Boolean model of information retrieval, the GVSM is used to model information retrieval due to its sound generalization of the traditional vector space model for computing the correlation of relevant terms. First the authors explained how the elements of Boolean algebra can be modeled as vectors in a vector space and by representing terms as Boolean expression by showing whether two vectors are identical or orthogonal. They show that if two vectors are identical (i.e., not orthogonal), then the corresponding Boolean expressions have at least one minterm in common. Using the GVSM, they generalized the term vector representation such that the representation of a document is taken to be a sum of term vectors. The vector sum operator and the document is hypothesized and expressed as vector sum of the associated term vectors.

Tsatoronis and Panagiotopoulous [15] propose the Generalized vector Space Model (GVSM) for retrieving semantic information for word thesauri like WordNet. They incorporated semantic information by modifying the standard vector space model. From the experimental evaluation, a test was conducted on the performance of the semantic relatedness measure (SR) for a pair of words using three benchmark data sets from TREC collections, the experimental result shows that semantic information can boost text retrieval performance. The correlations for the three data sets show that SR performs better than any other measure of semantic relatedness. This is because the semantic relatedness measure considers all of the semantic links in WordNet such as a graph, weight edges based on type and depth by computing the maximum relatedness between any two nodes, connected via one or more paths.

Geng et al. [16] proposed K-nearest neighbor (KNN) for retrieving information by ranking the pages. The work employ KNN to rank different queries and conducted query-dependent ranking. An online method which created a ranking model for a given feature space and then rank the documents with respect to the query using the model. Then they used two offline approximation of the method to create the ranking models in advance to enhance the efficiency of ranking. They further prove that the approximations are accurate in terms of difference in loss of prediction if the learning algorithm used is stable with respect to minor changes in training examples.

Kang and Kim [17] proposed a query-dependent method for ranking model construction using K-Nearest Neighbor (KNN). They classified queries three: 1) the topic relevance task, 2) the homepage finding task, and 3) the service finding task. The classification is based on search intention and two different ranking models were tuned and used for the two categories. The work proposed a user query scheme which uses the difference of distribution, mutual information, the usage rate as anchor texts, and the print of search (POS) information for the classification. Different algorithms were and in the classification. The queries were trained and were classified into the query feature space in which each query is represented by a point. Using ranking method, they retrieved its K-nearest training queries, learn a ranking model with these training queries, and then rank the documents associated with the test query.

Guru and Nagandraswamy [18] proposed a clustering model, a similarity measure for estimating the degree of similarity between two symbolic patterns using K-mutual nearest neighborhood and mutual similarity is employed. The model uses two layer clustering strategy. In the first layer, a similarity proximity matrix for symbolic pattern based on the proposed similarity measure is obtained. A position matrix is created from the similarity proximity matrix based on the similarity rank of pattern. The K-mutually nearest neighbor’s algorithm is applied on the position matrix to obtain clusters of patterns.

III. METHODOLOGY

In retrieving information using KNN technique, the documents are classified based on their similarities. KNN is then used to retrieve the most similar k documents for each query document. This is done by ranking the candidate categories by vote using the weighted average and then using support vector machine (SVM) to predict the categories of the document as the most voted categories. This is done by finding a set of keywords that have similarity in meanings for the k-nearest neighbor. Queries with similar meanings and contain tags are then searched using closely related keywords. An algorithm is then constructed for the retrieval of documents that are sent to the database based on the queries. As soon as a neighborhood word similar to the words in the search space is established, the algorithm considers on tagged queries or tagged keywords that are used in the query and then assign weight to them by calculating the weight for each tag which we refer to as w_t (i.e., weight for each tag). The average weight of these similar keywords is the calculated and the query tag is applied. Thus the KNN algorithm for this information retrieval is based on the selection of nearest neighbors and the selection of tags. The algorithm is shown in figure 1. In algorithm 1, the input consists of both labeled and unlabeled documents and it try to calculate the similarities and relationships between nearest documents

Algorithm 1: K-Nearest Neighbor for document retrieval

```
Input: $L_d$ = labeled documents, $U_d$ = unlabeled documents, k = number of
    nearest neighbors, n = number of tags in the query considered
Output: $E_r$ = the expected result, $R_r$ = retrieved result, $T_r$ = set of required tags
for each q ∈ Q containing $U_d$ do
    $s_q = sim(d,d^1)$ //sim stands for similarity
end // end the outer for loop
Let N be K nearst neighbors to $d^1$
for each q ∈ N do
    for each t that q applied to $d^1$ do
        $w_t = w_t + \frac{Sq}{k}$
        end // end the inner loop
    end //end the inner loop
begin
    Sort tags by $w_t$
    Let $E_r$ be the expected result, $R_r$ be the retrieved result, and $T_r$ be a set of required tags
return $T_r$ , $R_r$
end
```

