ON IMPROVED EXAMPLE-BASED SEARCH IN DIGITAL LIBRARIES VIA TERM RANKING

SULIEMAN BANI-AHMAD, GHADEER AL-DWEIK
Department of Information Technology. Al-Balqa Applied University, Salt campus. Jordan
Emails: sulieman@case.edu , ghadeerjad@hotmail.com

ABSTRACT

Example-based searching, where user provides an example publication to locate similar publications to, is becoming commonplace in literature digital libraries. Two approaches to estimate similarities between publications are (i) graph based approaches where citation relationships amongst publication are used to compute similarities, and (ii) text-based approaches where observing shared terms between publications is used as indicator of similarity. In this paper we introduce a new text-based publication-similarity measuring technique that enhances existing example-based searching through utilizing term importance information. Term importance is computed via a proposed graph-based term ranking (GBTR) algorithm. The GBTR algorithm is different from previous term ranking approaches as it recursively computes term importance from the entire publication where it is observed, rather than relying only on local specific information. GBTR works well when paired with Okapi BM25. We exhaustively evaluate the performance of GBTR and compare it against the performance of existing term-ranking methods such as the Chronological Term Rank (CTR) and the Term Proximity models. Significant improvements, in terms of precision, over existing approaches are observed. GBTR achieved around 10% improvement in precision over CTR and around 2% over TP with much less computational time and space complexity than the TP approach.

Keywords: *Okapi system, BM25, Text retrieval, Example-based search, TextRank, Term Proximity*.

1 INTRODUCTION

Searching literature digital libraries efficiently and effectively is becoming more and more important as the size and use of digital library collections expand at a very high rate. Examples are, (i) In Computer Science, ACM Digital Library [1] has close to 1 million full-text publications collected over 50 years, to search and download [6]; (ii) ScienceDirect [17], the world's leading scientific, technical and medical information resource celebrated its billionth article download in November'06 since launched in 1999 [6].

Example-based searching, where user provides an example publication to locate similar publications, is becoming commonplace in literature digital libraries [4]. Two approaches that can be used to estimate similarities between publications are (i) graph based approaches where citation relationships amongst publications are used to compute similarities [4], and (ii) text-based approaches where observing shared terms between publications is used as indicator of similarity [4; 18, 8].

Studies show that accurately ranking terms observed in the text to be searched can significantly enhance the accuracy and precision of search results of digital libraries [18], and thus making searching process more effective and efficient. In this paper we introduce a new text-based publication-similarity measuring technique that enhances existing example-based searching through utilizing *term importance information*.

Term frequency was the first to be used as an indicator of term importance [7], the use of term frequencies in order to estimate term significance was introduces in [11]. Since then, term frequency based methods have become the reference-point by which new research in document-relevance scoring is evaluated. Three more techniques for ranking tokens can be found in literature. Namely; (i) Chronological term rank (CTR) proposed in [18) (ii) TextRank score proposed in [12] to be used for text summarizing

and (iii) and the well-known Term Proximity [18; 20].

Chronological term rank (CTR) of a term **t** in a document **D** is computed as the position where the term **t** is first observed in **D**. In [18), it is used as an indicator of how important **t** is in **D**. In [18], this importance indicator is augmented in a well-known a tf.idf (term frequency / inverse document frequency) based relevance estimation technique called Okapi BM25.

TextRank is another way to compute the importance of a term in a particular document. TextRank [12] was first used as text summarizing tool and has been proven to be successful for that particular task [12]. TextRank algorithm takes mainly two steps; (i) the first involves computing importance scores of the set of word (token) observed in the text to be summarized. (ii) In the second step, the top-scored words are used to form phrases. Later, those phrases are used as summarization of the document at hand. In this proposal, we intend to imitate [18] and use the TextRank scores as term importance indicator and augment them in the Okapi BM25 formula. We compare the quality of search results based on using TextRank scores to the results based on CTR and the Term Proximity approaches.

In Term Proximity, documents where search terms are observed *physically close to each other* and *of the same order* to those provided by the user are considered to be more relevant to user interests (the topic of the example publication in example-based search) [18; 20].

In this paper we introduce a new text-based publication-similarity measuring technique that enhances existing example-based searching through utilizing *term importance information*. Term importance is computed via a proposed graph-based term ranking (GBTR) algorithm. The GBTR algorithm is different from previous term ranking approaches 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 main contributions of the papers is proposing and validating the usage of GBTR, or TextRank, scores of terms to improve publication relevance scores.

