DOI: 10.21917/ijsc.2021.0340

CLASSIFICATION OF QUERY RECOMMENDATION USING QUERY SEMANTIC FLOW GRAPH TECHNIQUE ON NOVEL AOL LOG METHOD

R. Balakrishnan

Department of Computer Science, Rathnavel Subramaniam College of Arts and Science, India

Abstract

Web mining is one among the thrust area of research in the data mining domain. The classification of query recommendation can be divided into two major classes that are document-based approach and log-based approach. Log-based method can get relatively good query recommendation and find query inner relation. Query flow graph is one of log-based method and get relatively good recommendation. However, query flow graph cannot get query semantic information. And there are many isolated nodes because of data sparseness. Therefore, word2vec is used to define the query semantic and add query semantic to query flow graph. That can be able to modify query transfer probability which is calculated by query flow graph. At the same time, we can get connection between the isolated queries that are related but no connection due to the data sparseness and the inaccuracy of session split. Empirical tests are conducted in accordance with the AOL log. From the results the efficiency of the approach in suggesting queries and F1 value is about 20% higher than the traditional query flow graph.

Keywords:

Query Recommendation, Word2vec, Semantics, Query Flow

1. INTRODUCTION

With the increasing amount of web content, it is increasingly hard to obtain helpful knowledge that can meet the need of the user on the basis of the actual search query [1]. So the users reconstruct a new query, which is identical to the actual search query and is nearly identical to the search goals of the user. For instance, when the users enter a new query "apple" to website, proper information is not retrieved for them. So the search engine will yield a set of new queries such as "apple website", "iPhone". By this means, the users can select a new query for searching relevant info and retrieve the message that they intend to get as quick as possible.

In the 1990s era, the query recommendation concept is first proposed to help user getting the next query that is nearly the same as the initial query. Afterwards, query recommendation technology has attracted the attention of a large number of research scholars. At present, the technology can be split into two groups which are document-based approach [2] [3] and log-based approach [4]-[6]. Document-based method finds the related queries or phrases through related documents containing the query and the existing dictionaries. However, how to construct the discovered words to a query is a major difficulty.

So at present, query recommendation technology generally uses log-based method. When the user searches for information on the search engine, the search engine records the search activity of the user and forms search logs. The query logs of search engine include query content, query time, click URL, and the URL location in the search page.

The log-based method uses the kinds of information in the query log to find out the relationship between the queries and recommend similar queries. Query flow graph [7] is one of query recommendation method. This method is based on log information to get transfer probability between the queries and suggests queries that are approximately related to initial queries. But we cannot find semantic information of query from search log. And query logs are sparse. There many isolated points in the query flow. Concurrently, when we build the query flow graph, we think that the queries have the same search goal in a session. The queries in a search goal have similarity. But sometimes the session partition is inaccurate so that the transfer probabilities of queries cannot calculate accurately.

So, in this paper, we add query semantic information into query flow graph to modify the transfer probability between queries. We use word2vec to represent queries which are nodes in query flow graph. We recalculate the query transfer probability which combines semantic information.

2. RELATED WORK

2.1 QUERY RECOMMENDATION

Antonellis et al. [8] added the idea of weight to traditional Simrank and applied it in advertising recommendation. Beeferman et al. [9] constructed query-URL bipartite graph from the historical log files in the search engine and used agglomerative clustering algorithm for the query clustering and URL to get related queries.

Ma et al. [10] used a union matrix that unifies query-URL bipartite graph and user-query bipartite graph for learning the low dimensional latent feature vectors belonging to the query and a solution was presented to calculate query similarity utilizing those feature vectors. Gupta et al. [11] used selectivity estimation to optimize query results. Zahera et al. [12] introduced a technique that depends on clustering procedures in which sets of semantically identical queries are found.

Boldi et al. [7] proposed Query flow graph and it considered query sequence. When queries successively appear in one session, the number of queries increases. Thus, rather than counting query number, counting query semantic might improve query recommendation by applying semantics to recommendation.

