http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 5 (1): 7-12, 2006
ISSN 1812-5638

Consolidation of Diversifying Terms Weighting Impact on IR System Performances

A. Kasam and Kwon Hyuk-Chul
AI Laboratory, Department of Computer Science and Engineering
Pusan National University, 609-735, Busan, Korea

Abstract: Search engines and internet crawlers present automatically and accurately user's relevant data from a high dimensional words basket or databag which we named high dimensional vector space model, documents are stored into that large databag as a number of indexed vectors in the terms of spaces. When a document is searched, a query is given through the search engine. Mainly two major computing operations are done here, one for query vector and another document vector. The component of vectors is determined by the term weights, a function of the frequencies of the terms in the document or query as well as throughout the collection. So searching documents ultimately goes to the meaning of terms weighting into the high dimensional data space which is a difficult task in the data industries. Same times using a single method of terms weighting also suffers from certain limitations in application issues. So that highly diversified and consolidation of terms weighting approach can be applied as an interesting tool for improving retrieval performances. In this study, consolidation of diversifying terms weighting approach has been proposed as an argument of cost effective method for improving the retrieval performances. Under the proposed approach, a certain amount of Meta data has been tested and finally obtaining throughout results strongly recommend us that our approach is effective and has positive values, further applicable to promote retrieval performances.

Key words: Information retrieval, terms-weighting

INTRODUCTION

Search engines and internet crawlers present automatically and accurately users relevant data from a high dimensional word basket or databag which we usually named high dimensional vector space model, the preprocessed documents are stored into that multi dimensional databag as a number of indexed vectors in the terms of spaces. The definition of a term is not inherent in the model, but terms are typically words and phrases[1]. If words are chosen as terms, then every word in the basket becomes an independent dimension in the high dimensional vector space. A relevance feedback policy is used to construct a personalized query or user profile[2], it means when a document is searched, a query is given through the search engine. Mainly two major computing operations are separately done here, one for query vector and another document vector (Table 1 and 2). The component of vectors is determined by the term weights, a function of the frequencies of the terms in the document or query as well as throughout the collection. So searching documents ultimately goes to the meaning of terms weighting into the high dimensional data space which is a difficult problem in the data industries[3,4]. People use different types of IR system in the forms of Internet search engine to retrieve the required informations[5,6]. As search engines or Internet crawlers are ultimately based on IR models, such as Boolean Model (BM), Probabilistic Model (PM) and Vector Space Model (VSM).

The state art of the IR assists the users to store, manipulate and retrieve a volume of useful data in the forms of documents[7]. Similarities between two documents are traditionally measured by the cosine of the angle between two vectors. It is based on the inner product operation and document length normalization[8]. It is useful to give a geometric interpretation to the vector space notion of similarity by considering the dot product equation.

$$\cos\theta = \frac{d.q}{\| d \| \| q \|}$$

In a vector space model, where, d is a document vector, q is a query vector and θ is the angle between them, if d and q are normalized so that the magnitudes are one, preceding equation then reduces to θ = d.q, so the similarity score is a measure of cosine of the angle between the vectors. If we rank the documents according

Corresponding Author: A. Kasam, AI Laboratory, Department of Computer Science and Engineering
Pusan National University 609-735, Busan, Korea

Table 1: Documents from the MED test collection

Document 1			Document 2		
Term ID	Word	Weight	Term ID	Word	Weight
900	Colestorol	0.09	1420	Levels	0.07
1120	Blood	0.36	1520	High	0.07
1420	Levels	0.18	1790	Fatty	0.07
1500	Low	0.09	1850	Normal	0.44
1520	Low	0.09	1850	Normal	0.44
1680	Fatal	0.18	1890	Glucose	0.07
1890	Glucose	0.09	2021	Free	0.29
2021	Free	0.09	2450	Acid	0.29
2390	Infant	0.19	2519	Toxin	0.29
2530	Lactoz	0.07	3033	Primary	0.25

Table 2: Query from the MED test collection

Query		
Term ID	Word	Weight
1420	Levels	0.30
1500	Low	0.30
1520	High	0.30
1790	Fatty	0.25
1890	Glucose	0.30
2021	Free	0.30
2530	Lactoz	0.36
2591	Toxin	0.30
3031	Iron	0.24

to their similarity score from highest to lowest, the highest scoring document has the smallest angle between itself and the query.