The major finding of the paper is that GBTR works well when paired with Okapi BM25. We exhaustively evaluate the performance of GBTR and compare it against the performance of existing term-ranking methods such as the Chronological Term Rank (CTR) and the Term Proximity models. Significant improvements, in terms of precision, over existing approaches are observed, measured by the major retrieval performance metric.

2 PROBLEM STATEMENT AND ESTIMATING DOCUMENT RELEVANCE SCORE

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**. A general approach to solve this problem is as follows:

FindTopKSimilarSet(P, S, K)

Input:

S: {the set of publications of the LDL repository}

P: The publication being searched for.

K: the required number of similar publications to P

Output:

TKSP: Top-K similar publications to P

Begin

Step 1: Identify the set of related publication to P, that is RP

Step 2: Rank each publication in RP based on its relevance score to P

Step 3: Return top K relevant publications

End

In keyword-based search, document **D** relevance to a given set of search keywords **W** is computed as the similarity measure between **W** and **D**. in the case of example-based search, the keywords used are those appearing in the abstract of the paper that we are searching for similar set to. One well-known document relevance estimation measure is the Okapi BM25 [18]. Okapi BM25 is a *tf.idf*-based based relevance estimation technique (tf.idf stands for term frequency / inverse document frequency)

[16]. 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 [18]. One famous and widely used group of formula to compute document relevance is the Okapi BM25 [14].

The following are a set of statistics are most commonly used in tf.idf-based text similarity estimation models like Okapi-BM25 [18; 16].

- ***tf***, term frequency, is the number of occurrences of a term in all document.
- *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.

Two famous formulas of Okapi BM25 model are [18)

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

$$\text{Score}_{\text{BM25}} = \sum_{t \in d \cap q} \frac{1 + \ln(1 + \ln(tf))}{0.8 + 0.2 \frac{dl}{avdl}} \cdot qtf \cdot \ln \frac{N+1}{df} \quad \ldots (2)$$

Where q is the set of query terms, i.e. the terms that we are searching for. And $d \cap q$ 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 publication that we are searching for similar set of publications.

3 USING TERM IMPORTANCE TO ENHANCE DOCUMENT RELEVANCE SCORES

3.1 Okapi IR System

Okapi is the name given to an experimental text retrieval system, based at City University, London. [14]. Okapi BM25 is a ranking equation used to retrieve documents in search engines upon relative ranking scores. Okapi BM25 has many forms. Examples are the two formulas of equations (1) and (2) above.

With the growing difficulty of achieving further retrieval improvements (higher precision and recall) using only term frequencies as in equations (1) and (2) above, there has been an increasing interest in information derived from document structure. Example of such information that can be derived the relative importance of words that appear in documents. Next we present multiple possible approaches to estimate term importance and augmenting it with the Okapi BM25 formulas.

3.2 Estimating term importance – The Chronological term rank

In [18] , 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 of a given term the position where that term appear first in the document. The motivation is that important terms are likely to occur near the beginning of documents. In [18], it has been experimentally proven that this has enhanced searching results in terms of precision [18].

The CTR model is defined as follows: let D = (t1, . . . , tn) be a document where ti are terms (words) ordered according to their appearing-sequence in D. The term rank *tr* of term *t* is the location *i* where *t* appears first in D.

CTR token-importance measure is augmented in equation (1) in multiple ways [18), two of which are

$$\text{Score}_{\text{BM25\_CTR1}} = \sum_{t \in d \cap q} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \frac{tf}{0.5 + 1.5 \cdot \frac{dl}{avdl} + tf} \cdot R_{CTR} \ldots (3)$$

$$\text{Score}_{\text{BM25\_CTR2}} = \sum_{t \in d \cap q} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \left(\frac{tf}{0.5 + 1.5 \cdot \frac{dl}{avdl} + tf} + R_{CTR} \right) \quad \ldots (4)$$

Where R_{CTR} is the CTR-based term importance score. In the first formula (equation 3), the R_{CTR} value is multiplied by the ratio before. In the second formula (equation 4), the R_{CTR} value is added to the second ratio. In our experiments, we compare the term-ranking approach used is compared to CTR in equation (4) as CTR perform well in the second formula as shown in [18].

In the two formulas above, R is computed as follows

$$R_{CTR} = C - \left(C.D.\log\left(\frac{tr-1}{20} + 10\right) / \log\left(\frac{dl}{20} + 10\right)\right) \quad \ldots (5)$$

Where C and D are constants and found to be C = 0.6 , D = 0.6 for best results [18). tr is the CTR-based term importance score and dl is the length of the document.