2.2 WORD VECTOR

With the development of deep learning, the method of word vector has garnered immense focus in Natural Language Processing. The simplest word vector is 0ne-hot representation. The main idea of this method is using word vector which has a dictionary size length to express every word. The ith position is non-zero, and other positions are zero. So it is easy to cause

dimensional disaster and Semantic gap. In 1986, Hinton et al. [13] proposed Distributed Representation which can solve the above problems. This method maps words into word vectors of fixed size, treats each word vector as a point in space, and calculates the spatial distance of each word vector. Many researchers express information by vector. Rygl et al. [14] presented a new scheme for 'vector similarity searching' over words and documents that are densely represented semantically.

Chowdhury et al. [15] proposed an approach to compute centroid vector for passages according to the wording and intention of the given query. Word2vec is a relevant model used to generate word vectors. It can define a word into vector form rapidly and efficiently through the optimized training model according to the given corpus. Li et al. [16] used a new hybrid framework known as mixed word embedding to capture the syntax information of words more accurately based on the word2vec toolbox. Singh et al. [17] use word2vec approach to choose the terms that are semantically identical with query once Borda count rank combining scheme is applied.

3. PROPOSED WORK

3.1 SEMANTIC REPRESENTATION OF QUERIES

The query flow graph mines the order of queries in query logs. Users input a query to the search engine. Then they submit a new query in a session. So we think that two successively submitted queries are relevant. In query flow, there is an edge between those two queries. If it is not rightly differentiated if the query belongs to the same search intent, we cannot accurately calculate the query transfer probability. At the same time, because of the sparseness of logs, there is no connection between many queries. So we can use query semantic information to modify query transfer probability. In this paper, we use vector to represent query semantic information. Google has opened word2vec for training word vectors under the study of the statistical language models. The learning procedure of a vector using word2vec explicitly encodes many language rules and patterns. Many of these modes can be formulated as linear transformations [18] [19]. For instance, the results can be found by just computing vector ("King")-vector("Man")+vector("Woman") is quite near to the vector of "Queen". So, considering the element-wise summation or mean of the word that embeds over every word in the sentence also generates a vector capable of encoding the meaning. In the event of the availability of the vector form of words in phrases and sentences, the techniques of vector sum or mean constitute the inexpensive models which are used to get the vector related to phrase and sentence. The queries in the search log are generally brief, spanning just two or three words.

So the semantic information of the query can be got by the linear combination of the word vectors. At the same time, the word vector of every word in the query is simply acquired using the corpus training. Therefore it is a time-conserving approach for and the most time saving method is to compute the word vector of the query by summing.

The word vector representation of queried can be divided into three steps:

In the first step, every query can be considered to be a group of words, indicated as $q = \{q_{w1}, q_{w2}, \cdots q_{wn}\}$, where q stands for the query, where q_{wi} refers to the word in the query.

Secondly, we use the word2vec model [20], [21] for training the word vectors of each word in query. Each word can be calculated as

$$V_q = \sum_{i=1}^{n} word2vec(q_{wi}) \quad (1)$$

where $word2vec(q_{wi})$ refers to the word i present in the query and n is the number of words present in the query.

Thirdly, the cosine similarity between each query vector is calculated, and the semantic information between queries is obtained.

We can compute the semantic transfer probability by the following formulas:

$$sim_{sem}(q_i, q_j) = sim(V_{qi}, V_{qj}) = \frac{\sum_{i=1}^{n}(x_i \times x_j)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(x_j)^2}} \quad (2)$$

where x_i, x_j represents the value of the word vector V_{qi}, V_{qj}.

3.2 TRANSFER PROBABILITY CALCULATION

The transfer probability between queries includes two parts. One is obtained through query flow graph; the other is calculated by own semantic information.

In query flow graph, the queries are correlated when their search intent is the same. When two queries in a session and q_j is input instantly after q_i, an edge exists from q_i to q_j in the query flow graph. We can define query flow graph as

$$G = (Q, E_{QQ}, w)$$

where,

$Q = \{q_1, q_2, \cdots, q_n\}$ is the set of distinct queries \$Q\$ submitted to the search engine.

