Khulna University Studies Volume 10 (1&2) : 223-232 : June-December 2010

SCIENCE AND ENGINEERING

AN IMPROVED TERM WEIGHTING AND DOCUMENT RANKING METHOD USING RANDOM WALK MODEL FOR INFORMATION RETRIEVAL

Md. Rafiqul Islam*, Buddha Dev Sarkar and Md. Rakibul Islam

Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

KUS: 08/37-260608
Manuscript received: June 26, 2008; Accepted: September 30, 2010

Abstract: Document representation is one of the most fundamental issues in information retrieval application. The graph-based ranking algorithms represent document as a graph. Once a document is represented as graph, the similarity of that document to a query can be calculated in various ways and the calculation provides ranking to documents. This paper introduces an improved random-walk method to rank a document by considering position of a term within a document and information gain of that term within the whole document set. The experiments on various collection sets show that our approach improves the recall and precision than other proposed methods.

Keywords: Information retrieval, random walk model, term weight, term position, information gain

Introduction

Information retrieval has very important role both in the World Wide Web and desktop application. The discipline of information retrieval is almost as old as the computer itself. Intelligent information retrieval (IR) has been variously defined by different people, but a consistent theme has been one of the machine or program doing something for the user, or the machine or program taking over some functions that previously had to be performed by human either user or intermediary. An old, definition of information retrieval is the following by Mooers (1950) "Information retrieval is the name of the process or method whereby a prospective user of information is able to convert his need for information into an actual list of citations to documents in storage containing information useful to him" .

A perfectly straightforward definition along these lines is given by Lancaster (1968) "Information retrieval is the term conventionally, though somewhat inaccurately, applied to the type of activity discussed in this volume. An information retrieval system does not inform the user on the subject of his inquiry. It merely informs on the existence or non-existence and whereabouts of documents relating to his request". Generally, information retrieval deals with the representation, storage, organization, and access of information items. An information retrieval system is a software program that stores and manages information on documents. The system assists users in finding the information they need.

Corresponding author: <dmri1978@yahoo.com>
DOI: https://doi.org/10.53808/KUS.2010.10.1and2.0837-E

Information retrieval has become both more difficult and more important in recent years. This is because of the increased amount of electronic information available and greater demand for search. People are surrounded with large quantities of information, but unable to use that information effectively because of its overabundance. By improving information retrieval, we can make accesses easy to the information.

The random walk ranking algorithm on a graph proposed by Brin and Page (1998), has been used in citation analysis, social networks and analysis of the link structure of the web (Blanco and Lioma, 2007). The basic idea implemented by a random walk algorithm is that, when one vertex links to another one, it is basically casting a vote for other vertex (Hassan *et al.*, 2006). The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

Term weighting is one of the most important parts of any information retrieval system. The purpose of a term weighting method is to classify the indexing terms by assigning them weights corresponding to how well they are in improving both the recall and precision of the retrieval. Blanco and Lioma (2007) first time use graph-based ranking algorithm to weight terms in the field of information retrieval. As random walk algorithm assigns weight to terms using term dependency, it provides better result than traditional term frequency methods in the case of information retrieval (Blanco and Lioma, 2007).

Hassan *et al*. (2006) propose a random walk model for term weighting. They consider term dependencies in their approach. However, they do not considered term positions to find out the weight of term. Sun *et al.* (2004) propose an improved term weighting scheme where the term positions and information gain of term have been considered for term weight. We combine the concept of term positions approach of Sun *et al.* (2007) to the random walk model proposed by Hassan *et al*. (2006) and propose a new approach. To weight a term, we exploit the relationship of local information of a vertex (term position) as well as global information (information gain) and term dependency. Taking into account these three important factors we have done experiment and experimental results show the improvement of random walk-model providing better term weighting for information retrieval.