3.3 Estimating term importance – TextRank

In [12], Text rank is introduced and used for text summarization [12]. TextRank algorithm involves mainly two steps as we stated before. In the first, importance of words, observed in text to be summarized, is computed. For that PageRank algorithm [13] is used.

TextRank is built on the PageRank algorithm developed by Page and Brin [13] and used in Google search engine to assign importance scores to web pages [13]. The PageRank algorithm, applied on webgraphs, determines the importance of a web page *p* as the weighted average of the importances of the pages which links to *p*.

TextRank applies PageRank on a special graph built from text as follows. Vertices of this graph are the content-bearing words. That is; all words observed in text excluding stopwords or noise words such as "the", "an" and "who" [21]. A link is established between two words (vertices) if they are observed together within a pre-given window size, *W*. In [12], best choice of *W* is found to be 20 for text-summarizing purposes. In this paper, we propose replacing R in equations (3) and (4) by the TextRank score of the term instead of the R_{CTR} value as done in [18].

3.4 Estimating term importance – 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 [19; 8; 18].

In Term Proximity [18; 20], the order of search terms provided by the current user is used as an indicator of how important search terms are. At search time, documents where the search terms are observed physically close and of the same order to those provided by the user are considered to be more relevant to user interests.

One way to augment proximity information to Okapi BM25 is presented in [2]. In [16] an efficient evaluation framework including a proximity scoring function integrated within a top-k query engine for text retrieval is presented.

$$Score_{BM25\_TP}(d, q) = Score_{BM25}(d, q) + R \quad \ldots. \quad (6)$$

Where $Score_{BM25}(d, q)$ is defined in equations (1) and (2) above, and

$$R = \sum_{t \in q} \min\{1, idf(t)\}.accd(t).(k1 + 1)/(accd(t) + K) \quad (7)$$

Where *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 $K = k.[(1 - b) + b.|d|/avdl]$

Where *b*, *k1*, and *k* are configurable parameters that are set to *b=0.5* and *k=k1=1.2*, respectively [16]. 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 graph-based TextRank scores. Thus, the R part of becomes as follows

$$R = \sum_{t \in q} \min\{1, idf(t)\}.GBTR(t).(k1 + 1)/(GBTR(t) + K) \quad \ldots (8)$$

The configurable parameters are assigned the same values used in [16; 2].

4 USING GBTR TO ENHANCE RELEVANCE SCORE IN EXAMPLE-BASED SEARCH

Google's PageRank [13] have been successfully used in citation analysis, social networks, and the analysis of the link-structure of the WWW. PageRank is a graph-based ranking algorithm that decides on the importance of a *vertex(webpage)* within a graph (web-graph) by taking into consideration the global linkage information recursively computed from the entire web graph. Link or web graph in the context of the web is citation structure between the set of webpages [13].

Applying a PageRank on semantic graphs extracted from natural language text produces a graph on which PageRank can be applied. 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 [12].

4.1 The Text Rank Model

The TextRank algorithm is based on PageRank [TR; 13]. PageRank is based on the following two assumptions [13; TR]: *Assumption 1*: "When one vertex links to another one, it is basically casting a vote or recommendation for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex." And *Assumption 2*: "The importance of the vertex casting the vote determines how important the vote itself is"

Formally, let *G=(V,E)* be a *directed graph* with the set of vertices *V* and set of edges *E*. For a given vertex *vi* , let *In(vi)* be the set of vertices that point to it, and *Out(vi)* be the set of vertices that vertex *vi* points to. The score of *vi* is recursively computed as follows [13]:

$$S(vi) = (1-d) + d * \sum_{j \in In(vi)} \frac{1}{|Out(vj)|} S(vj) \ldots (9)$$

where d is a damping factor that can be set between 0 and 1 [13]. In general, $|out(v)|$ is the number of pages cited by page v and $In(v)$ 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)* [13]. It has been found that choosing d to be 0.85 gives accurate ranking results [13].

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.

4.2 Weighted Graphs

In the context of Web surfing, it might be unusual for a webpage to include multiple citation to another page, thus, the original PageRank definition assumes graphs with equal weights of links. However, in TextRank model the graphs are build from natural language texts, and may include multiple links between the words that are extracted from text. It may be therefore useful to indicate and incorporate into the model the "strength" of the connection between two vertices v_i and v_j as a weight w_{ij} added to the corresponding edge that connects the two vertices.