$E_{QQ} = \{e\}$ refers to the set of edges which link queries that have been submitted sequentially in a query session.

$w \in (0,1)$ is weight that measures the probability of transition from query q_i to query q_j.

The transfer probability between queries takes into account the order of successively input between queries, depicting the query intent and the semantic relevance between the queries. The query transfer probability is computed as below: first, the search log in the search engine is split into session. Then, if q_i and q_j in the same session and q_j is input instantly after the query q_i, so there is an edge in the query flow graph that points from q_i to q_j. Finally, we can compute the similarity between the query and the query and construct query flow graph.

The query similarity is calculated as

$$sim_{QFG}(q_i, q_j) = \begin{cases} \dfrac{f(q_i, q_j)}{f(q_i)} & if\ \left(w(q_i, q_j)\right) > 0 \\ 0 & otherwise \end{cases} \quad (3)$$

where $sim_{QFG}(q_i, q_j)$ denotes the similarity between query q_i and query q_j. $f(x_i)$ denotes the number of times which q_i appears in the search log. $f(q_i, q_j)$ denotes the number of times that q_i is submitted after q_j in the search log.

In this article, the construction of a query flow graph model depends on the query session, and the 30min indicates the threshold for the session division of the search log. The query in the search log is indicated as each node, and the frequency of the query that are sequentially issued in the session is computed.

The semantic transfer probability is calculated in sec3.1. We apply the semantic representation of queries to the query flow graph to recalculate the transfer probability between queries, and get a new transfer probability. The new transfer probability can be defines as following:

$$sim_{query}(q_i, q_j) = \alpha sim_{QFG}(q_i, q_j) + \beta sim_{sem}(q_i, q_j) \quad (4)$$

As we can see, the parameters are weight which can balance the query information and its semantic information.

2.1 QUERY RECOMMENDATION ALGORITHM

In this work, query recommendation algorithm that depends on query semantic is provided as Algorithm 1. We calculate the transfer probability by using query information and semantic information. After the users submit the query in search engine, restart random walk is used [22] to suggest the query approximately equivalent to the input query. Random walk with restart is given by Eq.(5).

$$\vec{r_i} = cW\vec{r_i} + (1-c)\vec{e_i} \quad (5)$$

where c indicates the restart probability and W indicates the transfer probability matrix. $\vec{e}_i$ refers to the initial vector, The i^{th} element is 1, the remaining is 0. $\vec{r}_i$ indicates score vector.

During the recommendation process, the initial query is a point of start, and it randomly chooses the neighboring query with the initial query, and shifts to the neighboring query. Next, the present adjacent query is taken to be the initial queries and the above procedure of random walk is repeated. At last, the top queries is found to make recommendations to users that are identical to the initial query.

Algorithm 1: Query recommendation algorithm based on query semantic

Input: Query q

Output: Top N recommended queries

Step 1: Vector representation of queries

$$V_q = \sum_{i=1}^{n} word2vec(q_{wi}) \quad (6)$$

Step 2: Define $sim_{sem}(q_i, q_j)$ as

$$sim(V_{qi}, V_{qj}) = \frac{\sum_{i=1}^{n}(x_i \times x_j)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(x_j)^2}} \quad (7)$$

Step 3: Compute the hybrid query transfer probability $sim_{query}(q_i, q_j)$ as

$$sim_{query}(q_i, q_j) = \alpha sim_{QFG}(q_i, q_j) + \beta sim_{sem}(q_i, q_j) \quad (8)$$

Step 4: Do $\vec{r_i} = cW\vec{r_i} + (1-c)\vec{e_i}$

Step 5: Output first Top-5 results from the ranking vector $\vec{r}_i$

4. EXPERIMENT AND RESULT ANALYSIS

4.1 EXPERIMENTAL DATA AND EVALUATION APPROACHES

The data set is utilized in this work which is taken from search logs from AOL search engine from March to May in 2006. Experimental data contain 3558184 records. We extract 80% records in the form of training set and 20% in the form of test set.