Related work: Blanco and Lioma (2007) use the basic, original random walk graph-based ranking algorithm and its TextRank adaptation to derive term weights from textual graphs. Textual graphs encode term dependencies in text. They plug the random walk term weights into the *tf.idf* weighting model proposed by Salton and Buckley (1988) varying the window size of co-occurring terms from 2 up to 40. The original model uses the following equation 1 to compute the score of vertex.

$$s(v_i) = (1 - d) + d \times \sum_{j \in \ln(v_i)} \frac{s(v_j)}{\left|Out\ (v_j)\right|} \qquad (1)$$

Where $s(v_i)$ = score/ weight of vertex v_i, $s(v_j)$ = score / weight of vertex v_j, $Out(v_j)$ = out degree of vertex v_j, d = a damping factor (Blanco and Lioma, 2007).

Sun *et al.* (2004) propose a method to divide the documents into several areas such as title, abstract and text, etc. The terms in the title should have higher importance than those in the abstract, the terms in the abstract should have higher importance than those in the text, and so on.

Therefore, computing method of tf_{ij} uses some factors to reflect the importance of term position, shown in the equation 2.

$$tf_{ij} = \alpha\ tf_{ij1} + \beta\ tf_{ij2} + \gamma\ tf_{ij3} \quad (2)$$

Where, tf_{ijk} is the frequency of a term in the k^{th} area and *α, β, γ* are the factors that can be adjustable according to pre-experiments. Here $\alpha > \beta > \gamma >= 1$.

Moreover, to give more importance to a term that has discriminating power to identify a document, Sun *et al*. (2004) introduce information gain (IG_j) for a term *j*. To calculate *tf.idf* the equation given as bellow

$$tf_{ij} \times idf_j = \frac{tf_{ij} \times \log\left(\frac{N}{n_j} + 0.01\right) \times IG_j}{\left(\sum_{t \in d}\left[tf_{ij} \times \log\left(\frac{N}{n_j} + 0.01\right) \times IG_j\right]^2\right)^{0.5}} \quad (3)$$

Where tf_{ij} = Frequency of term *j* within document *I*, $idf_j = \log\left(\frac{N}{n_j} + 0.01\right)$ = Inverse document frequency of term *j*, n_j = Number of documents holding term *j*, *N* = Total number of documents in the corpus (Sun *et al.*, 2004).

Hassan *et al*. (2006) introduced a system that models the weighting problem as a "random-walk" rather than "random choice". They assumed an imaginary reader who steps through the text on a term by term basis. In this setting, the importance of the term is determined by the probability of the random-walker to encounter the target term in the text during the walk.

Hassan *et al*. (2006) followed several variations of random-walk models in their work, those are summarized as follows:

rw_0: It represents the basic, original model, as described in equation 1, in which it uses an undirected graph with a constant damping factor that adheres strictly to the traditional formula of PageRank.

$rw_e \cdot idf$: This model represents an undirected graph approach that uses the weighted edge version of PageRank with a variable damping factor. The weight of an edge is calculated by the following formula:

$$E_{v_1,v_2} = tf \cdot idfv_1 \times tf \cdot idfv_2 \quad (4)$$

Where, E_{v_1,v_2} is the edge connecting v_1 to v_2 and $tf \cdot idfv_1$, $tf \cdot idfv_2$ represent the term frequency multiplied by the inverse document frequency respect to vertices v_1 and v_2. The damping factor is expressed as a function of the 'incoming edges' weight, calculated as follows:

$$d_{E_{v1,v2}} = E_{v_1,v_2} / E_{\max} \quad (5)$$

Where $d_{E_{v1},v2}$ is the damping factor and $E_{\max}$ represents the highest weight for an edge in the graph. Thus the score of a vertex can be calculated by the formula is as follows:

$$s'(v_a) = \frac{(1-d)}{|N|} + \sum_{v_b \in \ln(v_a)} C \times \frac{d_{E_{v_a},v_b} \times s(v_b)}{|Out(v_b)|} \quad (6)$$

Where, $s'(v_a)$ = Score of vertex v_a, N = Total number of nodes, C = Scaling constant.

Here the new measure of term weighting integrates both the locality of a term and its relation to the surrounding context (Hassan *et al.,* 2006).