Example 1: Here, the common terms between doc1 and the query are level (1050), low (1725), high (1820), glucose (2461) and lactoz (3560) and the similarity score of the Doc 1 is:

$$(0.18*0.30) + (0.09*0.30) + (0.09*0.30) + (0.09*0.30) + (0.07*0.36) = 0.1602$$

The common terms between Doc 2 and the query are high (51), fatty (1790), glucose (2168), levels (2450) and toxin (2591) and the similarity score of Doc 2 is:

$$(0.07*0.30) + (0.07*0.30) + (0.07*0.25) + (0.07*0.30) + (0.29*0.30) + (0.29*0.30) = 0.2545$$

Into the Doc 2 has a higher similarity score than the doc1, so the Doc 2 would be retrieved before the Doc 1.

METHODS OF TERMS WEIGHTING

Proper terms weighting greatly impacts on the IR system performances[9]. Here, we have explained briefly fundamental and modification of term weighting methods and retrieval contribution among the data set. A list of popular terms weighting methods is given in the Table 3. Generally, three different types of term weighting

Table 3: List of popular terms weighting

Document weight	Query weight	Scheme name
LOGA ENPY COSN	LOGA ENPY	Log-entropy
LOGA IGFF COSN	ATF1 ENPY	IGFF- entropy
FREQ NONE NONE	FREQ NONE	Raw term frequency
FREQ NONE COSN	FREQ NONE	Raw cosine
LOGA NONE COSN	LOGA IDFB	SMART
LOGA NONE COSN	LOGA IDFB	Variation of SMART
FREQ IDFB COSN	ATF1 IDFB	Best fully weighted
ATF1 NONE NONE	BNRY IDFB	Best probability
LOGN NONE PUQN	LOGA IDFB	Pivoted unique new norm weight

methods[10] local, global and length normalization are used for practical purposes. The term weighting is given by;

$$L_{ij}G_iN_j;$$

Where, L_{ij} is the local weight for term i in document j, G_i is the global weight for term i and N_j is the length normalization factor for document j. Local weights are functions of how many times each term appears in a document, global weights are functions of how many times each term appears in the entire collection and the normalization factor compensates for discrepancies in the length of the documents. The local weight is computed according to the terms in the given document or the query. The global weights, however, is based on the document collection regardless of whether we are weighting documents or queries. The normalization is done after the local and global weighting of the document vectors but for the query vectors not necessary because it does not affect the relative order of the ranked document lists. The concept of the local term weighting schemes perform well, if they work on the basic principal that the terms with higher within-document frequency are more relevant to that document. Generally binary formats[11] are used for local weight and document frequency (FREQ), given respectively by;

$$L_{ij} = \begin{cases} 1, \text{ if } f_{ij} > 0 \\ 0, \text{ if } f_{ij} = 0; \text{ and} \end{cases}$$

$L_{ij} = f_{ij}$; Where, f_{ij} is the frequency of term i in document j. These weights are typically used for query weighting, where terms appear only one or twice. The principal problem of the local weight is that BINARY does not differentiate between the terms that appear frequently or only once in the document and FREQ also gives too much weight to terms that appear frequently. The logarithm plays a middle ground to adjust the document frequency because a term may appear ten times in a document is not necessary ten times as important as a term that appears once in that document. In the bellow, two modify local weighting formulas are similar because

Table 4: Initial local terms weighted formulas

Formula	Name	Abbreviations
1 if $f_{ij} > 0$ 0 if $f_{ij} = 0$	Binary	BNRY
f_{ij}	within-document frequency	FREQ
$1+\log f_{ij}$ if $f_{ij} > 0$ 0 if $f_{ij} = 0$	Log	LOGA
$\frac{1+\log f_{ij}}{1+\log aj}$ if $f_{ij} > 0$ or $f_{ij} = 0$	Normalizing log	LOGN
$0.5 + 0.5\left(\frac{f_{ij}}{xj}\right)$	Augmented	
0 if $f_{ij} > 0$ if $f_{ij} = 0$	Normalized term frequency	ATF1

