Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 01-009

A Balanced Term-Weighting Scheme for Effective Document Matching

Yunjae Jung, Haesun Park, and Ding-zhu Du

February 07, 2001

A Balanced Term-Weighting Scheme for Effective Document Matching

Yunjae Jung*, Haesun Park†, and Ding-Zhu Du‡
Department of Computer Science and Engineering
University of Minnesota, Minneapolis, MN 55455

ABSTRACT

A new weighting scheme for vector space model is presented to improve retrieval performance for an information retrieval system. In addition, a dimension compression method is introduced to reduce the computational cost of the weighting approach. The main idea of this approach is to consider not only occurrence terms but also absent terms in finding similarity patterns among document and query vectors. With a basic information retrieval development system which we are now developing, we evaluate the effect of the balanced weighting scheme and compare it with various combinations of weighting schemes in terms of retrieval performance. The experimental results show that the proposed scheme produces similar recall-precision results to the cosine measure, but more importantly enhances retrieval effectiveness. Since the scheme is based on the cosine measure, it is certain that it has insensitivity to weight variance. The results have convincingly illustrated that the new approach is effective and applicable.

*(e-mail: yunjae@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant CCR-9901992.

†(e-mail: hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grants CCR-9509085 and CCR-9901992.

‡(e-mail: dzd@cs.umn.edu).

1 Introduction

Computer users in this generation live in a world inundated with enormous volumes of data. The dilemma of the users dealing with the data is not the deficiency of the data but the difficulty of finding relevant data accurately. An Information Retrieval(IR) system assists the users to store, manipulate and retrieve useful data in the form of a document [7]. A fair amount of research has been carried out on similarity measures and weighting schemes, and on variations of their implementations to enhance retrieval performance. Most of the similarity measures [5, 14, 11] and weighting schemes [12, 16, 10, 1] are based on the inner product and the cosine measures. In this paper, we present a simple term-weight scheme for relevant document retrieval based on the cosine similarity measure.

An IR system based on a vector space model uses term-document matrix and term-vector representation as follows

$$A = [d_1, d_2, \ldots, d_n], \tag{1}$$

$$d_j = (t_{j1}, t_{j2}, \ldots, t_{jm})^T, \tag{2}$$

where each t_{ij} identifies a content term of the document d_i for finding a relevant document using indexing, weighting, term-matching, ranking and feedback [12]. In the vector space model, a document is located as a point in an m dimensional vector space where the dimension is the same as the number of terms in the data collection. Indexing is a procedure that transforms documents into digitized data structures to represent logical concepts of the documents. In term weighting, positive weights are assigned to the index terms. The occurrence of a term represents its proportional significance in representing the document concept. On the contrary, rareness of a term among documents discriminates the document containing the term from other documents with only frequently occurring terms in the data collection [4]. The text retrieval conducts term matching and ranks all documents by the degree of relevance in decreasing order of similarity.

In a vector space model, the commonly used basic operation for measuring document similarity is the inner product. The cosine measure has been one of most document similarity measures due

to its insensitivity to weight variation and sensitivity to document vector pattern. This measure is based on the inner product operation and the normalization by document length. Since the cosine similarity measure is insensitive with respect to radial and large component influence, it gives higher similarity rating when two document vectors have similar patterns [5].

Existing term-weighting schemes assign zero weights to absent terms in the vector space model. As the inner product operation is achieved via componentwise multiplications of the vectors, the absent term weights which are zeros mask the corresponding occurrence term weights. To resolve the problem, we suggest a new weighing method, Balanced Term-Weighting Scheme (BTWS), which applies negative term weights using inverse document frequency and document length normalization. The BTWS makes absent terms contribute to document similarity when the corresponding terms are both absent, and decrease the similarity only when one term of the pair is absent. The basic premise under the BTWS is that the document similarity is maximized if there is a perfect matching not only in occurrence terms but also in absent terms, because a similarity measure should give higher similarity to more similar term-weighting distributions.

Even though the normalization by document length reduces the undesirable effect of masking-by-zero problem, the cosine measure may produce a ranking different from that obtained by BTWS. This different ranking implies different retrieval performance. We tested the weighting scheme using various data sets to verify its practical applicability [6]. In our experimental results on BTWS retrieval performance, we illustrate that the cosine measure can be replaced with BTWS to achieve performance improvement without deteriorating the insensitivity to weight variation. Although the BTWS is not always the best weighting scheme, we believe that it is the first approach that takes account of the effect of masking-by-zero problem and considers the use of negative weights to alleviate this problem.