Mihalcea and Tarau (2006) introduced the TextRank graph-based ranking model for keyword and sentence extraction from natural language texts. Here, a new formula is introduced for graph-based ranking that takes into account edge weights while computing the score associated with a vertex in the graph.

$$WS(v_i) = (1-d) + d \times \sum_{v_j \in \ln(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{jk}} WS(v_j) \quad (7)$$

Where, $WS(v_i)$ is the weighting score of vertex v_i, d denotes the damping factor and w_{ji} is the edge weight of the vertex v_i and v_j (Mihalcea and Tarau, 2006).

Drawbacks of previous work: Blanco and Lioma (2007) do not take into account edge weights. Using edge weights, final vertex scores differ significantly as compared to their unweighted alternatives (Mihalcea and Tarau, 2006).

Hassan *et al*. (2006) do not consider the term position in a document to compute the edge weight, which can play very important role to express a document (Sun *et al.*, 2006). The method described by Mihalcea and Tarau (2006) has the similar problem.

The term weighting method described by Sun *et al.*(2004) for vector space model is a 'bag-of-words' model. Hassan *et al*. (2006) argue that the bag-of-words model may not be the best technique to capture term importance; instead a method that takes into account the structural properties of the context could lead to a better term weighting scheme.

Materials and Methods

We implement our proposed method using Terrier Information Retrieval System. As it is open source, its document representation and term weighting modules are modified to cope up with our proposed method to improve precision and recall.

To evaluate the effectiveness of the modified random-walk model, the experiments are made with various reference collections. Characteristics of collection sets are shown in Table 1.

Table 1. Properties of collection sets.

Reference collections	*Numberof distinct terms*	*Number of documents*	*Average numberof terms per document*	*Number of queries*	*Average number of terms per query*	*Average relevant documents per query*
CRAN	2601	1400	14.2	56	5.3	15
CFC	2105	1239	12.2	64	4.0	39
CACM	8716	1602	46.6	50	12.7	13
CISI	9728	1460	53.6	50	9.4	50
TREC-3	1749555	741855	301.1	50	18.58	106.38

The random walk models described previously do not consider term position within a document and information gain of term to compute the weight of that term. So, the weighting of terms calculated using above models does not reflect the actual weight of term. We solve the mentioned problem using the following steps:

First, to identify the text units or terms, the document is tokenized using text operations stopword removals and stemming (Yun-tao *et al*., 2004). We use the list of stop words enlisted in the Smart System.

Second, for each term in the processed documents we calculate *tf.idf* and random-walk weights (rw). To calculate the *tf.idf* of each vertex we use the equation 3. With the value of *tf.idf* weight of an edge is calculated using equation 4. Weight of each edge is required to determine the damping factor $d_{E_{v_a,v_b}}$ in equation 5. The value of damping factor E_{v_a,v_b} varies in equation 6 to reflect importance of term. The more the value of damping factor, the term is more important to retrieve the related information. All the terms in the document are added as vertices in a graph to represent the document. Identifying relations that connect such terms, we draw edges between vertices in the graph. All the terms that fall in the vicinity of a given term are considered dependent term. This is represented by a set of edges that connect the terms to all other terms in a fixed window size. After constructing the graph we iterate the random walk algorithm until convergence. The threshold value of convergence is 0.0001.

At the last step, vertices are sorted on their final score. For ranking decision we use the scores associated with each vertex.

Our proposed model is a modified method of model $rw_e.idf$, which is proposed by Hassan *et al*. (2006). As $rw_e.idf$ model provides better result than the original model (Hassan *et al.*, 2006), therefore we integrate our ideas with this model.