Table 5: Global terms weighting formulas

Formula	Meaning (Name)	Abbreviation
$\log\left(\frac{N}{n_i}\right)$	Inverse document frequency	IDFB
$\log\left(\frac{N-n_i}{n_i}\right)$	Probabilistic inverse	IDFB
$1+\sum_{j=1}^{N}\frac{\frac{f_{ij}}{F_i}\log\frac{f_{ij}}{F_i}}{\log N}$	Entropy	ENPY
$\frac{F_i}{n_i}$	Global frequency	IGF

each of them used logarithm. They are (Table 4) (LOGA) and normalized log (LOGN), given respectively by;

$$L_{ij} = \begin{cases} 1+\log f_{ij} & ; \text{if } f_{ij} > 0 \\ 0; & \text{if } f_{ij} = 0; \text{ and} \end{cases}$$

$$L_{ij} = \begin{cases} \frac{1+\log f_{ij}}{1+\log a_j}; & \text{If } f_{ij} > 0 \\ 0; & \text{If } f_{ij} = 0 \end{cases}$$

Where, a_j is the average frequency of the terms that appear in document j^2. Because LOGN is normalized by the $(1 + \log a_j)$ term; the weight given by LOGN will be always lower than the weight given by LOGA for the same term and document and both are suitable for local document and query weights. Another modify local weight that is a middle ground between binary and term frequency is argument normalized term frequency (ATF1)[12];

$$L_{ij} = \begin{cases} .5+.5\left(\frac{f_{ij}}{x_j}\right) & ; \text{ if } f_{ij} > 0 \text{ and} \\ 0; & \text{if } f_{ij} = 0 \end{cases};$$

Where, x_j is the maximum frequency of any term in document j, ATF1 awards weight to a term for appearing in the document and then awards additional weight for appearing frequently. The formula L_{ij} varies only between 0.5 and 1 for terms that appear in the document. The global weights try to give a discrimination value to each term. Many schemes are based on the idea that the less frequently a term appears in the whole collection. A commonly used global weight is the inverse document frequency measure, or IDF, derived by Spark Jones. We have used two variations, IDFB[12] and IDFP[13] given respectively by;

$$G_i = \log\left(\frac{N}{n_i}\right);$$

And

$$G_i = \log\left(\frac{N-n_i}{n_i}\right);$$

Where, N is the number of documents in the collection and n_i is the number of documents in which term i appears. IDFB is the logarithm of the inverse of the probability that term i appears in a random document. IDFP is the logarithm of the inverse of the odds that term i appears in random document. IDFB and IDFP are similar that they both award high weight for terms appearing in few documents and law weight for terms appearing many documents in the collection; however, they differ because IDFP actually awards negative weight for terms appearing in more than half of the documents in the collection and the lowest weight of IDFB is one. In addition, we have used the Entropy weight (ENPY)[14,15] given by;

$$G_i = 1+\sum_{j=1}^{N}\frac{f_{ij}\log\frac{f_{ij}}{F_i}}{\log N}$$

Where, F_i is the frequency of term i throughout the enter collections. Entropy is a useful weight because it gives higher weight for terms that appear several times in a small number of documents.

We also use a global weight (Table 5) frequency IDF (IGFF) given by:

$$G_i = \frac{F_i}{n_i}$$

This weight often works better when combine with different global weights on the query vector (Table 8-11). The third component of the weighting scheme is the normalization factor. It is useful to normalize the document vectors, so the documents are retrieved independently of their lengths.