The rest of the paper is organized as follows. The new term-weighting scheme is illustrated in Section 2. In Section 3, a dimension compression method for the BTWS is described. In Section 4, the experimental results are presented, which indicate successful applicability of the method.

2 Balanced Term Weighting Scheme based on Vector Space Model

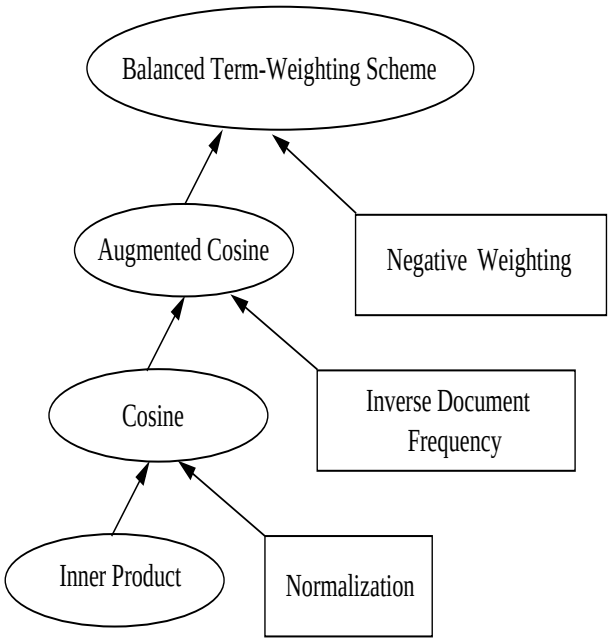

Figure 1: Hierarchical structure of balanced term-weighting scheme

In all existing methods of weighting, consider only occurrence terms are considered, and non-existing terms are encoded as zeros. On the contrary, BTWS takes count of absent terms as well. Additionally, it uses the cosine measure and complements statistical loss of term information caused by the masking by zero problem. The main procedures of BTWS consists of assigning weights, applying the inverse document frequency and performing the normalization by the document length, and conducting the inner product operation to determine the final document similarity. After assigning negative weights to absent terms, BTWS normalizes positive and negative weights independently according to global and local importance of each term.

According to the experiments, the unified normalization does not achieve high retrieval performance because of the mutual interaction between positive and negative weights. Hence, BTWS handles occurrence terms and absent terms separately. The basic concepts of the balanced term-

weighting scheme is outlined by Figure 1.

Parameters used throughout this paper are illustrated as follows.

f_i = term frequency of term i

w_i = weight of term i: w_i^d and w_i^q are for document and query, respectively.

n_i = number of documents having term i

α_i = inverse document frequency of term i

t = number of occurrence terms in a vector. t^d and t^q are for document and query, respectively.

For an occurrence term, BTWS applies term frequency, inverse document frequency and finally normalizes each document. The basic weighting formula for occurrence term w_i is expressed as

$$w_i = \frac{f_i \cdot \log_2(\frac{n}{n_i} + 1)}{\sqrt{\sum_j^t f_j^2 (\log_2(\frac{n}{n_j} + 1))^2}}, \tag{3}$$

where adjustment constant one is added in the logarithm with base two so that the dominant terms have no contribution to similarity when the inverse document frequency $\frac{n}{n_i}$ is equal to one, i.e., $n = n_i$.

For absent term, a negative constant value is assigned as a basic weight of the absent term:

$$w_i = -a, \tag{4}$$

where a is a positive constant. The assumption behind this basic weighting is that an absent term is counted a number of times in a document and the importance of a concept is proportional to the frequency of the term that represents the concept. The number of occurrences of a term becomes the basic weight of the occurrence term. In the same context, it is assumed that an absent term, for example term i, is counted as many times as a_i.

In general, the number of absent terms dominates that of occurrence terms both in a document and a query since the term-document matrix is typically very sparse in a large document collection.

Therefore, it is important to reduce the unexpected effect caused by the dominant absent terms. A critical problem is how to assign negative weights to the absent terms so that both occurrence and absent terms affect the similarity measure in such a way to improve overall retrieval performance. BTWS prevents the side effect caused by the dominant absent terms by applying the inverse document frequency to the absent terms before performing document length normalization. When we apply the inverse document frequency to the basic weight of an absent term, the formula reflects the global importance of the term to amplify the discriminant term weights as

$$w_i = -a_i \cdot \log_2(\frac{n}{n - n_i} + 1), \tag{5}$$

where the denominator $(n - n_i)$ represents the total number of documents that contain term i. The total number of documents of a data collection, n, is usually greater than the number of documents containing a term, n_i. When the inverse document frequency, n/n_i, is very large, the amplified weight results the dominant component problem [5]. Hence, the adjustment is applied not only to positive weight but to negative one using logarithm.