Vector Space Model has been employed to calculate the similarity between query and document to determine the document rank. Collection sets are represented as term vector. If $T = \{t_j\}$ is term set of collection sets then the query vector v_j can be expressed v_j =

$(v_{j1}, v_{j2}, \cdots, v_{jn})$, in which v_{jk} denotes the weight of t_k in v_j. The vector $D_i = (d_{i1}, d_{i1}, \cdots, d_{in})$ denotes a document, which is represented as a graph using each term (vertex). The weight of term t_k in D_i is calculated by our proposed method. The similarity between v_j and D_i is determined by following formula.

$$s_j = \sum_{k=1}^{n} d_{ik} \times v_{jk} \tag{8}$$

The higher the s_j value of a document, the higher the similarity to the query. We set the value of v_{jk} to 1 for calculation.

***Basic example to build a graph*:** Here, we give an example to build a graph of document number .I 509 in CRAN corpus (Thomas Hofmann, 1999). In Fig. 1 a new term is added as a node in the graph. A term can only be represented by one node in the graph. Terms that co-occurs within a given window size are connected by an edge. Here we use the window size of 2. We remove the stop words and stem the remaining words to construct the graph.

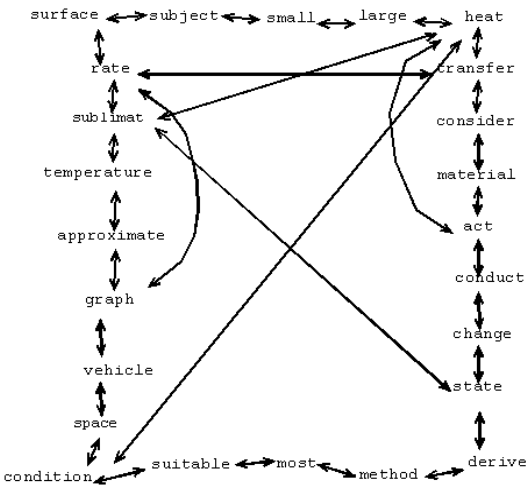

Fig. 1. Sample graph of a document.

Results

Table 2 shows weights of terms by applying, the term weighting methods upon CRAN corpus (Thomas Hofmann, 1999) using window size 2, 3 and 4; we can observe that there are significant changes in the weights of terms by our proposed method. As the words 'heat', 'surface', 'temperature', 'sublimat', 'rate' and 'transfer' are situated in the title, weights of those terms calculated by our method are greater than the Hassan *et al.* (2006) calculated values. Moreover, the value of 'sublimat' is increased highly as its information gain value is very high. As weights are computed according to the contents of documents, the proposed method provides better term weighting than other methods. During the computation of term weight, the constant value

of $\alpha_1, \alpha_2, \alpha_3$ should be adjusted according to pre experiments. We get similar results for other corpuses, those we use for our experiment. Table 3 contains the MAP for queries on various corpuses.

Table 2. Term weight according to various methods over CRAN corpus

	Weight of Terms			
Term	Hassan *et al.*	Proposed Method (Widow Size N)		
	N=2	N=2	N=3	N=4
Heat	0.01104	0.02101	0.0353	0.03642
Surface	0.01081	0.01152	0.04419	0.04428
Temperature	0.02868	0.04326	0.2795	0.26667
Sublimate	0.04776	0.26754	0.16596	0.16441
Rate	0.01701	0.02453	0.0285	0.02557
Transfer	0.02560	0.03150	0.01951	0.01340

The performance improves using our proposed method instead of Blanco and Lioma (2007) method for term co-occurring within a window of between 2 and 15 terms. Variation of the window size of co-occurring terms, affects retrieval performance.

Table 3. Mean average precision values over various corpuses

N	*Mean Average Precision*					
	CRAN		CACM		CISI	
	Blanco *et al*	Our Method	Blanco *et al*	Our Method	Blanco *et al*	Our Method
2	0.1495	0.1511	0.1495	0.2731	0.1495	0.2094
4	0.1417	0.1528	0.1417	0.2698	0.1417	0.2164
5	0.1435	0.1498	0.1435	0.2761	0.1435	0.1987
9	0.1358	0.1379	0.1358	0.2751	0.1358	0.2035
12	0.1237	0.1369	0.1237	0.2521	0.1237	0.1961
14	0.1426	0.1489	0.1426	0.2654	0.1426	0.2101
15	0.1336	0.1461	0.1336	0.2745	0.1336	0.2147

In order to allow us to understand the results we get, we have done our testing with a relatively small corpuses. By running the same tests on a larger corpus, we can get a better idea of how successful the modifications we have made to the random walk model.

