



Research Journal of  
**Information  
Technology**

ISSN 1815-7432



Academic  
Journals Inc.

[www.academicjournals.com](http://www.academicjournals.com)

## **A New Term-ranking Approach that Supports Improved Searching in Literature Digital Libraries**

<sup>1</sup>Suliaman Bani-Ahmad and <sup>2</sup>Ghadeer Al-Dweik

<sup>1</sup>Al-Balqa Applied University, Salt Campus, Jordan

<sup>2</sup>Ministry of Education, Jordan

*Corresponding Author: Suliaman Bani-Ahmad, Al-Balqa Applied University, Salt Campus, Jordan*

### **ABSTRACT**

In example-based searching, users look for the set of most similar publications to a given one. This requires estimating similarities between publications. A tf.idf formula can be used to compute publication-to-publication text-based similarity, e.g., the Okapi BM25 formula. Studies show that augmenting the importance of search terms in the BM25 formulae improve similarity scores. To this end, we introduce a term-ranking technique and use it for improving publication similarity scores. The proposed term-ranking algorithm is a slight modification of the TextRank algorithm that utilizes the well-known PageRank algorithm to identify the important term/phrases within texts. The proposed approach considers the length of sentences to identify links between terms rather than considering fixed window size. We experimentally found that the proposed approach works well when paired with Okapi BM25.

**Key words:** TextRank, punctuation, term-ranking, example-based searching, digital libraries

### **INTRODUCTION**

Example-based searching, where user provides an example publication to locate the most similar set of publications to it, is becoming commonplace in literature digital libraries (Bani-Ahmad *et al.*, 2005). The text-based approaches for computing similarity between publications consider observing shared terms between publications to be an indicator of similarity (Bani-Ahmad *et al.*, 2005; Troy and Zhang, 2007; Hawking and Thistlewaite, 1996).

Besides implementing example-based searching, publication-to-publication similarity computation can prove useful for digital libraries to classify publications into contexts as well (Bani-Ahmad and Ozsoyoglu, 2010). This task is important as it can significantly enhance sorting and grouping search output of digital collections (Bani-Ahmad *et al.*, 2005; Bani-Ahmad and Ozsoyoglu, 2010).

Studies show that accurately ranking terms can significantly enhance the precision of search results of digital libraries (Troy and Zhang, 2007) and thus making searching process more effective and efficient.

Studies show that augmenting the importance of terms may improve similarity scores in keyword-based search (Troy and Zhang, 2007). Term frequency was the first to be used as an indicator of term importance (Bennett *et al.*, 2008), the use of term frequencies in order to estimate term significance was introduced first in Luhn (1958). Since then, term-frequency-based methods have become the reference-point by which new research in document-relevance computation is

evaluated. Two more techniques for ranking terms can be found in literature. Namely; (i) Chronological Term Rank (CTR) proposed by Troy and Zhang (2007) and (ii) and the well-known Term Proximity (TP) (Troy and Zhang, 2007; Voorhees, 2005).

The CTR value of a term in a document is computed as the position where the term is first observed in the given document. In Term Proximity, the documents where search-terms are observed physically close to each other are considered to be more relevant to user query (Troy and Zhang, 2007; Voorhees, 2005). In Bani-Ahmad and Al-Dweik (2010), the authors utilize TextRank to compute the importance of terms and use it to improve the precision of example-based search in digital collections.

What motivated applying variable window sizes rather than fixed window size is that we have experimentally observed performance discrepancy when using TextRank with different sizes of windows. One possible cause for that discrepancy of TextRank performance is that the best window size to be used is probably different for different texts.

In TextRank algorithm, two terms are linked to each other if they are observed within a given window size even if they are from two different sentences or even paragraphs. This occurs to the terms that are located toward the end of one sentence and those observed at the beginning of the very next statement. Consequently, those terms will be linked in the final constructed graph. Thus, two terms that are from two sentences or even paragraphs can still recommend each other in TextRank. This raises an important question; why would a term give a recommendation to another term if the two are from two different sentences or even paragraphs that are probably topically different to some degree?

We believe that these cases of probably improper term-to-term recommendation relation produced the observed discrepancy of TextRank term-scoring approach for different sizes of windows.