Table 6: Diversify local terms weighting formulas

Formula	Name	Abbreviation
$0.2 + 0.8\left(\frac{f_{ij}}{xj}\right)$ if $f_{ij} > 0$ or if $f_{ij} = 0$	Changed-coefficient ATF!	ATFC
$0.9 + 0.1\left(\frac{f_{ij}}{a_j}\right)$ if $f_{ij} > 0$ or if $f_{ij} = 0$	Augment average term frequency	ATFA
$0.2 + 0.8 \log(f_{ij} + 1)$ if $f_{ij} > 0$ or if $f_{ij} = 0$	Augmented log	LOGG
$\sqrt{f_{ij} - 0.5} + 1$ if $f_{ij} > 0$ or if $f_{ij} = 0$	Square root	SQRT
0		

Table 7: Diversify global terms weighted formulas

Farmula	Name	Abbreviation
$\log\left(\frac{F_i}{n_i} + 1\right)$	Log-global frequency IDF	IGFL
$\frac{F_i}{n_i} + 1$	Incremental global frequency IDF	IGFI
$\sqrt{\frac{F_i}{n_i} - 0.9}$	Square root global frequency IDF	IGFS

Cosine normalization (COSN) is a popular form of normalization, given by;

$$N_j = \frac{1}{\sqrt{\sum_{i=0}^{m} (G_i L_{ij})^2}};$$

With COSN, the longer documents are given as smaller individual term weights; so that smaller documents are favored over the longer ones in retrieval. Pivoted Unique Normalization (PUQN)[16] is relatively a new normalizing method that is used to correct the problems of favoring short documents given by;

$$N_j = \frac{1}{(1 - \text{slope}) + \text{slope}\, l_j};$$

The basic principle behind the PUQN is to correct the discrepancies based on document length between the probability that a document is relevant and the probability that the documents will be retrieved.

CONSOLIDATION OF THE DIVERSIFYING TERMS WEIGHTING FORMULAS

Here, diversifying and consolidation of various terms weighting methods and their impact on the datasets have been explained (Table 6). Tow local diversifying weighting formulas are changed coefficient ATF1 (ATFC) and augmented average term frequency (ATFA), given, respectively by:

$$L_{ij} = \begin{cases} 0.2 + 0.8\left(\frac{f_{ij}}{x_j}\right); & \text{if } f_{ij} > 0 \text{ or } f_{ij} = 0; \\ 0; \end{cases} \quad \text{and}$$

$$L_{ij} = \begin{cases} 0.9 + 0.1; & \text{if } f_{ij} > 0 \text{ or } f_{ij} = 0; \\ 0; \end{cases}$$

ATFC is developed using a general version of ATF1

$$L_{ij} = \begin{cases} K + (1 - K)\left(\frac{f_{ij}}{x_j}\right); & \text{if } f_{ij} > 0 \text{ or } f_{ij} = 0; \\ 0; \end{cases}$$

Changed-coefficient ATF1 works well because it assigns weight to a term merely for appearing in a document, then adds more weights, if the term appears frequently in the document. ATFA is similar to ATF1 but it is normalized differently. ATF1 is normalized by the maximum within document frequency of a particular document and ATFA is normalized by the average within-document frequency of a document also the coefficients are different. ATFA gives more weights to a term for just appearing and adds less weight if a terms appears frequently. It is noted that the maximum value of the ATFC is one, whereas one is the average value for ATFA. Another modify new local weight is augmented log (LOGG), a variation of ATFC, given by;

$$L_{ij} = \begin{cases} 0.2 + 0.8 \log(f_{ij} + 1) \\ 0; \end{cases}; \quad \text{if } f_{ij} > 0 \text{ or } f_{ij} = 0;$$

We simply modified (f_{ij}/x_j) to $\log(f_{ij}+1)$ because log seems to be a better local weight than within-document frequency. Note that now L_{ij} can be greater than one. Another modify new local weight is square root (SQRT); given by;

$$L_{ij} = \begin{cases} \sqrt{f_{ij} - 0.5} + 1; & \text{if } f_{ij} > 0 \text{ or } f_{ij} = 0; \\ 0; \end{cases}$$

In the development of SQRT, we model the formula resembled (Table 4) that of LOGA, a top performer (Table 8-11) among established local weight formulas. We looked at the graph of LOGA and noted that the function $\sqrt{f_{ij}}$ would have a similar shape. As f_{ij} gets large and SQRT has a larger value than LOGA. We have three new global weights (Table 5), the first is log-global frequency IDF (IGFL), given by:

Table 8: MED data testing under VTWS

Document weight	Query weight	IAP	Top 23
SQRT*IGFFCONS	BNRY IDFB	59.55	6.83
SQRT*IDFSCONS	BNRY IDFP	59.05	6.80
SQRT*IGFFCONS	BNRY IDFB	59.01	6.83
LOGG*IGFSCONS	ATFA*ENPY	58.98	6.77
SQRT*IGFICONS	BNRY IDFB	58.91	6.87
ATFA*IGFS*CONS	SQRT*IDFB	58.70	6.73
LOGG*IGFICONS	BNRYIDFB	58.43	6.90
ATFA*IGFS*CONS	BNRYIDFP	58.31	6.87
ATFC*IGFS*CONS	LOGG*IDFP	58.33	6.63
LOGG*IGFI*CONS	BNRYIDFP	58.33	6.87
ATFA*IGFS*CONS	SQRT*ENPY	58.31	6.77
ATFC*IGFI*CONS	SQRT*ENPY	58.17	6.70
SQRT*IGFS*CONS	BNRYIDFB	57.87	6.83
SQRT*IGFI*CONS	LOGAIDFP	57.82	6.47
ATFC*IGFL*CONS	ATFC*ENPY	57.68	6.64
SQRT*IGFL*CONS	BNRYIDFB	57.67	6.87
LOGA IGSS CONS	LOGG*IDFB	57.67	6.67
LOGA NONE CONS	ATF1ENPY	53.41	6.63
LOGA NONE CONS	LOGAIDFP	53.29	6.20
LOGA ENPY CONS	LOGAIDFB	52.92	6.17
FREQ IBFP CONS	LOGAENPY	52.47	6.30
LOGA NONE PUQN	ATF1IDFB	52.44	6.17
ATF1 NONENONE	LOGAIDFB	51.84	6.00
FREQNONECOSN	BNRYIDFP	48.19	6.27

Table 9: CISI data testing under VTWS

Document weight	Query weight	LAP	Top 23
SQRT*IGFS*CONS	LOGA*IDFP	19.40	2.40
LOGG*IGFS*CONS	LOGG*IDFP	19.21	2.80
SQRT*IGFI*CONS	ATFC*ENPY	18.92	3.00
SQRT*IGFS*CONS	BNRYIDFP	18.90	2.74
SQRT*IGFL*CONS	LOGG*IDFB	18.78	3.14
LOGG*IGFS*CONS	SQRT*IDFB	18.65	2.95
SQRT*IGFS*CONS	SQRTTuENPY	18.58	3.05
SQRT*IGFF*CONS	BNRYIDFB	18.56	2.97
ATFA*IGFS*CONS	BNRYIDFB	18.50	2.94
LOGG*IIGFII CONS	ARFA*ENPY	18.42	3.03
AFCC*IGFS CONS	BNRYIDBP	18.38	2.94
LOGZ ENPY CONS	AQRTuENPY	18.28	3.09
LOGANONEPUQN	BNRYIDFB	18.20	2.89
LOGAENPY CONS	LOCAIDFB	18.12	2.91
LOGANOME CONS	LOCAENPY	18.03	2.89
LOGANOME CONS	LOCAIGFB	18.03	2.91
ATFC*IGFL*CONS	LOCAIDFB	17.90	2.87
ATFC*IGFI CONS	BNRYIDFB	17.64	2.86
FREQ IDFB CONS	BRYTIDFB	17.61	2.86
LOGA IGFFCSN	ATF1IDFB	15.22	2.86
ATFI NONENONE	ATR1ENPY	13.51	3.00
FREQNONE CONS	AVRYIDFP	13.40	2.40
FREQNONENONE	GREQNONE	13.40	2.00

$$G_i = \log\left(\frac{F_i}{n_i} + 1\right)$$

IGFL is simply a combination of the IDF and IGFI weights. We also observed that the IGFI weight was working well (Table 8-11).