Consequently, a new formula for graph-based ranking that takes into account edge weights when computing the score associated with a vertex in the graph is introduced [12].

4.3 Text as a 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. Notice that Edges can be *directed* or *undirected*, *weighted* or *unweighted* [12].

Any relation that can be defined between two text units is useful and can be added between their vertices [12]. For our experiment and implementation of TextRank, we are using a *co-occurrence* relation, controlled by the distance between word occurrences as in [12; 4]. 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 multiple values for W that ranges from 2 to the length of the entire 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 (Porter stemmer [15; 10] algorithm applied as in [12]). (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 [4].

5. EXPERIMENTAL SETUP

In this section we compare the performance of using TextRank Graph-based term-rank against the use of Chronological Term Rank [18] and the Term Proximity [2; 16] approaches.

5.1 Procedure

In order to show the effectiveness of ranking terms via the TextRank model (the GBTR 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.

5.2 Performance metrics

The following three metrics are used to measure the quality of a retrieval result:

Precision at top-K or (P@K) which 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, 15, and 20.

Average Position of Relevant Documents (APRD): this is computed as the average position of the retrieved relevant documents in the list of search results. Low APRD value implies higher quality search results.

Average Position of Irrelevant documents (APID): this is computed as the average position of the irrelevant documents in the retrieved list of search results. High APID value implies higher quality search results.

5.3 The used publication collection

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 *examples* of the singled-out one. Details of the three groups are presented next:

The Operating Systems group: 30 abstracts from the field of *operating systems*. The group is further subdivided to 10 abstracts from the topic of *memory management*, 10 from the topic of *process management* and 10 from *memory protection*. Notice that there is some degree of similarity between the three subgroups; this helped us to critically test our proposal with the existence of *topic diffusion*.

The Artificial Intelligence group: 30 abstracts from the field of *artificial intelligence*. The group is further subdivided into 10 abstracts from *genetic algorithm*s, 10 from *fuzzy logic* and 10 from *neural networks*.

The Database Systems group: 30 abstracts from the topic of *database systems*. And also further subdivided into 10 abstracts from the topic of *relational databases*, 10 from the topic of *data warehouses* and 10 from *information retrieval*.

5.4 Summary of the used equations

For our experiments, we used equation (4) above to compute $\text{Score}_{\text{BM25\_CTR2}}$ and equations (6) and (7) to compute the Term Proximity (TP) score ($\text{Score}_{\text{BM25\_TP}}$). We augmented the TextRank (GBTR) score of terms with Okapi BM25 using the formula of equation (10) shown next (which is similar to equation (4) above).

$$\text{Score}_{\text{BM25\_GBTR1}} = \sum_{t \in d \cap q} \ln \frac{N - df + 0.5}{df + 0.5} \cdot \left(\frac{tf}{0.5 + 1.5 \cdot \frac{dl}{avdl} + tf} + R_{GBTR} \right) \ldots (10)$$

The GBTR score of terms are also used with the formula of equation (6) with R being as shown in equation (8). Thus, the new formula is

$$\text{Score}_{\text{BM25\_GBTR2}}(d, q) = \text{Score}_{\text{BM25}}(d, q) + \sum_{t \in q} \min\{1, idf(t)\} . GBTR(t) . (k1 + 1) / (GBTR(t) + K) \ldots (11)$$

5.5 Evaluation step and the used Evaluation metrics

In our experiments, we use the *leave-one-out technique* to prepare search queries. That is; selecting the abstract of one of the publications from a subgroup and search for the other members of the same subgroup. The *precision at top-K* results is used as an indicator of the quality of search.

To emphasize the value of observing the set of relevant document on top of the search-results, we computed the Average Position of Relevant Documents (APRD) for all the term-ranking approaches examined and augmented to the BM25 formula. Low APRD value gives an indication that related documents appear closer to the top of search results set (usually in the first page if the search results are organized into pages).

To further emphasize the value of observing the set of irrelevant document down toward the bottom of the search result, we computed the Average Position of *I*rrelevant Documents (APID) for all the term-ranking approaches examined and augmented to the BM25 formula. In principle, higher APID value indicates higher quality search results to the user as this means that the irrelevant document that causes diffusion to the search results will be positioned far from the top of search results.

Processing each abstract to compute the TextRank scores involves (i) tokenizing it words, (ii) removing stop words (iii) stemming the tokens using the Porter stemmer (Salton, 1989; Kowalski, G. 1997) as in [12). TextRank Graph-based term rank algorithm was applied with windows sizes of 1, 2, 3, 4, 5, 10, 20, 30, 40 and the window size that encompasses all abstract length.