Local importance of a weighted term is reflected in the vector representation through L_2(Euclidean) norm. The normalization by the document length only with absent terms produces an equation as

$$w_i = \frac{-a_i \cdot \log_2(\frac{n}{n-n_i} + 1)}{\sqrt{\sum_{k=1}^{m-t}(a_k \cdot \log_2(\frac{n}{n-n_k} + 1))^2}}. \tag{6}$$

In the above equation, the absent term frequency a_i can be canceled out since the value is constant for all absent terms. Thus we can assume that the weight of an absent term is minus one, and the Eqn.(5) is simplified as

$$w_i = \frac{-\log_2(\frac{n}{n-n_i} + 1)}{\sqrt{\sum_{k=1}^{m-t}(\log_2(\frac{n}{n-n_k} + 1))^2}}. \tag{7}$$

Especially for a query, n is equal to one but n_i equals zero since only one query vector is considered. Thus the equation becomes more simplified expression as

$$w_i^q = \frac{-1}{\sqrt{m-t}}. \tag{8}$$

In the last step, the inner product operation is applied to two vectors as

$$Similarity(D,Q) = \frac{1}{2} \cdot \sum_{i=1}^{m} w_i^q \cdot w_i^d + \frac{1}{2}, \tag{9}$$

where

$$w_i^d = \begin{cases} \frac{f_i \cdot \log_2(\frac{n}{n_i}+1)}{\sqrt{\sum_j^t f_j^2 (\log_2(\frac{n}{n_j}+1))^2}} & \text{for occurrence terms} \\ \\ \frac{-\log_2(\frac{n}{n-n_i}+1)}{\sqrt{\sum_{k=1}^{m-t} (\log_2(\frac{n}{n-n_k}+1))^2}} & \text{for absent terms} \end{cases}$$

and

$$w_i^q = \begin{cases} \frac{f_i \cdot \log_2(\frac{n}{n_i}+1)}{\sqrt{\sum_j^t f_j^2 (\log_2(\frac{n}{n_j}+1))^2}} & \text{for occurrence terms} \\ \\ \frac{-1}{\sqrt{m-t}} & \text{for absent terms.} \end{cases}$$

3 Dimension Compression Method

A problem of the balanced term weighting scheme is the computational cost. Since BTWS uses every term in a vector for weighting, its computational complexity for matching becomes $O(m \cdot n)$. For the purpose of reducing the computation cost, we applied a dimension compression method without the loss of retrieval performance. The dimension compression of BTWS is a combination of two approximations. First, redundant inner product operations between negative weights are avoided. Given a document collection, the maximum document length is much smaller than the number of terms of the data collection. If we permute the terms in a document vector so as to put all absent terms after occurrence terms as in Figure 2, there are at least $m - (t^q + t^d_{max})$ negatively weighted terms both in the query and documents. Two redundant vectors, rd_i and rq_j, represent the subvectors with the same number of negative terms in a document and a query, respectively. Even though the term permutations of the vectors are different, similarities between rd_i and rq_j are almost the same for all i's because the mean weights of rd_i are nearly equivalent given the query

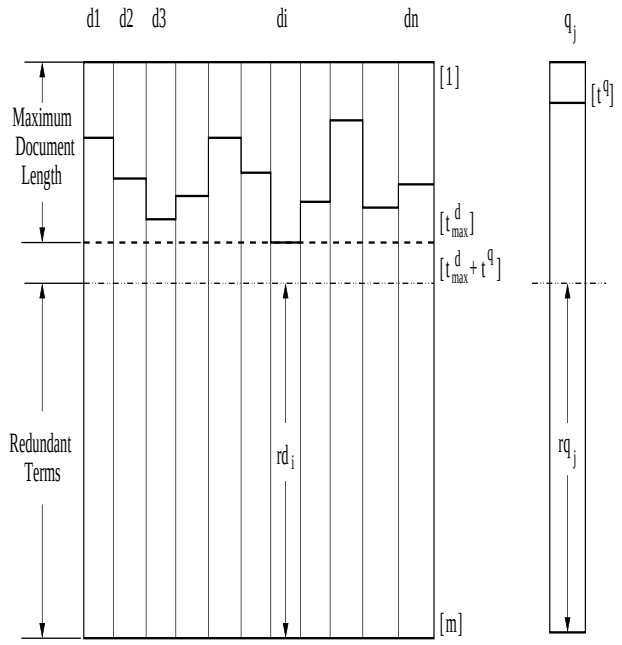

Figure 2: Dimension compression by eliminating redundant absent terms

vector rq_j. On the basis of this observation, we are able to make dimension compression to get an approximated similarity between a document and a query vector with at most $t^q + t^d_{max}$ terms.