The above observations have led us to consider the length of sentences as a way to identify edges between terms. This new term-ranking algorithm is slight modification of the TextRank algorithm. TextRank is a text summarizing tool that involves ranking terms (Mihalcea and Tarau, 2004). We refer to the new modified version of TextRank by Punctuation-Aware TextRank or PATeR for short. PATeR is similar to TextRank except that it considers variable size of window when forming text graphs. That is; the window size is determined by the length of sentences rather than considering fixed size windows. For that, PATeR takes into consideration the punctuation of the text to capture the beginning and end of statements.

In this study, we introduce enhancing example-based searching through term-ranking computed by PATeR. The PATeR algorithm is superior to CTR as it recursively computes term importance from the entire publication textual content where it is observed, rather than relying only on local specific information.

The major finding of the paper is that PATeR works well when paired with Okapi BM25. We evaluate the performance of PATeR and compare it against the performance of CTR and the TP approaches. Up to 9% improvement, in terms of precision, over CTR is observed.

## **THE OKAPI BM25 MODEL**

As mentioned before, in example-based searching in Literature Digital Libraries (LDL), the current user provides a publication P and is asking for a set of the top-K similar publications to P from the publications in the repository being searched S.

In keyword-based search, document D relevance to a given set of search keywords W is computed as the similarity level between W and D. In the context of example-based searching, W are the words appearing in the abstract of the example publication (Bani-Ahmad *et al.*, 2005). One well-known document relevance estimation measure is the Okapi BM25 (Troy and Zhang, 2007). The Okapi BM25 is a term-frequency/inverse-document-frequency (tf.idf) based group of relevance estimation measures (Schenkel *et al.*, 2007). This means that it uses (i) the number of times a term is observed in a document and (ii) the number of documents where that term is observed, in addition to (iii) a set of other statistical measures, to compute the document relevance based on some formula (Troy and Zhang, 2007). One famous and widely used group of formulae to compute document relevance is the Okapi BM25 (Robertson *et al.*, 1992).

One famous formula of Okapi BM25 model is (Troy and Zhang, 2007):

$$\text{Score}_{\text{BM25}} = \sum_{t \in d \cap q} \frac{\ln(N - df + 0.5)}{df + 0.5} \cdot \frac{tf}{0.5 + 1.5 \cdot dl / avdl + tf} \quad (1)$$

where, tf, term frequency, is the number of occurrences of a term in all documents. qtf, the number of occurrences of a term t in the keywords of the query q. df, the number of documents in the collection containing the term of interest. idf, document frequency is most commonly used in term weights as inverse document frequency. dl, document length. N, Number of documents in the collection. avdl, the average length of all documents. q is the set of query terms, i.e. the terms that we are searching for.  $d \cap q$  is the set of terms observed in both q and the document at hand d.

In the context of example-based searching q is the set of words observed in the example publication.

### OKAPI BM25 MODEL PAIRED WITH IMPORTANCE OF TERMS

Okapi BM25 is based on simple statistics. However, with the growing difficulty of achieving further retrieval improvements (higher precision and recall) using only term frequencies as in Eq. 1 above, there has been an increasing interest in information derived from document structure. Example of such information that can be derived is the relative importance of terms that appear in documents. Next we present multiple possible approaches to estimate term importance and augmenting it with Eq. 1.

**The chronological term rank:** In Troy and Zhang (2007), Chronological term rank (CTR), which captures the positions of terms as they occur in the sequence of words in a document, were used to enhance relevance score between search terms and the documents to be searched. The CTR model is defined as follows: let  $D = (t_1, \dots, t_n)$  be a document where  $t_i$  are terms (words) ordered according to their appearing-sequence in D. The term rank  $t_r$  of term t is the location i where t appears first in D.

CTR token-importance measure is augmented in Eq. 1 in multiple ways (Troy and Zhang, 2007), one of which is:

$$\text{Score}_{\text{BM25}_{\text{CTR}}} = \sum_{t \in d \cap q} \frac{\ln(N - df + 0.5)}{df + 0.5} \cdot \left( \frac{tf}{0.5 + 1.5 \cdot dl / avdl + tf} + R_{\text{CTR}} \right) \quad (2)$$

where,  $R_{\text{CTR}}$  is the CTR-based term importance score. The  $R_{\text{CTR}}$  value is added by the ratio before.