The reprocessing of the training set is divided into three steps: First of all, threshold of 30min is used for the session split in order to make an estimation on the probability of two queries having the search target to be the same. Then, the query with www and other navigation vocabulary are eliminated, reducing noise. At last, we get rid of the edges which connect the queries below than five.

In the process of testing, we find the entire queries which submit behind query q in test set and are in a session with q to create a query set with relevance.

In case the recommended query is present in the relevant query set, then the recommended query is regarded as achieved success. In this paper, we choose the first N queries for assessing the precision, recall and F1 measure. The precision, recall and F1 metrics are formulated as below:

$$\text{Precision} = \frac{\text{Number of correct queries}}{\text{Number of total queries}} \quad (9)$$

$$\text{Recall} = \frac{\text{Number of correct queries}}{\text{Number of total correct queries}} \quad (10)$$

$$\text{F1} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision}+\text{Recall}} \quad (11)$$

4.2 RESULTS OF EXPERIMENTS AND ANALYSIS

4.2.1 Impact of Parameters (α,β):

At first, we conduct multi-group experiments for the values of two parameters in the calculation of transfer probability. In the experiments, the parameters α, β are satisfied at more than 0 and less than 1. At the same time, the range of each parameter are changed to 0.1 and satisfy $\alpha, \beta = 1$. We notice the impact of the precision, the recall and the F1 measure at Top10. The experimental results show in table 1.

Table.1. Impact of Parameters

Parameters	**Precision**	**Recall**	**F1**
α=0.1, β=0.9	0.4	0.143428	0.144081
α=0.2, β=0.8	0.4014706	0.143661	0.144494
α=0.3, β=0.7	0.401471	0.144067	0.14501
α=0.4, β=0.6	0.401471	0.144089	0.145053
α=0.5, β=0.5	0.4044	0.14421	0.14528
α=0.6, β=0.4	0.404412	**0.144286**	**0.145409**

α=0.7, β=0.3	0.404411	0.144181	0.145249
α=08, β=02	0.402941	0.143872	0.144726
α=0.9, β=0.1	**0.405882**	0.144248	0.145397

As show in Table.1, when α is 0.6 and β is 0.4, recall and F1 measure are highest. And α is 0.9 and β is 0.1, we can get the highest precision. Considering the above situation, we take parameters α is 0.6 and β is 0.4. In the later experiment, we use those values of parameters to calculate the transfer probability.

4.2.2 Evaluation of Efficiency:

One of our main goals of this paper is to show that query semantic information has the positive effects on the quality of query recommendation. We compare our method with the traditional query flow graph and only using query semantics to query recommendation.

- ***QFG***: a traditional method for query recommendation, which calculate the number of queries in a session.
- ***SemQuery***: a method only use query semantic information.
- ***SemQFG***: We add semantic information to query flow graph.

The precision measured is illustrated in Fig.1.

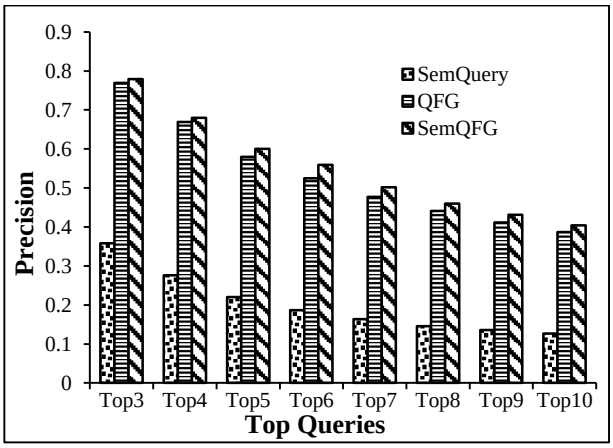

Fig.1. Precision of Adding Query Semantic

We can see that only using semantic information have low precision. User submit a query to search engine, we recommend new query which is not only have semantic similar but also have relevance. For example, when we search "travel", we recommend query such as "a long trip" which have semantic similarity with initial queries. But we also need recommend query such as "shanghai" which is relevant to initial queries. However, those queried does not have semantic similarity with initial queries. So we add query semantic to query flow. Query flow graph can mine query relation by query log. Query semantic can be used as supplementary to modify query transfer probability. We can more accurately calculated query transfer probability.