The compressed vectors are built up using only occurrence terms that appear either in document or in query vector as in Figure 3. We begin with d and q that contain only occurrence terms and are sorted by term identifier(tid). Given i and j, we compare tids in $d[i]$ and $q[j]$. If a term appears only in $d[i](q[j])$, the term weight is copied to $dr[k](qr[k])$ and negative term weight is assigned to $qr[k](dr[k])$, and then increment $i(j)$ and k by one. Otherwise, we copy $d[i]$ to $dr[k]$ and $q[i]$ to $qr[k]$, and increment i, j and k.

Secondly, the sum of absent term weights in a compressed vector is replaced with approximated sum that will be used for the normalization of the compressed vectors. The sum of negatively

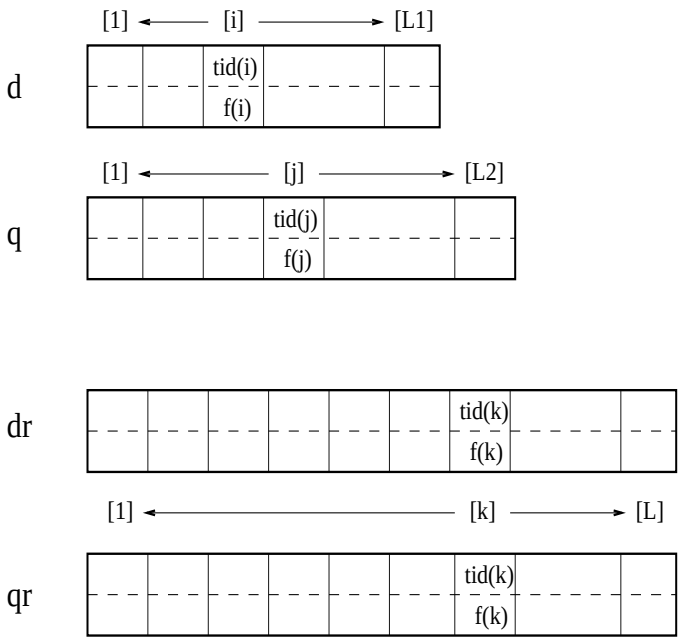

Figure 3: Compressed Vector Representations

weighted terms, S, is defined as

$$S = \frac{\Sigma_{i=1}^{L} I(i) \cdot f(i)}{L - t'} \cdot (m - t), \tag{10}$$

$$I(i) = \begin{cases} 0 & \text{if } f(i) \geq 0,\ i = 1, 2, \cdots, L \\ 1 & \text{otherwise,} \end{cases} \tag{11}$$

where t' is the number of occurrence terms in the compressed vector. According to the experimental results in Figure 4, this dimension compression method saves computational cost dramatically without loss of retrieval performance. When the inverted file structure is used with the dimension compression method, m and n decrease to $t^d_{max} + t^q$ and n_{max}, respectively. Consequently, the computational cost of BTWS is decreased and above by $O(t^d_{max} \cdot n_{max})$, where t^d_{max} and n_{max} are much smaller than m and n, respectively.