The second new global weight is square root global frequency IDF (IGFS), given by:

$$G_i = \sqrt{\frac{F_i}{n_i}} - 0.9$$

Table 10: CARN data testing under VTWS

Document weight	Query weight	Lap	Top 23
SQRT*IGFL*COSN	LOGG*IDFB	43.06	3.02
SQRT*IGFI*COSN	BNRYIDFB	43.04	3.03
SQRT*IGFI*COSN	ATFC*ENPy	43.00	3.03
LOGG*IGFI*COSN	SQRT*ENTY	42.88	3.04
SQRT*IGFF*COSN	BNRYIDFB	42.70	3.02
LOGG*IGFF*COSN	ATFA*ENPY	42.67	3.00
SQRT*IGFS*COSN	BNRYIDFB	42.64	3.04
LOGG*IGFS*COSN	SQRT*IDFB	42.53	3.01
SQRT*IGFS*COSN	BNRYIDFP	42.50	3.02
ATFC*IGFI*COSN	LOGG*IDFP	42.34	3.00
ATFC*IGFI*COSN	LOGAIDFP	42.19	3.92
ATFA*IGFS*COSN	BNRYIDFB	41.14	2.94
LOGA*IGFF*COSN	SQRT*ENPY	41.03	2.94
ATFA*IGFS*COSN	ATF1ENPY	41.93	2.96
ATFC*IGFS*COSN	BNRYIDFB	41.90	2.90
LOGA NONE COSN	BNRYIDBP	41.79	2.93
LOGN NONE PUQN	LOGAIDFB	41.76	2.88
LOGA NONE COSN	LOGAIDFB	41.52	2.86
LOGA ENPY COSN	LOGAENPY	41.20	2.81
ATF1 NONE COSN	BNRYIDFP	39.95	2.80
FREQ IDFB PUQN	ATF1IDFB	39.92	2.79
FREQ NONE COSN	FREQNONE	38.52	2.78
FREQ NONE COSN	FREQNONE	35.71	2.78

Table 11: CACAM data testing under VTWS

Document weight	Query weight	IAP	Top 23
SQRT*IGFF COSN	BNRY IDFB	49.55	5.83
SQRT*IGFS CONS	BNRY IDFP	49.05	5.80
SQRT*IGFSCONS	BNRY IDFB	49.01	5.83
LOGG*IGFS*CONS	ATFA*ENPY	48.98	5.77
SQRT*IGFI CONS	BNRYIDFB	48.91	5.87
ATFA*IGFS*CONS	SQRT*IDFB	48.40	5.73
LOGG*IGFICONS	BNRY IDFB	48.33	5.90
ATFC* IGFS*CONS	LOGG*IDFP	48.30	5.63
LOGG*IGFICONS	BNRY IDFP	48.23	5.87
ATFA*IGFS*CONS	SQRT*ENPY	48.11	5.77
ATFC*IGFI*CONS	SQRT*ENPY	47.77	5.70
SQRT*IGFS*CONS	BNRY IDFB	47.57	5.83
SQRT*IGFI*COHS	LOGA IDFP	47.42	5.47
ATFC*IGFL*CONS	ATFC*ENPY	47.38	5.63
SQRT*IGFL*CONS	BNRY IDFB	47.07	5.87
LOGA IGGF CONS	LOGG*IDFB	47.01	5.67
LOGA NONE CONS	ATF1 ENPY	46.81	5.63
LOGA NONE CONS	LOGA IDFP	46.69	5.20
LOGA ENPY CONS	LOGA IDFB	46.32	5.17
FREQ IDFB CONS	LOGG ENPY	46.17	4.90
LOGN NONE PUQN	ATF1 IDFB	45.44	4.87
ATF1 NONE NONE	LOGA IDFB	44.84	4.70
FREQ NONE COSN	BNRY IDFP	43.91	4.67

Like IGFL, IGFS is a combination of formulas. We found that subtracting larger numbers from F_i/n_i improved performance but didn't subtract one because that could cause a global weight of zero for some terms.

The third new global weight is incremented global frequency IDF (IGFI) given by:

$$G_i = \frac{F_i}{n_i} + 1$$

For improving the global weights, we found to see that adding one to a formula significantly improves its

performances. So we thought it might carry over to the global weights. Since IGFS already performed good, we tried adding one to it and the result was IGFI.