In the formula above,  $R_{CTR}$  is computed as follows:

$$R_{CTR} = C - \left( C.D. \frac{\log\left(\frac{tr-1}{20} + 10\right)}{\log\left(\frac{dl}{20} + 10\right)} \right) \quad (3)$$

where, C and D are constants and found to be  $C = 0.6$ ,  $D = 0.6$  for best results (Troy and Zhang, 2007).  $tr$  is the CTR-based term importance score and  $dl$  is the length of the document.

**TextRank:** In Mihalcea and Tarau (2004), TextRank is introduced and used for text summarization. TextRank algorithm involves mainly two steps as we stated before. In the first, importances of words, observed in text to be summarized, is computed. For that PageRank algorithm (Page and Brin, 1998) is used.

TextRank applies PageRank on a special graph built from text as follows. Vertices of this graph are the content-bearing words (i.e., all words observed in text excluding stopwords or noise words such as “the”, “an” and “who” (Wikipedia, n.d.)). A link is established between two words (vertices) if they are observed together within a pre-given window size,  $W$ . In Mihalcea and Tarau (2004), best choice of  $W$  is found to be 20 for text-summarizing purposes. PATeR is similar to TextRank except in that the window size is the length of statements, which may vary.

**Term proximity:** Term proximity refers to the lexical distance between search-query terms and is calculated as the number of words (including or excluding stop-words) separating query terms in a document (Vechtomova and Karamuftuoglu, 2008; Hawking and Thistlewaite, 1996; Troy and Zhang, 2007).

One way to augment term proximity information to Okapi BM25 is presented in Buttcher *et al.* (2006). In Schenkel *et al.* (2007), an efficient evaluation framework including a proximity scoring function integrated within a query engine for text retrieval is presented. The following equation is used in this study:

$$Score_{BM25_{TP}}(d, q) = Score_{BM25}(d, q) + \sum_{t \in q} \min\{1, idf(t)\} \cdot accd(t) \cdot \frac{k1+1}{accd(t)+K} \quad (4)$$

where,  $Score_{BM25}(d, q)$  is defined in Eq. 1 above and  $accd(t)$  is the accumulated interim score ( $acc$ ) for the query term  $t$  that depends on the distance of this term occurrences to other, adjacent query term occurrences.

The value  $K$  is computed as:

$$K = k \cdot \left[ (1-b) + b \cdot \frac{|d|}{avdl} \right]$$

where,  $b$ ,  $k1$  and  $k$  are configurable parameters that are set to  $b=0.5$  and  $k=k1=1.2$ , respectively (Schenkel *et al.*, 2007). And finally the  $avdl$  is the average document length.

Next in our experiments we propose and evaluate replacing the  $accd$  score of terms with the PATeR scores. The configurable parameters are assigned the same values used in Schenkel *et al.* (2007) and Buttcher *et al.* (2006).

## THE PATeR TERM RANKING APPROACH

TextRank applies the PageRank algorithm on semantic graphs extracted from natural language text. In such a graph terms represents vertices and links represents semantic relationships between those terms. Thus, the PageRank scores obtained, which are referred to as TextRank scores, are found to give a good approximation of the relative importance of the terms within the document where they are observed (Mihalcea and Tarau, 2004).

Formally, let  $G=(V,E)$  be a directed graph with the set of vertices  $V$  and set of edges  $E$ . For a given vertex  $v_i$ , let  $In(v_i)$  be the set of vertices that point to it and  $Out(v_i)$  be the set of vertices that vertex  $v_i$  points to. The score of  $v_i$  is recursively computed as follows (Page and Brin, 1998):

$$S(v_i) = (1 - d) + d \times \sum_{j \in In(v_i)} \frac{1}{|Out(v_j)|} S(v_j) \quad (5)$$

where,  $d$  is a damping factor that can be set between 0 and 1 (Page and Brin, 1998). In general,  $|out(v)|$  is the number of pages cited by page  $v$  and  $ln(v)$  is the set of pages citing page  $v$ .  $S(v)$  is the PageRank score of vertex  $v$ .

In the context of Web surfing, the PageRank algorithm implements the “random surfer model”, where a web-user is assumed to randomly click on links with a probability level of  $d$  and jumps to a completely new page with probability  $(1-d)$  (Page and Brin, 1998). It has been found that choosing  $d$  to be 0.85 gives accurate ranking results (Page and Brin, 1998).