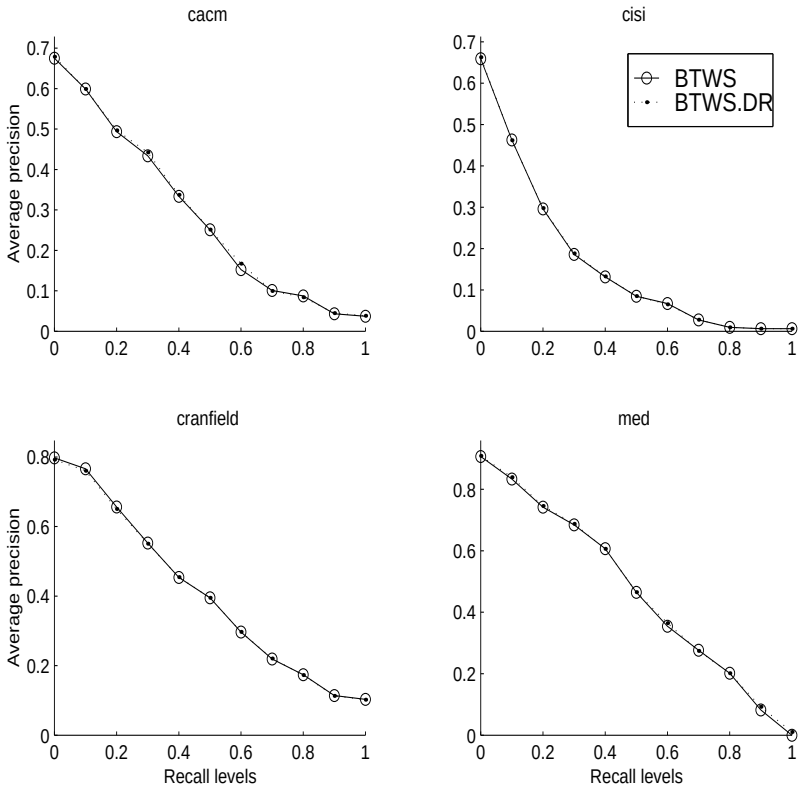

Figure 4: Effect of dimension compression on retrieval performance

4 Test Environment and Experimental Results

In the previous report [6], the retrieval performance of BTWS was evaluated using SMART retrieval system [1] with Cranfield and Med data collections. The results of the test showed successful applicability of the scheme. To evaluate reliability of BTWS, we implemented a Basic Information Retrieval Development System (BIRDS), applied the scheme to various data sets and measured retrieval performance using the recall and precision [14]. In BIRDS, we were able to test effects of various weighting schemes, and make comparisons among them with more flexibility than SMART IR system [2]. The configuration of the system is depicted in Figure 5.

The BIRDS adopted the stop-list from SMART retrieval system and used Porter's Algorithm for stemming [9]. Even if a term in a query does not appear in the stop-list, the term is removed from

[1] ftp.cs.cornell.edu/pub/smart/

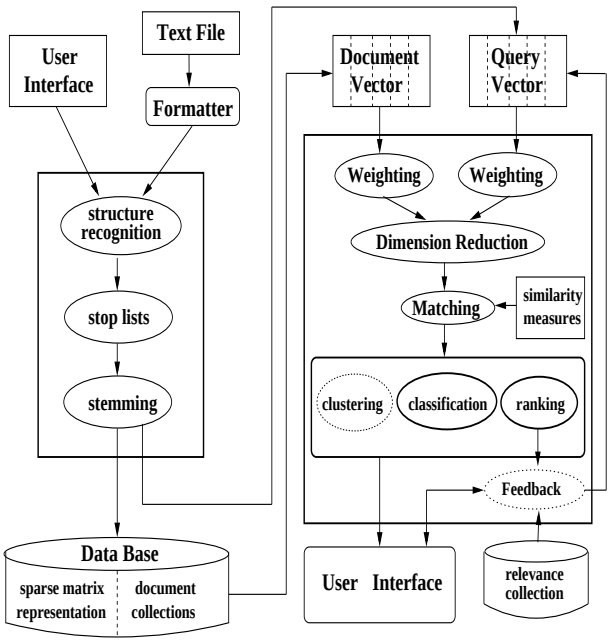

Figure 5: Basic Information Retrieval Development System(BIRDS)

the query vector when the term does not occur in document dictionary. If the inverse document frequency of a term equals one, the term is ruled out from the vector since the term does not contribute to document discrimination. There is no noun-grouping in BIRDS. Because of difference between preprocessing procedures, the statistical information of these data sets might be a little different from those of other experiments. Data sets for this experiment are described in Table 1 and their statistical information is presented in Table 2 and Table 3.