RESULTS AND DISCUSSION

To evaluate the reliability of the consolidation of diversifying terms weighting approach, we have implemented the vector space model in C and tested the proposed method on several datasets which included the correct answer. For a given terms weighting method, we have computed the similarities between the documents and each query in the testing data collection and returned a list of ranked documents according to the order of their similarity scores.

Four sets of well known data (MED, CISI, CARN, CACAM) have been used to perform the whole experiments. We have computed two scores here; Interpolated Average Precision (IAP) and top twenty-three (from the highest to the lowest values). The diversifying new terms weighing formulas are denoted by an asterisk [*]. From the testing results (Table 8-11), we found that the consolidation of these diversifying new terms weighting methods offer the improvement over the fundamental, local and global methods.

The new weights work well combining with both the other new weights and with the fundamental weights. The combination of local and global weights makes different performances too. Because a particular local weight when combined with one particular global weight performs well but with different global weights perform poorly.

CONCLUSIONS

IR performance is always comparative and model selection is superlative. In the real world, designing an IR system means reformulating terms weighting and similarity measuring. Since last few years, retrieval performances were largely dependent on the upgrading conventional term-weighting schemes[8,9]. This concept does not deserve a good hypothetical recommendation for solving problems today. In the real world, many problems must be solved in the intelligent ways, from the testing results of various terms weighting schemes (Table 8-11), we found that the consolidation of the diversifying new terms weighting methods increased the retrieval performances dramatically. We believe that our proposed methods could be used as a complement for others that work better on high precision tasks. It is still important because if a user can understand how to intensify the selective model for data retrieval purposes, a great deal of time can be saved.

REFERENCES

1. Singhal, A., 2001. Modern Information Retrieval: A Brief Overview. Google, Inc.
2. Djoerd, H. and S. Robertson, 2001. Relevance Feedback for Best Match Term Weighting Algorithms in Information Retrieval. Microsoft Research Group, Cambridge, UK.
3. Jung, Y., H. Park and D. Dul, 2000. An effective term weighting scheme for IR. CST Report, University of Minnesota.
4. Eamonn, K.C., S. Mehrotra and M. Pazzani, 2002. Locally adaptive dimensionality reduction for indexing large time series databases. ACM., 27: 188-228.
5. Moffat and Zobel, 2002. Information Retrieval Systems from Large Document Collections. TREC-3, NIST, Gaithersburg, MD, pp: 500-525.
6. Shankar, S. and G. Karypic, 2000. Weight adjustment schemes for centroid based classifier. CST Report: TR00-035, Univresity of Minnesota, Minnesota.
7. Kowalski and J. Gerald, 1997. Information Retrieval Systems Theory and Implementation. Kluer Academic Publishers, 1: 296.
8. Lee, D. and H. Chung, 1997. Document Ranking and the Vector Space Model. HKUST, Hong Kong, China.
9. Sanderson, M. and C.J. Van Rijsbergen, 1999. The impact of IR effectiveness of skewed frequency distributions. ACM, pp: 440-465.
10. Chisholm, E. and T.G. Kolda, 1999. New term weighting formulas for the vector space method in information retrieval. Oak Ridge National Laboratory, Oak Ridge, TN 37831-6367.
11. Greogry, B., 1992. Information Space Gets Normal. Frakes and B.Vates, pp: 372-375.
12. Buckley, C. and G. Salton, 1988. Term weighting approaches in automatic text retrieval. J. Inform. Proc. Manage., 24: 513- 523.
13. Croft, W.B. and D. Harper, 1979. Using probabilistic models of document retrieval without relevance information. J. Documentation, 35: 285- 295.
14. Warren, R., G. Jay and M. Ponte, 2000. The maximum entropy approach and probabilistic IR Models. ACM, pp: 246-287.
15. Pavlov, D., H. Mannila and P. Smyth, 2000. Maximum entropy techniques for analyzing large transaction datasets. Project Report. University of California, Irvine, CA 92697-3425.
16. Sinhale, A., C. Buckley, M. Mitra and G. Salton, 1995. Pivoted document length normalization. Technical Report: TR 95-1560, Cornell Univrersity, Itahaca, NY, USA.