Starting from arbitrary PageRank scores assigned to each node in the graph at hand, the computation iterates until convergence. A PageRank score after convergence of some vertex  $v$  represents the “importance” of  $v$  within the graph.

To enable the application of graph-based ranking algorithms to texts, we have to build a graph that represents the text and interconnects words can be added as vertices in the graph. The application of graph-based ranking algorithms to natural language texts consists of the following main steps: (1) identifying text units (words) and adding them as vertices in the graph, (i) identify relations that connect the text units identified in step 1 and use these relations to draw edges between vertices in the graph (Mihalcea and Tarau, 2004).

Any relation that can be defined between two text units is useful and can be added between their vertices (Mihalcea and Tarau, 2004). For our experiment and implementation of TextRank, we are using a co-occurrence relation, controlled by the distance between word occurrences as in (Mihalcea and Tarau, 2004; Bani-Ahmad *et al.*, 2005; Bani-Ahmad, 2009). Two vertices are linked if their corresponding text units co-occur within a window of maximum  $W$  words, where  $W$  is the window size and can be set anywhere from 2 to 10 words (or even more). In our experiment, we considered  $W$  to be covering the length of the sentence where terms are observed within the abstract of the publication at hand.

The TextRank keyword extraction algorithm proceeds as follows. (i) The text is tokenized, (ii) stop words are removed and (iii) tokens are stemmed using Porter stemmer (Salton, 1989; Kowalski, 1997) algorithm as in Mihalcea and Tarau (2004). (iv) All text units are added to the graph and an edge is added between them are added. After the graph is constructed, the score associated with each vertex is set to an initial value of 1 and PageRank is run on the graph for several iterations until PageRank scores converge, at a maximum threshold of 0.0001 as in Bani-Ahmad (2009).

## EXPERIMENTAL SETUP

In this section we compare the performance of using TextRank Graph-based term-rank against the use of Chronological Term Rank (Troy and Zhang, 2007) and the Term Proximity (Buttcher *et al.*, 2006; Schenkel *et al.*, 2007) approaches.

In order to show the effectiveness of ranking terms via the PATeR approach, information retrieval experiments were conducted based on a collection of 90 publications that are manually selected by domain experts of the research topics of those publications. To implement example-based search, the abstract of a publication from the collection is used to search for similar publications within the complete set. Top-K relevant documents (similar publications) are retrieved using an Okapi-based information retrieval experimental system developed for evaluation purposes. The Okapi BM25-based relevance scores of the result set are later modified by augmenting the TextRank scores of terms as described earlier in the paper. Next we present the list of performance metrics used to comparatively evaluate the proposed approach with the CTR and the TP approaches.

The Precision at top-K or (P@K) metric is used to measure the quality of a retrieval result: P@K is computed as the percentage of relevant publications (or similar publications to the example publication at hand) amongst the retrieved top scored K publications. We computed the precision considering multiple values for K; namely, K=5, 10 and 15.

A set of 90 abstracts of publications is used as a testbed. Those publications are divided into three groups each is of size 30. The three groups are carefully selected by domain experts from three different research-areas all in computer science. Each group is further subdivided into three subgroups of size 10 each. Each subgroup represents publications that are highly topically related. This means that, if you single-out one publication from one of the subgroups, the other 9 publications in that subgroup are good similar examples of the singled-out one.

For our experiments, we used Eq. 2 above to compute  $Score_{BM25_{CTR}}$  and Eq. 4 to compute the Term Proximity (TP) score ( $Score_{EM25_{CTR}}$ ). We augmented the PATeR score of terms with Okapi BM25 using the formula of Eq. 6 shown next.

$$Score_{BM25_{PATeR1}} = \sum_{t \in d \cap q} \frac{\ln(N - df + 0.5)}{df + 0.5} \left( \frac{tf}{0.5 + 1.5 \cdot dl / avdl + tf} + PATeR(t) \right) \quad (6)$$

The PATeR score of terms are also used with the formula of Eq. 4. The new formula is:

$$Score_{BM25_{PATeR2}}(d, q) = Score_{BM25}(d, q) + \sum_{t \in q} \min\{1, idf(t)\} \cdot PATeR(t) \cdot \frac{k1 + 1}{PATeR(t) + K} \quad (7)$$

## EXPERIMENTAL RESULTS AND EVALUATION

**The effect of window size on TextRank performance:** What motivated applying variable window sizes rather than fixed window size is that we experimentally observed discrepancy of TextRank term-scoring approach for different sizes of windows. Figure 1 shows the correlation between the size of the window applied and the performance of example-based search using the modified BM25 formula. The peak improvement was observed at window size of 1 and 3. However, we also observed that in rare cases other window sizes were superior to those two sizes. One possible cause for that discrepancy of TextRank performance is that the best window size to be used is probably different for different texts.