We used interpolated average precision at each recall level to measure retrieval effectiveness among several weighting schemes [2]. The recall is the ratio of relevant documents retrieved for a given query over the number of relevant documents for that query in a data collection. Precision is defined as the ratio of the number of relevant documents retrieved over the number of retrieved documents. The precision is computed after each relevant document is retrieved. All precision values are then averaged together to obtain a single number for the performance of a query. The weighting schemes and similarity measures are shown in Table 4. M2(Method 2) and BFWS(best fully

Classes	*Contents*
ADI	Library Science
CACM	Computer Science
CISI	Information Science
CRAN	Aeronautics
MED	Biomedicine collections
TIME	World news articles from 1963 Time magazine
CR93e	Congressional Records 1993
CR93h	Congressional Records 1993

Table 1: Test data collections

	size	No. of Terms	Density	Avg. Length	Avg. Term Freq.
ADI	82	972	0.0268	26.0122	34.1829
CACM	3204	7307	0.0037	27.0652	38.2603
CISI	1460	6965	0.0065	45.3507	62.3788
CRAN	1400	5364	0.0102	54.7407	92.0400
MED	1033	8714	0.0059	51.0726	76.7706
TIME	425	13971	0.0142	197.8635	274.7671
cr93e	11358	51315	0.00331	170.0312	318.4555
cr93h	16564	55756	0.00338	188.5712	815.5536

Table 2: Statistical comparison of test data collections(Documents)

	size	Density	Avg. Length	Av. Term Freq.
adi	35	0.0067	6.5143	6.9714
cacm	64	0.0016	11.7656	14.4688
cisi	112	0.0042	28.9554	40.8214
cran	225	0.0016	8.8044	9.1911
med	30	0.0011	9.8667	10.9667
time	83	0.0006	8.0964	8.4217
cr93e	50	0.00053	27.2000	40.5400
cr93h	50	0.00049	27.3400	40.6800

Table 3: Statistical comparison of test data collections(Queries)

number	*name*	*weighting*	*similarity measure*
1	BTWS	$w_i^d = \begin{cases} \frac{f_i \cdot \log_2(\frac{n}{n_i}+1)}{\sqrt{\sum_j^t f_j^2 (\log_2(\frac{n}{n_j}+1))^2}} \\ \frac{-\log_2(\frac{n}{n-n_i}+1)}{\sqrt{\sum_{k=1}^{m-t}(\log_2(\frac{n}{n-n_k}+1))^2}} \end{cases}$, $w_i^q = \begin{cases} \frac{f_i \cdot \log_2(\frac{n}{n_i}+1)}{\sqrt{\sum_j^t f_j^2 (\log_2(\frac{n}{n_j}+1))^2}} \\ \frac{-1}{\sqrt{m-t}} \end{cases}$	$\frac{1}{2}\sum_{i=1}^m w_i^d \cdot w_i^q + \frac{1}{2}$
2	Euclidean	$w_i = f_i$	$\frac{1}{\sqrt{\sum_{i=1}^m (w_i^d - w_i^q)^2}}$
3	Inner	$w_i = f_i$	$\sum_{i=1}^m w_i^d \cdot w_i^q$
4	M2	$w_i = f_i$	$\frac{\sum_{i=1}^m w_i^d \cdot w_i^q}{\sqrt{\text{length}(D)}}$
5	Cosine	$w_i = \frac{f_i}{\sqrt{\sum_{j=1}^m f_j^2}}$	$\sum_{i=1}^m w_i^d \cdot w_i^q$
6	BFWS	$w_i^d = \frac{f_i \cdot \log \frac{n}{n_i}}{\sqrt{\sum_{j=1}^m (f_j \cdot (\log \frac{n}{n_j})^2}}$, $w_i^q = \frac{1}{2} + \frac{f_i}{2 \cdot \max(f)} \cdot \log \frac{n}{n_i}$	$\sum_{i=1}^m w_i^d \cdot w_i^q$

Table 4: Comparing weighting schemes and similarity measures

weighted system) in the table were tested by Lee and Chuang [8], and by Salton and Buckley [13], respectively.

The experimental results showed that the retrieval performance of BTWS is superior to that of the cosine measure with respect to the recall-precision measurement. In most data collections, BTWS outperforms the cosine measure and its precision shape is very similar to that of the cosine measure. Overall results depicted in the Figure 6 and Figure 7, especially the result from cr93h, clearly indicate that the precision produced by BTWS is very similar to that of the cosine measure, but occupies higher positions. We have tested other combinations of weighting schemes as follows.

- balanced term-weighting only with term frequency.
- balanced term-weighting only with inverse document frequency.
- balanced term-weighting only with document length normalization.

The experimental results showed that BTWS outperforms all of the above combinations.