Fig. 1: An algorithm for retrieving information using K-nearest neighbor

K-nearest neighbor has a major drawback when used to retrieve information alone. This is because the greater percentage of its computation usually takes place after the query has been submitted to the database engine especially if a large dataset is being considered in the query since the number of similarities required to perform the calculation is considerably reduced. Based on this, we decided to combine. KNN with another tool called the vector space model. The VSM solves the problem of KNN by ensuring that the computation is done at a greater speed and lesser time so that information can be retrieved quickly. This is also complemented by the fact that the VSM model also has some drawbacks. Therefore, using these two models together helps to overcome the problems inherent in each of the models.

First, we modeled k-Nearest neighbor by providing relevant equations that can help us determine the ranking of each document. This is done by using the vector space model. Thus user-based k-nearest neighbor is often used in information retrieval that can be modified to include the query. Usually, the KNN algorithm finds a set of documents similar to user query. From these neighbors, a set of recommended terms are constructed based on these documents and the user query. Thus as soon as queries with similar keywords or terms that are the same with those in the terms are discovered in each document, the algorithm then calculates a weight for each query, say w_t , and the average similarity of the neighbors of queries having similar terms in the document. Then the recall and precision is calculated for each term in these documents. However, it must be noted that the k-NN is a lazy algorithm since the major part of its computation is done after the query has been processed. Therefore, there is need to hybridize it with another model to avoid this problem and to ensure that the performance of the technique is very efficient and produces better results than when used singly. Thus in this work, we combined k-Nearest Neighbor with Vector Space Model to take advantage of both models and reduce the problems inherent in them.

Using the vector space model, we construct a vector space algorithm that computes the similarity score and the retrieval of documents. Algorithm 2 is the vector space algorithm. Figure 2 illustrate the algorithm for retrieving information using vector space model. The input contain score and the output consist of retrieved result of the top k score

Algorithm 2: Vector space model for document retrieval

```
Input: score(N)
Output: $R_r$ = retrieved result of top k score
for each d
    do initialize Length(d) to the length of the document d
end for
    for each query term t
        do calculate $W_{t,q}$ and fetch posting list for t
        end for
    for each pair (d, $tf_t$) in postings list
        do Scores[d] ← Scores[d] = ($Wf_{t,d}$ x $w_{t,q}$)
        end for
Read the array length[d]
for each d
    do Scores[d] ← Scores[d]/Length[d]
end for
return Top K components of Scores [ ]
```

Fig. 2: An algorithm for retrieving information using vector space model

We then collected data from five document collections. These collections are: ADINUL, CRANFILED, CRN4NUL. The last two MEDNUL and MEDLARS are subsets of documents obtained from the national Library of Medicine.

Table 1 shows the summary of the contents of these document collections.

	ADINUL	CRANFIELD	CRN4NUL	MEDNUL	MEDLARS
Number of Documents	82	1398	424	400	1,038
Number of Queries	35	225	155	30	30
Ave. No. of Doc. Per query	5	8	4	9	7
Av. No. of Rel.Queries per Doc.	0.5	0.2	0.3	0.2	0.4

VI. RESULTS AND DISCUSSION

The precision for each recall value is then used for computing the performances of vector space (VSM) and k-nearest neighbor (KNN) based on the weighting index terms and document-by-term. Recall is the proportion of relevant documents retrieved while precision is the proportion of the retrieved relevant documents. Also, the precision when the hybridized model was used is also computed. The overall performance of the technique is then computed using the average number of queries per document and then computing the average precision for all the queries using the recall value of 0.1, 0.2, …, 1.0. Finally, the percentage improvement of the hybridized technique is then computed using the COSINE of angles between the query vector and document vectors. The set of precision values corresponding to each recall value are then tabulated for ADINUL, CRANFIELD, CRN4NUL, MEDNUL, and MEDLARS.