Discussion

In our experiment, to see the visual improvement of our method we provide the average precision to the six points of recall. The better the performance, the plotting points will be the further up and to the right on the precision-recall graph. We compare our term weighting method and

information retrieval performance with the previous works of Hassan *et al.* (2006) and Blanco and Lioma (2007). Though Hassan *et al.* (2006) method is used for text classification but we incorporate it as well as Blanco and Lioma (2007) method in Terrier IR platform to compare with our method.

We operate the queries .I 05, .I 10, .I 28, .I 31, .I 32, and .I 41. upon the CRAN (Thomas Hofmann, 1999) corpus. For all queries we calculate the average recall and precision points. Plotting the points in the graph, we find the Fig. 2.

Over CACM (Thomas Hofmann, 1999) corpus we operate the queries .I 03, .I 14, .I 38, .I 40, .I 45, and .I 50. We find out the recall and precision points and plotting these points on the graph find the Fig. 3. Similarly over CISI corpus (Thomas Hofmann, 1999) we operate the queries .I 01, .I 15, .I 23, .I 31, .I 39, and .I 41 and find the Fig. 4.

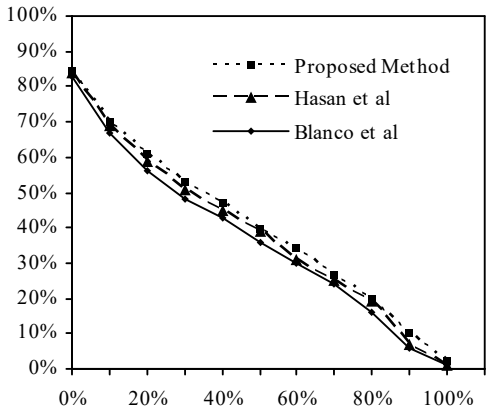

Fig. 2. Recall - Precision for CRAN.

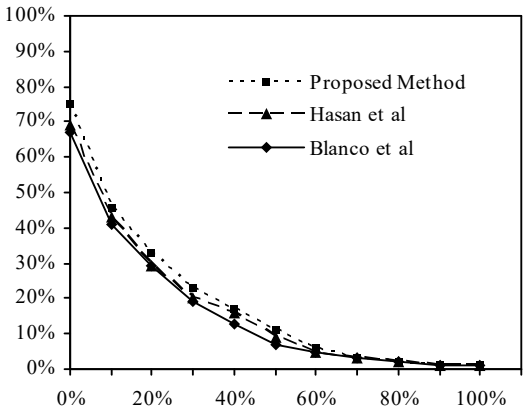

Fig. 3. Recall - Precision for CACM.

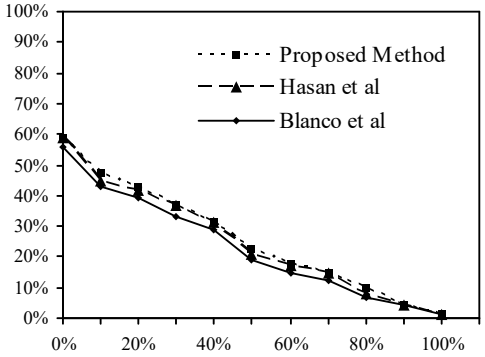

Fig. 4. Recall - Precision for CISI.

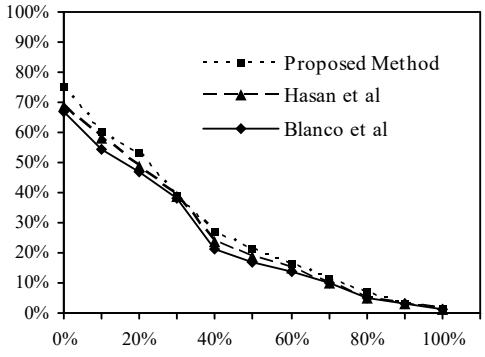

Fig. 5. Recall - Precision for CFC.