In Fig.2, we can find that our method can get better recall than traditional query flow graph. The F1 measure is shown in Fig.3. They all have the same trend as that observed in Fig.1. In Fig.3, F1 measures are 0.14321 in top3, 0.14019 in top5, and 0.12514 in top10 which are obtained by QFG. We can get the average of F1 measures is 0.13618. It is worth noting that the F1 measures are 0.17791, 0.16557, and 0.14541 respectively in Top1, Top2, and Top3 in SemQFG. We get the average is 0.16296. So, we found from the experimental results that the F1 value increased by 20% if we use query semantic information.

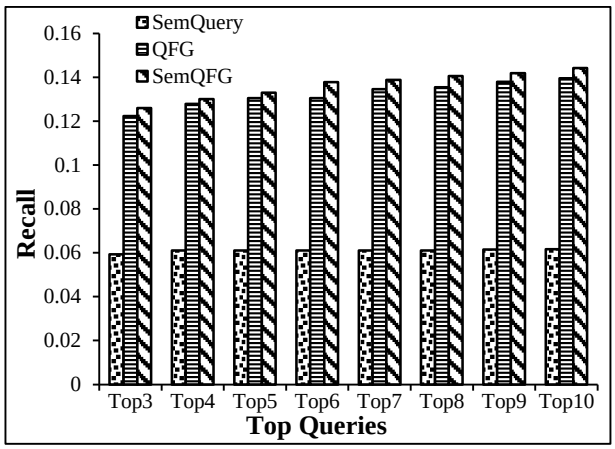

Fig.2. Recall of Adding Query Semantic

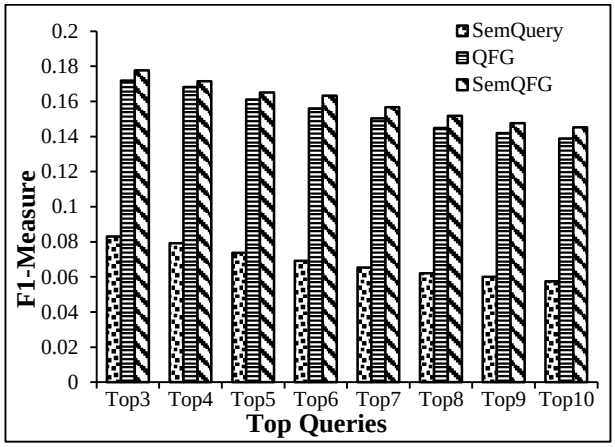

Fig.3. F1 of Adding Query Semantic

To evaluate the efficiency of the technique, the performance comparison of the following methods: (1) QUBG [8] uses query information and URL information in logs to construct Query-URL bipartite graph and recommend related queries. (2) CQM [12] use query clustering method for query recommendation. The results of the experiment are illustrated in Fig.4 - Fig.6.

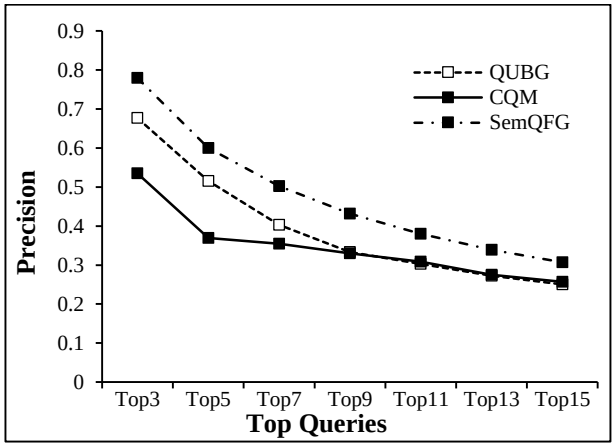

Fig.4. Evaluation of Precision

With an increase in recommended number, Fig.4 shows the change curve of the precision. It reveals that adding the semantic information to the query flow graph has much better precision compared to the other two approaches. Query semantic can modify query transfer probability which calculate by query flow graph. Also query semantic transfer probability can obtain transfer probability between queries which is isolated nodes in query flow graph.