6 EXPERIMENTAL RESULTS AND EVALUATION

6.1 Augmenting BM25 with GBTR scores – Formula of equation (11) and equation (4)

In this section we compare the performance of GBTR (equation 11) against the *Chronological Term Rank* (equation 4) and the *Term Proximity* (equations 6 and 7). Best results were achieved for GBTR (with all the considered window sizes, namely; 1, 2, 3, 4, 5, 10, 20, 30, 40 and the case of considering all abstract length that we refer to as GBTR_DL).

Observation: (Table 1 and Figure 1) GBTR performed better than CTR and TP approaches in terms of top-K precision for all considered values of K.

Observation: **(Table 1 and Figure 1)** amongst all the considered window sizes, the best was window size of 1, then comes 3 is coming after it.

Observation: **(Table 1, Figure 1)** In general, as the window size decrease precision values increase.

This observation can be explained as follows: at relatively large window sizes, probably more noisy word appear within the window, which may result in

(i) false-relatedness between the search terms and the surrounding terms within the document.

(ii) increasing the numbers of neighboring terms around, which in turn increases the Page Rank, or the Text Rank score of the terms.

This, in turn, increases the relevance score of the paper computed via the BM25 formula, where the search terms are observed, to the query terms. Consequently, this pushes that paper up in the set of resulting relevant documents. Thus the precision value degrades.

Table 1: Precision when applying GBTR.

	p@5	p@10	p@15	p@20
GBTR1	0.667	0.549	0.439	0.365
GBTR3	0.664	0.550	0.441	0.365
CTR	0.607	0.504	0.413	0.346
TP	0.660	0.539	0.431	0.366

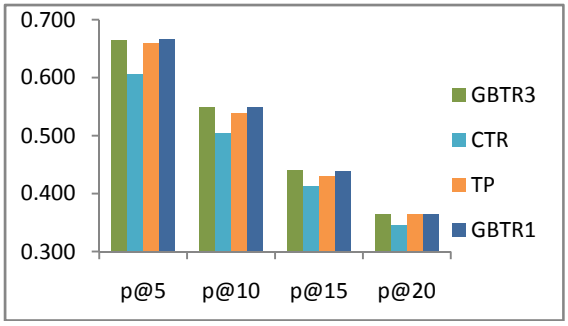

Figure 1: Comparing GBTR to CTR and TP based on the BM25 formula.

Notice for instance in table 2 that the case when the window size was 4 showed higher enhancement in precision than the case when the window size is increased to 40.

By listing the different precision values against the size of the window considered as illustrated in table 2, we clearly observe that the general trend is that precision declines as the size of the window size increases.

Table 2: Precision enhancement Of GBTR over CTR

	p@5	p@10	p@15	p@20
GBTR1	9.89%	8.81%	6.28%	5.63%
GBTR2	8.79%	8.15%	5.92%	5.79%
GBTR3	9.52%	9.03%	7.00%	5.63%
GBTR4	9.52%	8.81%	6.46%	5.31%
GBTR5	9.16%	8.37%	6.64%	5.47%
GBTR10	9.16%	7.71%	6.82%	5.63%
GBTR20	8.79%	7.05%	6.46%	5.31%
GBTR30	8.79%	6.83%	6.10%	5.14%
GBTR40	7.69%	6.39%	5.57%	4.98%
GBTR_DL	6.96%	6.17%	5.21%	4.18%
Average	**8.83%**	**7.73%**	**6.25%**	**5.31%**
Maximum	**9.89%**	**9.03%**	**7.00%**	**5.79%**

Table 2 shows clearly that when augmenting the GBTR scores of terms with the BM25 formula, a maximum of 10% enhancement of precession values over the CTR is achieved.

Observation: (Figure 2 and Table 2) Compared to (CTR) results, (GBTR) showed improvement in precision over (CTR).

This observation generally applies to all considered levels of precision; namely; precision at top-5 (p@5), top-10 (p@10), top-15 (p@15), and top-20 (p@20).