Table 2: Results for ADINUL collection with 82 documents and 35 queries

ADINUL			
		Precision	
Recall	**kNN**	**VSM**	**kNN/VSM**
0.1	0.3542	0.3812	0.4856
0.2	0.3251	0.3301	0.4627
0.3	0.3026	0.3114	0.4291
0.4	0.2647	0.2812	0.3736
0.5	0.2461	0.2501	0.3242
0.6	0.2014	0.2096	0.3013
0.7	0.1827	0.1904	0.2406
0.8	0.1465	0.1686	0.2129
0.9	0.1102	0.1227	0.1677
1.0	0.1086	0.1194	0.1428
Performance Improvement = 36.34%			

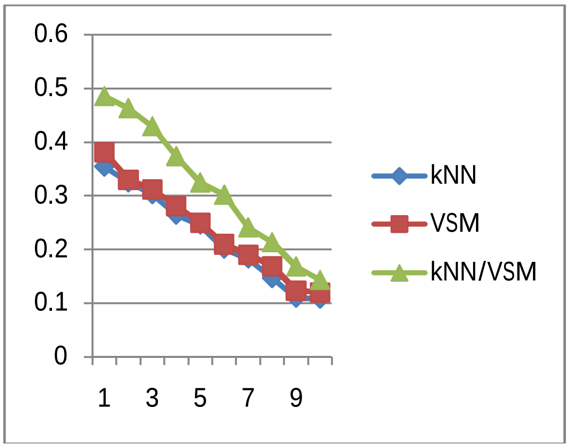

Figure 3: A graph of CRANFIELD collection with 1398 documents and 225 queries

DOI: 10.15680/IJIRCCE.2016. 0404336

Table 43: Results for CRANFIELD collection with 1398 documents and 225 queries

CRANFIELD			
		Precision	
Recall	kNN	VSM	kNN/VSM
0.1	0.3752	0.4267	0.5673
0.2	0.3537	0.4013	0.5347
0.3	0.3389	0.3731	0.4526
0.4	0.3128	0.3465	0.4728
0.5	0.3056	0.3289	0.4654
0.6	0.2893	0.3011	0.4543
0.7	0.2514	0.2872	0.3572
0.8	0.2287	0.2645	0.3639
0.9	0.1685	0.2459	0.3782
1.0	0.1357	0.2186	0.3527
Performance Improvement = 47.8%			

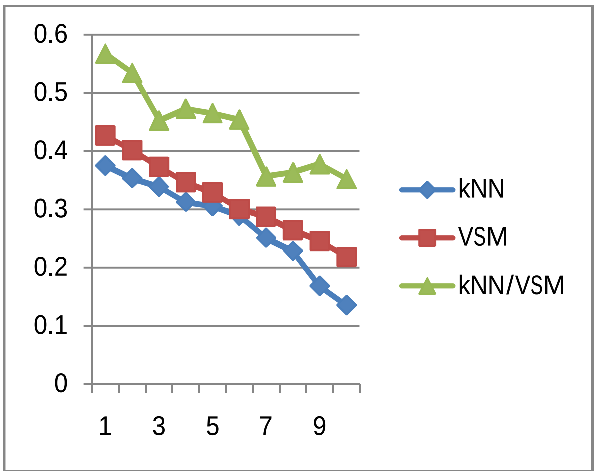

Figure 4: A graph of CRANFIELD collection with 1398 documents and 225 queries

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Table 4: Results for CRN4NUL collection with 424 documents and 155 queries

CRN4NUL			
		Precision	
Recall	kNN	VSM	kNN/VSM
0.1	0.3956	0.4562	0.4934
0.2	0.3893	0.4278	0.4713
0.3	0.3539	0.4012	0.4365
0.4	0.3298	0.3793	0.4164
0.5	0.3018	0.3438	0.3632
0.6	0.2736	0.3187	0.3374
0.7	0.2592	0.2963	0.3021
0.8	0.24i4	0.2578	0.2732
0.9	0.2193	0.2372	0.2614
1.0	0.1816	0.2027	0.2343
Performance Improvement = 18.4%			