To provide additional performance evaluation, we used R-precision and exact precision evaluation measures. R-precision is the precision after the total number of relevant documents are retrieved for a query. Exact precision is the precision after a specific number of documents have

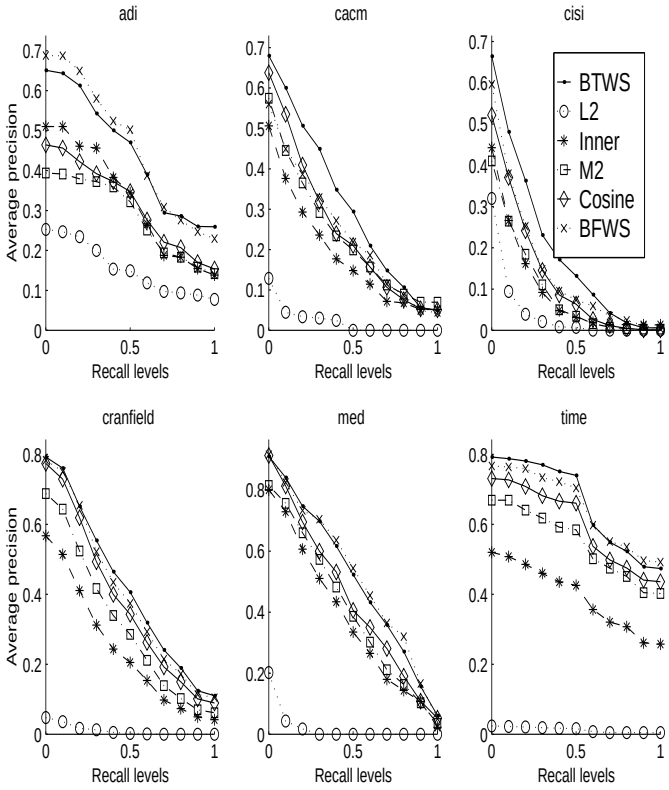

Figure 6: Average recall-precision for classic data sets

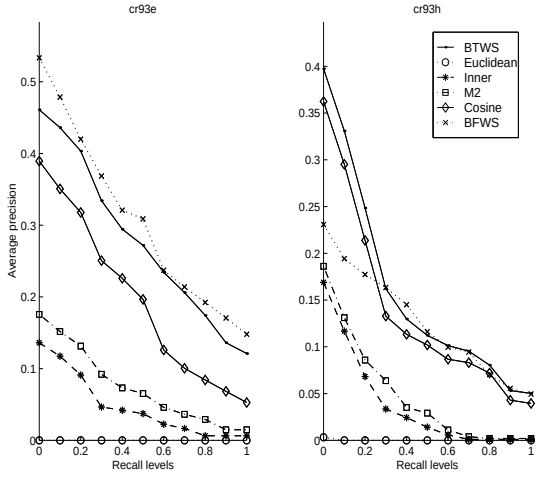

Figure 7: Average recall-precision for TREC data sets

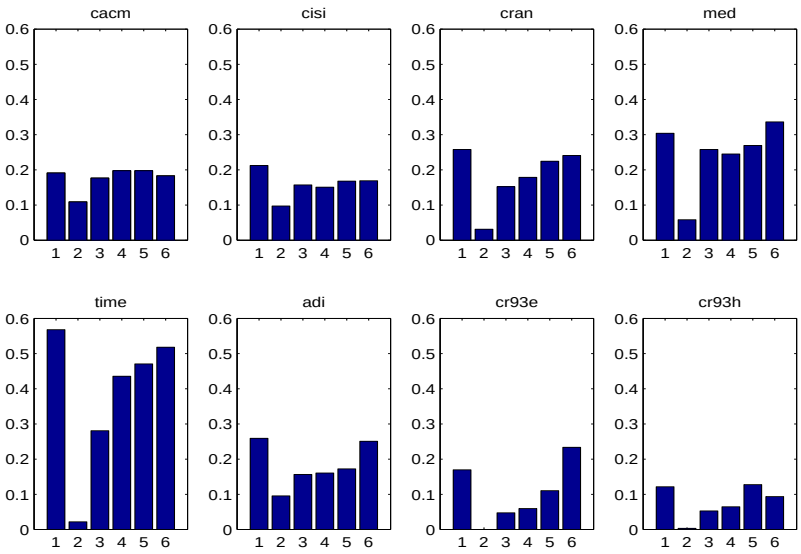

Figure 8: R-precision for classic and TREC data sets

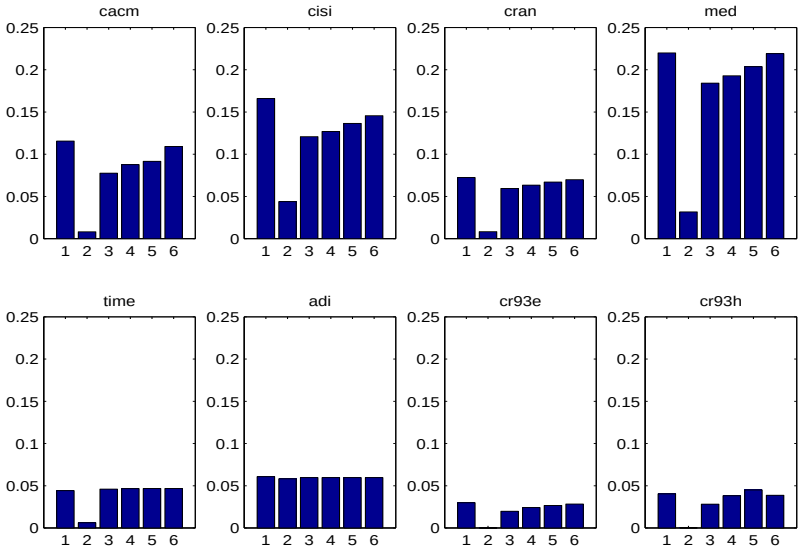

Figure 9: Exact precision for classic and TREC data sets

been retrieved. It is measured when twenty documents are retrieved. R-precision and exact precision are depicted in Figure 8 and Figure 9. From the results, it is verified that BTWS is reliable and comparable.

5 Conclusion

In this paper, we have described the new approach to term-weighting scheme called Balanced Term-Weighting Scheme(BTWS)on the basis of vector space model. Additionally, the dimension compression method has been presented to reduce the computational cost of the balanced term weighting scheme. With currently used data collections, we tested the new scheme and evaluated its retrieval effectiveness. According to the experimental results, the BTWS produces similar recall-precision pattern to the cosine similarity measure but achieves higher retrieval performance. Even though the BTWS scheme is not the best weighting scheme for all data sets, the results have convincingly illustrated that the new approach is effective and applicable. Therefore BTWS can be used anywhere the cosine measure is used and there is high expectation of improved retrieval performance with insensitivity to weight variations. In addition to text retrieval, a combination of weighting schemes and similarity measures can be directly applied to text classification as a distance measure. Applying BTWS, for example to k nearest neighbor classification [3] and centroid oriented classification [15], will be the next step to demonstrate its usefulness for classification fields.

REFERENCES