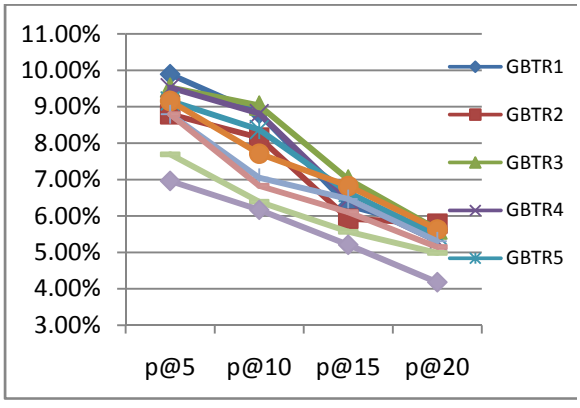

Figure 2: Precision enhancements of GBTR over Chronological Term Rank

Observation: (Table 3) Compared to (TP) results, (GBTR) showed slight improvement in precision over (TP).

This observation generally applies to all considered levels of precision; namely; precision at top-5 (p@5), top-10 (p@10), top-15 (p@15), and top-20 (p@20). But if we consider the time and space complexity required by the TP approach, the GBTR approach provides us with less time and space complexity at close or slightly better level of precision than TP.

Observation: (Table 3) Precision of GBTR is less than the precision of TP for relatively large window sizes.

This clearly observed in table 3 with window sizes of 40 and the window sizes that covers the complete abstract size. This observation can be explained as follows: at relatively large window sizes, probably more noisy word appear within the window, which, in turn, increases the relevance score of the paper computed via the BM25 formula, where the search terms are observed, to the query terms. Consequently, this pushes that paper up in the set of resulting relevant documents. Thus the precision value degrades.

Observation: (Figure 3) Enhancement of GBTR over TP reaches the peak when considering to 15 relevant documents.

As a conclusion of that, using is recommended over TP. Because users usually check to scored documents and do not usually scan the complete list of returning relevant documents.

Observation: (Table 4) GBTR with window sizes from 1 to 10 has better APRD than both CTR and TP.

This observation gives an indication that related documents appear closer to the top of search results set (usually in the first page if the search results are organized into pages). In other words, lower APRD values mean earlier appearance of relevant documents in the retrieved set of relevant documents. This observation is better emphasized in table 4 and the corresponding graph appearing in figure 4.

We have also experimentally observed that GBTR showed *higher APID* values compared to CTR and TP for *small window sizes of TextRank.*

Table 3: Enhancements of GBTR over Term Proximity Rank

	p@5	p@10	p@15	p@20
GBTR1	1.01%	1.86%	1.72%	-0.15%
GBTR2	0.00%	1.24%	1.37%	0.00%
GBTR3	0.67%	2.06%	2.41%	-0.15%
GBTR4	0.67%	1.86%	1.89%	-0.46%
GBTR5	0.34%	1.44%	2.06%	-0.30%
GBTR10	0.34%	0.82%	2.23%	-0.15%
GBTR20	0.00%	0.21%	1.89%	-0.46%
GBTR30	0.00%	0.00%	1.55%	-0.61%
GBTR40	-1.01%	-0.41%	1.03%	-0.76%
GBTR_DL	-1.68%	-0.62%	0.69%	-1.52%
Average	**0.03%**	**0.85%**	**1.68%**	**-0.46%**
Maximum	**1.01%**	**2.06%**	**2.41%**	**0.00%**

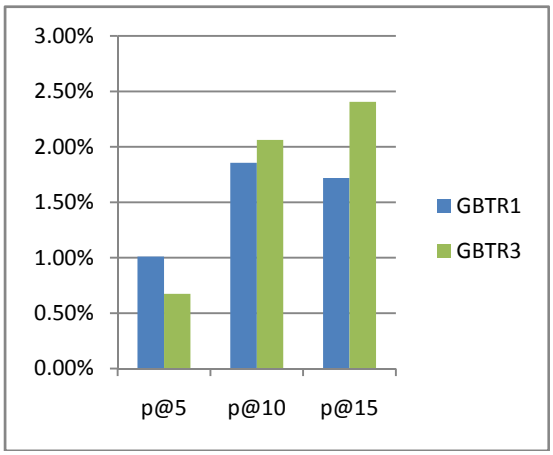

Figure 3: Enhancement of GBTR over term proximity rank–considering the BM25 formula

Table 4: The Average Position of Relevant Documents (APRD)

	p@5	p@10	p@15	p@20
GBTR1	2.300	4.056	4.456	3.200
GBTR2	2.289	4.022	4.433	3.322
GBTR3	2.256	4.067	4.500	3.122
GBTR4	2.244	4.044	4.456	3.267
GBTR5	2.233	4.011	4.467	3.189
GBTR10	2.256	4.011	4.611	3.211
GBTR20	2.233	3.978	4.789	3.211
GBTR30	2.244	3.900	4.833	3.289
GBTR40	2.222	3.944	4.644	3.400
GBTR_DL	2.222	4.000	4.656	3.333
CTR	2.156	3.978	4.611	3.278
TP	2.256	3.967	4.489	3.856