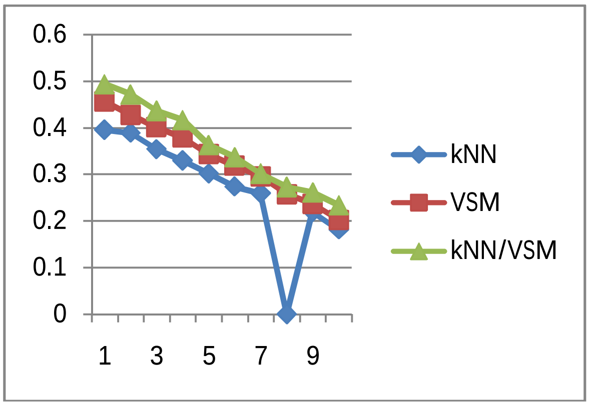

Figure 5: A graph of CRN4NUL collection with 1398 documents and 155 queries

Table 5: Results for MEDNUL collection with 400 documents and 30 queries

MEDNUL			
		Precision	
Recall	kNN	VSM	kNN/VSM
0.1	0.4234	0.4989	0.5472
0.2	0.4086	0.4648	0.5149
0.3	0.3828	0.4421	0.4827
0.4	0.3483	0.4267	0.4562
0.5	0.3352	0.4012	0.4211
0.6	0.3136	0.3746	0.4052
0.7	0.2943	0.3354	0.3654
0.8	0.2638	0.3173	0.3471
0.9	0.2463	0.3082	0.3297
1.0	0.2183	0.2602	0.2169
Performance Improvement = 15.7%			

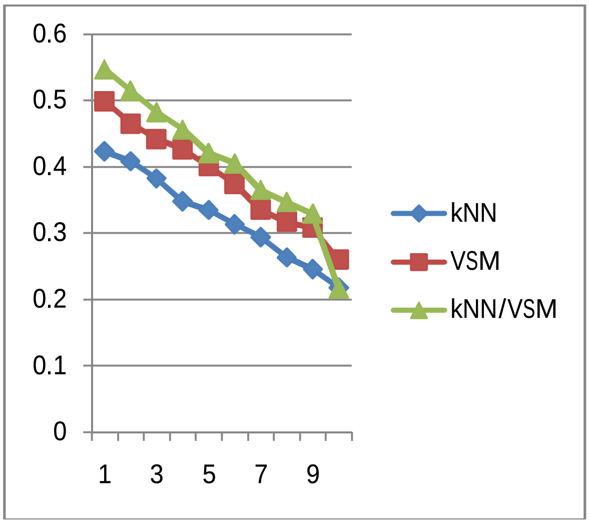

Figure 6: A graph of MEDNUL collection with 1398 documents and 30 queries

Table 6: Results for MEDLARS collection with 1,083 documents and 30 queries

MEDLARS			
		Precision	
Recall	**kNN**	**VSM**	**kNN/VSM**
0.1	0.4361	0.4831	0.5732
0.2	0.4180	0.4698	0.5522
0.3	0.3912	0.4532	0.5021
0.4	0.3623	0.4352	0.4833
0.5	0.3438	0.4127	0.4632
0.6	0.3202	0.3972	0.4383
0.7	0.3012	0.3634	0.4128
0.8	0.2873	0.3432	0.3846
0.9	0.2632	0.3326	0.3635
1.0	0.2287	0.2214	0.3233
Performance Improvement = 23.8%			

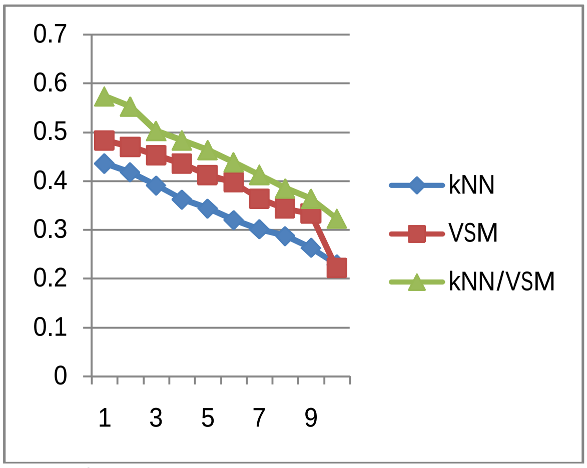

Figure 7: A graph of MEDNUL collection with 1398 documents and 30 queries

V. CONCLUSION

When we compared this hybridized technique with KNN and VSM separately used, it was found that the performance of this approach is better than when the tools are used separately. That is, our technique is able to retrieve information faster with significant lesser time. Using the recall values of 0.1, 0.2, …, 1.0, the precision for KNN and VSM for each recall value is computed. Having obtained these values, the corresponding KNN/SVM hybridized values are then computed for ADINUL, CRANFIELD, CRN4NUL, MEDNUL, MEDLARS respectively. The results obtained are shown in Tables 2, 3, 4, 5, 6 with their corresponding graphs in figures 3, 4, 5, 6, 7 respectively. The performance improvement in each case is also computed. Based on these performances, it was discovered that the combined KNN/VSM retrieval models outperforms that of KNN or VSM when used separately. The performance improvement for each database collection was also computed. ADINUL is 36.4%, CRANFIELD is 47.8%, CRN4NUL is 18.4%, MEDNUL is 15.7%, and MEDLARS is 23.8%. Thus we conclude that the hybridized KNN/VSM model is better in ranking and retrieving relevant documents than the previous techniques.