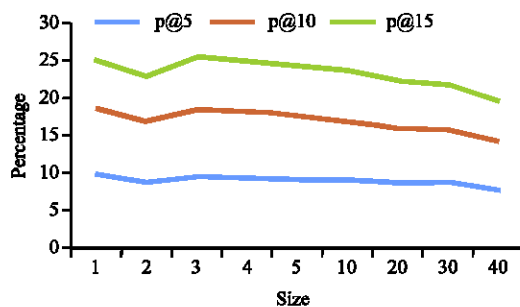


Fig. 1: Percentage precision enhancement TextRank over CTR (y axis) vs. the applied size of window size (x axis)

Table 1: Precision values (P@K) of the different approaches for different selections of K

Model	p@5	p@10	p@15
PATeR	0.658	0.533	0.427
CTR	0.607	0.504	0.413
TP	0.660	0.539	0.431

Table 2: Enhancements of PATeR over CTR

	p@5	p@10	p@15
PATeR	8.42%	5.73%	3.41%

Notice that the terms that are located toward the end of one sentence and those observed at the beginning of the very next statement are considered to be co-located by TextRank. Consequently, those terms will be linked in the final constructed graph. Thus, two terms that are from two sentences or even paragraphs can still recommend each other in TextRank. This raises an important question; why would a term give a recommendation to another term if the two are from two different sentences or even paragraphs that are probably topically different to some degree?

We believe that these cases of probably improper term-to-term recommendation relation produced the observed discrepancy of TextRank term-scoring approach for different sizes of windows.

The above observations have led us to consider the length of sentences as a way to identify edges between terms.

**BM25 with PATeR-equation 7:** In this section we compare the performance of PATeR (Eq. 7) against the Chronological Term Rank (Eq. 2) and the Term Proximity (Eq. 4).

**Observation (Table 1):** PATeR performed better than CTR in terms of top-K precision for all considered values of K.

Table 2 shows clearly that when augmenting the PATeR scores of terms with the BM25 formula, a maximum of 8.42% enhancement of precession values over that the CTR could achieve.

**Observation (Table 1):** Compared to (TP) results, (PATeR) showed similar or comparable improvement in precision to TP.

This observation generally applies to all considered levels of precision; namely, precision at top-5 (p@5), top-10 (p@10) and top-15 (p@15). But if we consider the time and space complexity required



Table 3: Precision when applying PATeR

Model	p@5	p@10	p@15
PATeR	0.651	0.499	0.415
CTR	0.607	0.504	0.413
TP	0.660	0.539	0.431

Table 4: Precision enhancement of PATeR over CTR

	p@5	p@10	p@15
PATeR	7.33%	-1.10%	0.54%

by the TP approach, the PATeR approach provides us with less time and space complex alternative to the TP approach at a close or slightly better level of precision than TP.

**BM25 with PATeR-equation 6:** In this section we compare the performance of PATeR (Eq. 6) against the Chronological Term Rank (Eq. 2) and the Term Proximity (Eq. 4).

**Observation (Table 3 and 4):** Compared to CTR, PATeR showed comparable and in some cases higher precision values, to those of CTR.

We also observed that PATeR has relatively higher top-K precision values for K=5. This saves the time of the current user as s/he finds what s/he is looking for on top of the search results list.

To summarize, we also observed that, in general, PATeR outperforms or is comparable to TP and outperforms CTR.

## CONCLUSION

In this research we introduced enhancing relevance scores with the Punctuation-Aware TextRank term ranking algorithm (PATeR). Just like TextRank. PATeR is different than TextRank in the sense that the edges between terms are added considering variable rather than fixed window sizes. The window size in PATeR is the length of the statement. Present experiments show that PATeR has significant improvement over CTR when augmented into the well-known BM25 formulae. PATeR has slight improvement over TP, but with less computational overhead.

## ACKNOWLEDGMENTS

We would like to thank the reviewers for their very valuable comments that have substantially improved the paper in both substance and in presentation.