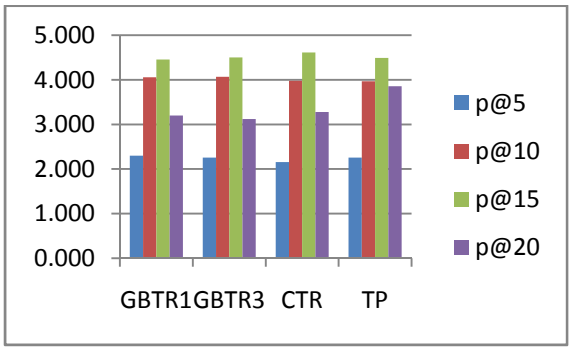

Figure 4: The average position of relevant documents (APRD) – GBTR of window sizes of 1 and 3, CTR and TP.

6.2 Augmenting BM25 with GBTR scores – Formula of equation (10)

Observation: (Table 5) Compared to CTR, GBTR showed comparable, and in some cases higher precision values, to those of CTR.

In terms of precision, the highest improvement of GBTR over CTR occurs at window sizes of 2 and 10. We also observed that GBTR has relatively higher top-K precision values for K=5. This saves the time of the current user as he/she finds what s/he is looking for on top of the search results list.

We also observed that, in general, GBTR outperforms or is comparable to TP considering equation (10) formula in terms of both APRD and APID.

6.3 Summary of Results

By Comparing GBTR to CTR and TP, we observed that

- Improved precision can be achieved by augmenting the GBTR term rank with the BM25 formulae (compared to the CTR and the TP approaches). Compared to TP results, GBTR showed improvement in precision over TP for low values of the window size (less than 10).
- GBTR with smaller window sizes performed better than the other wider window sizes in terms of precision.
- Enhancement of GBTR over TP reaches the peak when considering to 15 relevant documents. As a conclusion of that, using is recommended over TP. Because users usually check top scored documents and do not usually scan the complete list of returning relevant documents.
- GBTR with window sizes from 1 to 10 has better APRD and APID than both CTR and TP

(especially for small window sizes). This gives an indication that related publications appear closer to the top of search-result list and irrelevant documents are pushed down the list.

- GBTR showed higher APID values compared to CTR and TP for small window sizes.

Table 5: enhancements of GBTR over CTR.

	p@5	p@10	p@15	p@20
GBTR2	3.66%	0.00%	1.08%	1.13%
GBTR10	4.40%	0.44%	1.44%	2.09%

Table 6: Enhancements Of GBTR Over CTR

	p@5
GBTR2	3.66%
GBTR3	1.83%
GBTR4	0.73%
GBTR10	4.40%
GBTR20	0.73%

Table 7 : The Average Position Of Relevant Documents (APRD)

	p@5	p@10	p@15	p@20
GBTR1	2.122	4.000	4.500	3.500
GBTR2	2.156	4.033	4.656	3.344
GBTR3	2.111	4.011	5.067	3.122
GBTR4	2.122	4.011	4.389	3.444
GBTR5	2.089	3.856	5.000	3.222
GBTR10	2.200	4.044	4.656	3.478
GBTR20	2.100	4.067	4.800	3.544
GBTR30	2.067	4.144	4.367	3.667
GBTR40	2.044	4.111	4.700	3.033
GBTR_DL	2.100	3.856	4.533	3.400
CTR	2.156	3.978	4.611	3.278
TP	2.256	3.967	4.489	3.856

7. CONCLUSION

In this work we introduced enhanced relevance scores with TextRank Graph based term ranking. Intuitively, TextRank works well because it does not only rely on the local context of a text unit (vertex), but rather it takes into account information recursively drawn from the entire text (graph).

Through the graphs it builds on texts, TextRank identifies connections between various words in a text, and implements the concept of *recommendation*. A word recommends other related words, and the strength of the recommendation is recursively computed based on the importance of the words making the recommendation.

We proposed augmenting the well-known BM25 formula with the TextRank score of terms. The modified BM25 formula produces significant improvements in the precision of search results in example-based queries.

Our experiments show that GBTR has significant improvement over CTR. While with comparing GBTR with TP, (GBTR) has slight improvement (less that 2% improvement in precision) over (TP) on some of tested window sizes.