[1] K.J. Barkla. Construction of weighted term profiles by measuring by frequency and specificity in relevant items. In *The Second International Cranfield Conference on Mechanized Information Storage and Retrieval Systems*, Cranfield, Bedford, 1969.

[2] C. Buckley. Implementation of the smart information retrieval system. Computer Science Technical Report TR85-686, Department of Computer Science, Cornell University, Ithaca, N.Y. 14853, May 1985.

[3] E.S. Han, G. Karypis, and V. Kumar. Text categorization using weight adjusted k-nearest neighbor classification. Computer Science Technical Report TR99-019, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota, 1999.

[4] K. Spark Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–20, March 1972.

[5] K. Spark Jones and G.W. Furnas. Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38(6):420–442, November 1987.

[6] Y. Jung, H. Park, and D. Du. An effective term-weighting scheme for information retrieval. Computer Science Technical Report TR008, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota, 2000.

[7] G. Kowalski. *Information Retrieval Systems – Theory and Implementation.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[8] D. Lee, H. Chuang, and K. Seamons. Document ranking and the vector-space model. *IEEE Software*, pages 67–75, March/April 1997.

[9] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[10] G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing.* Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

[11] G. Salton. Automatic term class construction using relevace - a summary of work in automatic pseudoclassification. *Information Processing and Management*, 16:1–15, 1980.

[12] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. Computer Science Technical Report TR88-898, Department of Computer Science, Cornell University, Ithaca, N.Y., 1988.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[14] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval.* McGraw Hill, New York, 1983.

[15] S. Shankar and G. Karypis. Weight adjustment schemes for a centroid based classifier. Computer Science Technical Report TR00-035, Department of Computer Science, University of Minnesota, Minneapolis, Minnesota, 2000.

[16] C.T. Yu and G. Salton. Precision weighting - an effective automatic indexing method. *Journal of the ACM*, 23:76–88, 1976.