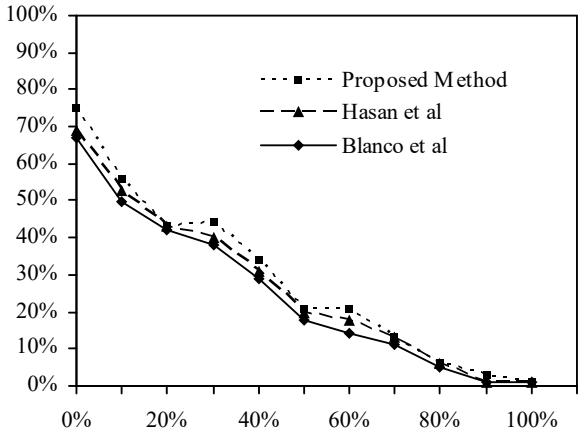

Fig. 6. Recall - Precision for TREC-3.

Over CFC corpus we operate the queries .I 05, .I 16, .I 22, .I 31, .I 41, and .I 43 and find the Fig. 5. At last, for TREC-3 corpus we operate the queries .I 03, .I 17, .I 20, .I31, .I 39 and .I 41 and find the Fig. 6.

We can see from the Table 3 and precision versus recall graphs through Fig. 2 to Fig.6 that our proposed document ranking method performs reasonably well, that achieves better precision of answer sets than Hassan *et al.* (2006) and Blanco and Lioma (2007). Therefore, we can conclude that the top ranked relevant documents will be retrieved more effectively by our method.

Future work may include the adaptation of our method with the model proposed by Mihalcea *et al.* Moreover, we want to apply our method to other fields, as Keyword Extraction, Sentence Extraction etc.

Conclusion

Efficient and effective retrieval techniques are critical in managing the increasing amount of information available in electronic form. Most existing text retrieval techniques rely on indexing keywords. Unfortunately, keywords or index terms alone cannot adequately capture the document contents, resulting in poor retrieval performance. In this paper, we show how effectively random walk model can be used for term weighting. In our method, for term weighting and document ranking, we use the two important factors, term positions and information gain of term with term dependency by combining the method proposed by Sun *et al.* to the random walk model proposed by Hassan *et al.* The experimental results of our method over various corpuses show the improvement of precision and recall to retrieve documents.

References

Blanco, R. and Lioma, C. 2007. Random Walk Term Weighting for Information Retrieval. In: *Proceedings of Special Interest Group Information Retrieval* Amsterdam, Netherlands

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN system*

Hassan, S., Mihalcea, R. and Banea, C. 2006. Random Walk Term Weighting for Improved Text Classification. In: *Proceedings of TextGraps: 2nd Workshop on Graph Based Methods for Natural Language Processing* ACL: 53-60

Lancaster, F.W.1968. *Information Retrieval Systems: Characteristics, Testing and Evaluation*. Wiley, New York

Mihalcea, R. and Tarau, P. 2006. TextRank: Bringing Order into Texts. In: *Proceedings of Empirical Methods in Natural Language Processing* ACL: 404-411

Mooers,C.N. 1950. Information retrieval viewed as temporal signaling. pp 572-573. In: *Proceedings of the International Congress of M athematicians*. Volume 1

Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal* 24(5): 513-523

Sun, Y., He, P. and Chen, Z. 2004. An Improved Term Weighting Scheme for Vector Space Model. In: *Proceedings of the Third International Conference on Machine Learning & Cybernatics*. Shanghai

Yun-tao, Z., Ling, G. and Yong-cheng, W. 2004. An improved TF-IDF approach for text classification. *Journal of Zhejiang University SCIENCE*:

Hofmann, T. 1999. Probabilistic Latent Semantic Indexing. In: *Proceedings of the Twenty-Second Annual International SIGIR Conference on Research and Development in Information Retrieval*