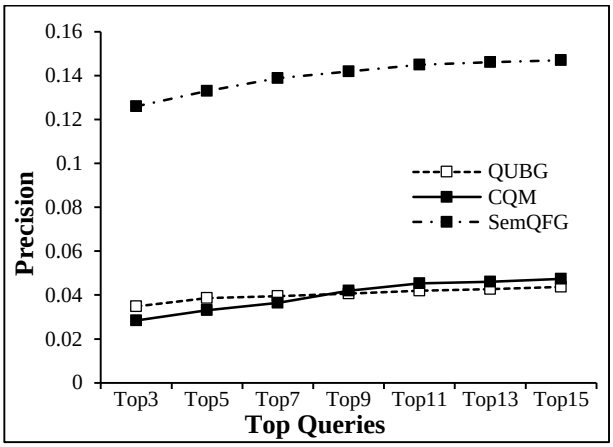

Fig.5. Evaluation of Recall

The Fig.5 illustrates the change curve of the three approaches.

We compare the recall with other two approaches. It can be seen that with the increase in the number of recommendations, the recall rate of this technique and other approaches rise. But this approach always achieves a better recall compared to the other two techniques.

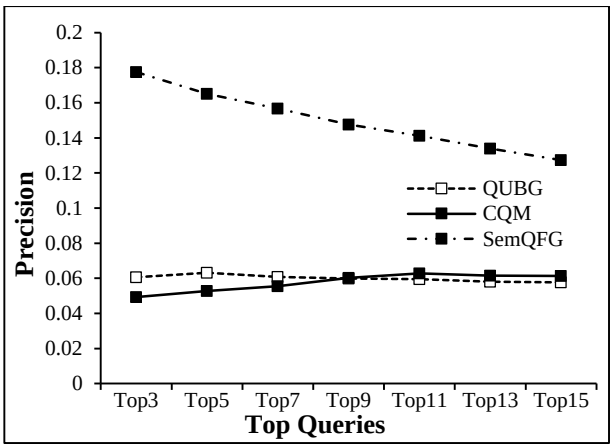

Fig.6. Evaluation of F1

As seen from Fig.6, it exhibits the same trend as that found in Fig.5. All the results can prove that adding query semantics can help improving the results of query recommendation in query flow graph.

5. SUMMARY

In this work, we used word2vec to represent each query in the query flow graph. Also, we recalculated the transfer probability between queries when we added the semantic information to query flow graph. Experiments based on AOL log showed that this technique had improved performance compared to the conventional query flow graph, and the precision, recall rate and F1 measure had been significantly improved. For the work intended for the future, consider other information present in the search log for improving the recommended result that can be very close to the query intent.

REFERENCES

[1] R. Kop, "The Unexpected Connection: Serendipity and Human Mediation in Networked Learning", *Journal of Educational and Technology Society*, Vol. 15, No. 2, pp. 2-11, 2012.

[2] R.W. White and G. Marchionini, "Examining the Effectiveness of Real-Time Query Expansion", *Information Processing and Management*, Vol. 43, No. 3, pp. 685-704, 2007.

[3] S. Noor and S. Bashir, "Evaluating Bias in Retrieval Systems for Recall Oriented Documents Retrieval", *International Arab Journal of Information Technology*, Vol. 12, No. 1, pp. 53-59, 2015.

[4] N.J. Belkin, C. Cool, J. Head, J. Jeng, D. Kelly, S. Lin and L. Lobash, "Relevance Feedback versus Local Context Analysis as Term Suggestion Devices: Rutgers TREC-8 Interactive Track Experience", *Proceedings of International Conference on Text Retrieval*, pp. 565-574, 2000.