REFERENCES

1. Baeza-Yates, R., and Ribeiro-Neto, B. Modern Information Retrieval, ACM Press, 1999.
2. Beitzel, S. M. Jensen, E. C. Chowdhury, A. and Frieder, O. 'Varying Approaches to Topical Web Query Classification'. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 783–784, New York, NY, USA, 2007.
3. **A**rasu, A., Cho, J., Garcia-Molina, H., Paepcke, A. and Raghavan, S. 'Searching the Web**,** ACM Transaction on Internet Technology', Vol. 1, No. 1, pp. 2 – 43, 2001.
4. Fraternali, P. 'Tools and Approaches for Developing Data Intensive Web Applications: A Survey', ACM Computing Survey, Vol. 31, No. 3, pp. 227 – 263, 1999.
5. Gandal, N. 'The Dynamic of Competition in the Internet Search Engine Market', International Journal Industrial Organization, Vol. 19, No. 7, pp. 1103 – 1117, 2001.
6. Laerty, J. and Zhai, C. 'Document Language Models, Query Models, and Risk Minimization for Information Retrieval'. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval , pp. 111–119, New York, NY, USA, 2001. ACM.
7. Lee, U., Liu, Z. and Cho, J. 'Automatic Identification of User Goals in Web Search'. In Proceedings of the 14th International Conference on World Wide Web (WWW'05), pp. 391–400, New York, NY, USA, 2005.
8. Liu, T. Y., Xu, J., Qin, T., Xiong, W. and Li, H. 'LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval'. In Proceedings of the Learning to Rank workshop in the 30th annual International ACM SIGIR Conference (SIGIR'07) on Research and Development in Information Retrieval, 2007.
9. Liu, T.-Y. Yang, H. Wan, Zeng, H.-J., Chen, Z. and Ma, W.-Y. 'Support Vector Machines Classification with a Very Large-Scale Taxonomy', SIGKDD Explor. Newsl. , Vol. 7, No. , pp. 36–43, 2005.
10. Nallapati, R. 'Discriminative Models for Information Retrieval.' In Proceedings of the 27th Annual International ACM SIGIR conference (SIGIR'04) on Research and Development in Information retrieval, pp. 64–71, New York, NY, USA, 2004. ACM.
11. Lv, Y. and Zhai, C. 'Adaptive Relevance Feedback in Information Retrieval.' In Proceedings of CIK'09, November 2 – 6, Hong Kong, China, 2009.

12. Ponte, J. M. and Croft, W. B. 'A Language Modeling Approach to Information Retrieval.' In Research and Development on Information Retrieval, pp. 275–281, 1998.
13. Xing, E. , Ng, A., Jordan, M. and Russell, S. 'Distance Metric Learning, with Application to Clustering with Side-Information.' In Advances in NIPS, Vol. 15, 2003.
14. Wong, S. K. M., Ziarko, W., Raghaven, V. V., Wong, P. C. W. 'On Modeling of Information Retrieval Concepts in Vector Spaces', ACM Transaction on Database Systems, vol. 12, No. 2, pp. 299-321, 1987.
15. Tsatsaronis, G. and Panagiotopoulou, V. 'A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness'. In Proceedings of the EACL 2009 Student Research Workshop, Athens, Greece, pp. 70-78, 2009.
16. Geng, X., Liu, T. Y., Qin, T., Arnold, A., Li, H. and Shum, H.-Y. 'Query Dependent Ranking Using K-Nearest Neighbor', SIGIR'08, July 20-24, 2008, Singapore.
17. Kang, J. and Kim, G. 'Query Type Classification for Web Document Retrieval'. In Proceedings of the 27th Annual Int'l ACM SIGIR Conference on Research and Development in Information Retrieval, 2003.
18. Guru, D. S. and Nagendraswamy, H. S. 'Clustering of Interval-Valued Symbolic Patterns Based on Mutual Similarity Value and the Concept of Mutual Nearest Neighbour'. In ACCV(2), pp. 234-243, 2006.