## THE CASE EXPLORER PROJECT

The work conducted in this paper is part of the CASE EXPLORER project (2003-2008). The project is resumed by Sulieman Bani-Ahmad at Al-Balqa Applied University in Jordan. The CASE EXPLORER is a score-guided searching and querying prototype portal for ACM SIGMOD Anthology, a digital library for the database systems research community, containing about 15,000 papers.

As an extension of the CASE EXPLORER project, Bani-Ahmad resumed the project in Jordan and worked on enhancing example-based searching in literature digital libraries through term ranking. A part of this is the work is presented in this article. The other part can be found in Bani-Ahmad and Al-Dweik (2010).

## REFERENCES

- Bani-Ahmad, S. and G. Al-Dweik, 2010. On improved example-based search in digital libraries via term ranking. *Int. J. Theor. Applied Inform. Technol.*, 19: 44-54.
- Bani-Ahmad, S. and G. Ozsoyoglu, 2010. On using the research-pyramid model to enhance literature digital libraries. *Inform. Technol. J.*, 9: 1093-1103.
- Bani-Ahmad, S., 2009. On context-driven online search-phrase suggesters for large textual document repositories. *Proceeding of IADIS International Conference Information Systems 2009*, Feb. 25-27, Barcelona, Spain, pp: 1-3.
- Bani-Ahmad, S., A. Cakmak, A. Al-Hamdani and G. Ozsoyoglu, 2005. Evaluating score and publication similarity functions in digital libraries. *Comput. Sci.*, 3815: 483-485.
- Bennett, G., F. Scholer and A. Uitdenbogerd, 2008. A comparative study of probabilistic and language models for information retrieval. *ACM Int. Conf. Proc. Ser.*, 313: 65-74.
- Butcher, S., C.L.A. Clarke and B. Lushman, 2006. Term proximity scoring for ad-hoc retrieval on very large text collections. *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Aug. 06-11, ACM New York, NY, USA, pp: 621-622.
- Hawking, D. and P. Thistlewaite, 1996. Relevance weighting using distance between term occurrences. Technical Report TR-CS-96-08, The Australian National University, August 1996. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.7475&rep=rep1&type=pdf>.
- Kowalski, G., 1997. *Information Retrieval Systems Theory and Implementation*. 1st Edn., Kluwer Academic Publishers, Norwell, MA., USA., pp: 296.
- Luhn, H.P., 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2: 159-165.
- Mihalcea, R. and P. Tarau, 2004. TextRank: Bringing order into texts. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. <http://www.citeline.org/user/johnkork/article/430523>.
- Page, L., and S. Brin, 1998. The anatomy of large-scale hypertextual web search engine. *Proceeding of Computer Networks and ISDN System*, April 1, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, pp: 107-117.
- Robertson, S.E., S. Walker, M. Hancock-Beaulieu, A. Gull and M. Lau, 1992. Okapi at TREC. *Proceedings of the Text REtrieval Conference*, pp: 21-30. <http://www.citeline.org/user/aliku/article/131616>.
- Salton, G., 1989. *Automatic Text Processing*. Addison Wesley Longman Publishing Co. Inc., Boston, MA, USA., ISBN:0-2:1-1227-8, pp: 450.
- Schenkel, R., A. Broschart, S. Hwang, M. Theobald and G. Weikum, 2007. Efficient text proximity search. *Proceedings of the 14th International Conference on String Processing and Information Retrieval*, Santiago, Chile, Oct. 29-31, Springer-Verlag Berlin, Heidelberg, pp: 287-299.
- Troy, A.D. and G.Q. Zhang, 2007. Enhancing relevance scoring with chronological termrank. *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 23-27, ACM New York, USA., pp: 599-606.
- Vechtomoova, O. and M. Karamuftuoglu, 2008. Lexical cohesion and term proximity in document ranking. *Inform. Process. Manage. Int. J.*, 44: 1485-1502.
- Voorhees, E.M., 2005. Overview of TREC 2005. *Proceedings of the 14th Text Retrieval Conference (TREC 2005)* NIST Special Publication 500-266, National Institute of Standards and Technology. [http://trec.nist.gov/pubs/trec14/t14\\_proceedings.html](http://trec.nist.gov/pubs/trec14/t14_proceedings.html).