One important thing to notice that, despite the fact that the performance of TP approach is comparable to our approach, TP is computationally expensive (in both time and space requirements) compared to the GBTR approach, Thus our approach proves to be an equivalent (in terms of quality) and a suitable alternative to the TP approach with less computational overhead.

TP is also not applicable to compute similarity scores on the fly. This is a basic requirement for online literature digital libraries. That is why GBTR is recommended to be used instead of TP as it significantly reduces the query-execution time as well as space requirements.

A direct and probably important future work to this study is to apply the proposed idea into a real literature digital library.

REFERENCES

[1] ACM, http://portal.acm.org/dl.cfm. Viewed on August 2009.

[2] S. Büttcher, C. L Clarke, and B. Lushman. *"Term proximity scoring for ad-hoc retrieval on very large text collections"*. In Proceedings of the 29th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Seattle, Washington, USA, August 06 - 11, 2006). SIGIR '06. ACM, New York, NY, 621-622. DOI= http://doi.acm.org/10.1145/1148170.1148285.

[3] S. Bani-Ahmad. *"On Context-Driven Online Search-Phrase Suggesters for Large Textual Document Repositories*". IADIS International Conference Information Systems 2009, Barcelona, Spain, 25 - 27 February 2009.

[4] S. Bani-Ahmad, A. Cakmak, G. Özsoyoglu, A. Al-Hamdani. *"Evaluating Publication Similarity Measures"*, IEEE Data Eng. Bull. 28(4): 21-28, 2005

[5] S. Bani-Ahmad, A. Cakmak, G. Özsoyoglu, A. Al-Hamdani. *"Evaluating Score and Publication Similarity Functions in Digital Libraries"*. ICADL 2005: 483-485

[6] S. Bani-Ahmad, G. Özsoyoglu. *"Improved Publication Scores for Online Digital Libraries via Research Pyramids"*. In the proceeding of ECDL 2007: 50-62

[7] G. Bennett, F. Scholer, and A. Uitdenbogerd. *"A Comparative Study of Probabalistic and Language Models for Information Retrieval"*. In Proc. Nineteenth Australasian Database Conference (ADC 2008), Wollongong, NSW, Australia. CRPIT, 75. Fekete, A. and Lin, X., Eds. ACS. 65-74.

[8] D. Hawking and P. Thistlewaite. *"Relevance weighting using distance between term occurrences"*. Technical Report TR-CS-96-08, The Australian National University, August 1996.

[9] R. Jin, A. G. Hauptmann, and C. X. Zhai. *"Title language model for information retrieval"*. In the proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pages 42–48.

[10] G. Kowalski. "Information retrieval systems: theory and implementation". Kluwer Academic Publishers. First edition 1997.

[11] Luhn, H. P. 1958. The automatic creation of literature abstracts. IBM journal of Research and Development, 2:159-168.

[12] R. Mihalcea and P. Tarau. *"Textrank: Bringing order into texts"*. In L. Dekang and W. Dekai, editors, Proceedings of EMNLP 2004, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[13] S. Brin and L. Page. *"The anatomy of a large-scale hypertextual Web search engine"*. Comput. Netw. ISDN Syst. 30, 1-7 (Apr. 1998), 107-117. DOI= http://dx.doi.org/10.1016/S0169-7552(98)00110-X.

[14] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. *"Okapi at trec"*. In Text REtrieval Conference, pp. 21-30.

[15] G. Salton. *"Automatic Text Processing"*. Addison Wesley, 1989.

[16] R. Schenkel, A. Broschart, S. Hwang, M. Theobald, and G. Weikum. *"Efficient Text Proximity Search"*. In the proceeding of SPIRE 2007. Pages 287-299.

[17] ScienceDirect, www.sciencedirect.info, Viewed on August 2009.

[18] A. D. Troy, G. Zhang. *"Enhancing Relevance Scoring with Chronological TermRank"*. Proceedings of Special Interest Group in Information Retrieval (SIGIR) 2007. Pages: 599 – 606.

[19] O. Vechtomova and M. Karamuftuoglu. *"Lexical cohesion and term proximity in document ranking"*. Journal of Information Processing Management. 44, 4 (Jul. 2008), 1485-1502. DOI= http://dx.doi.org/10.1016/j.ipm.2008.01.003

[20] E. M. Voorhees and L. P. Buckland. *"Proceedings of the Fourteenth Text Retrieval Conference (TREC 2005)"*. NIST Special Publication 500-266. National Institute of Standards and Technology.

[21] Stopwords. http://en.21.org/wiki/Stop_words, viewed on Feb 5th 2009