[5] Z. Cheng, B. Gao and T.Y. Liu, "Actively Predicting Diverse Search Intent from User Browsing Behaviors", *Proceedings of International Conference on World Wide Web*, pp. 221-230, 2010.

[6] B. Zhang, B. Zhang, S. Zhang and C. Ma, "Query Recommendation based on Irrelevant Feedback Analysis", *Proceedings of International Conference on Biomedical Engineering and Informatics*, pp. 644-648, 2016.

[7] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis and S. Vigna, "The Query-Flow Graph: Model and Applications", *Proceedings of ACM Conference on Information and Knowledge Management*, pp. 609-618, 2008.

[8] I. Antonellis, H. GarciabMolina and C.C. Chang, "Simrank++: Query Rewriting Through Link Analysis of the Click Graph", *Proceedings of International Conference on World Wide Web*, pp. 408-421, 2007.

[9] D. Beeferman and A. Berger, "Agglomerative Clustering of a Search Engine Query Log", *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 407-416, 2000.

[10] H. Ma, H. Yang, I. King and M.R. Lyu, "Learning Latent Semantic Relations from Click through Data for Query Suggestion", *Proceedings of ACM Conference on Information and Knowledge Management*, pp. 709-718, 2008.

[11] S. Gupta and D. Garg, "Selectivity Estimation of Range Queries in Data Streams using Micro-Clustering", *International Arab Journal of Information Technology*, Vol. 13, No. 4, pp. 396-402, 2016.

[12] H.M. Zahera, G.F. El Hady and W.F. Abd El Wahed, "Query Recommendation for Improving Search Engine Results", *Lecture Notes in Engineering Computer Science*, Vol. 2186, No. 1, pp. 45-52, 2010.

[13] G.E. Hinton, "Learning Distributed Representations of Concepts", *Proceedings of 8th International Conference of the Cognitive Science Society*, pp. 1-12, 1986.

[14] J. Rygl and P. Sojka, "Semantic Vector Encoding and Similarity Search using Fulltext Search Engines", *Proceedings of Workshop on Representation Learning for NLP*, pp. 81-90, 2017.

[15] M.F.M. Chowdhury, V. Chenthamarakshan, R. Chakravarti and A.M. Gliozzo, "Query Focused Variable Centroid Vectors for Passage Re-ranking in Semantic Search", *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pp. 1-14, 2018.

[16] J. Li, J. Li, X. Fu, M.A. Masud and J.Z. Huang, "Learning Distributed Word Representation with Multi-Contextual Mixed Embedding", *Knowledge-Based Systems*, Vol. 106, No. 3, pp. 220-230, 2016.

[17] J. Singh and A. Sharan, "Relevance Feedback-based Query Expansion Model using Ranks Combining and Word2Vec Approach", *IETE Journal of Research*, Vol. 62, No. 5, pp. 591-604, 2016.

[18] L. White, R. Togneri, W. Liu and M. Bennamoun, "How Well Sentence Embeddings Capture Meaning", *Proceedings of 20th Australasian Symposium on Document Computing*, pp. 1-8, 2015.

[19] X. Rong, "Word2vec Parameter Learning Explained", *Proceedings of International Conference on Computer Science*, pp. 1-21, 2014.

[20] Y. Li and K. Lyons, "Word Representation using a Deep Neural Network", *Proceedings of International Conference on Computer Science and Software Engineering*, pp. 268-279, 2016.

[21] W. Ling, C. Dyer, A.W. Black and I. Trancoso, "Two/Too Simple Adaptations of Word2Vec for Syntax Problems", Proceedings of Conference on North American Chapter of the Association for Computational Linguistics - Human Language Technologies, pp. 1299-1304, 2015.

[22] H. Tong, C. Faloutsos and J.Y. Pan, "Fast Random Walk with Restart and its Applications", *Proceedings of International Conference on Data Mining*, pp. 613-622, 2006.