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\cite{3}. 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\cite{4}, 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\cite{5,6}. People use different types of IR system in the forms of Internet search engine to retrieve the required informations\cite{7,8}. 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\cite{9,10}. 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\cite{11,12}. It is useful to give a geometric interpretation to the vector space notion of similarity by considering the dot product equation.

\[ \cos \theta = \frac{d \cdot q}{\|d\| \|q\|} \]

In a vector space model, where, d is a document vector, q is a query vector and \( \theta \) is the angle between them, if d and q are normalized so that the magnitudes are one, preceding equation then reduces to \( \theta = d \cdot q \), 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

<table>
<thead>
<tr>
<th>Term ID</th>
<th>Word</th>
<th>Weight</th>
<th>Term ID</th>
<th>Word</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>900</td>
<td>Colestorol</td>
<td>0.09</td>
<td>1420</td>
<td>Levels</td>
<td>0.07</td>
</tr>
<tr>
<td>1120</td>
<td>Blood</td>
<td>0.36</td>
<td>1520</td>
<td>High</td>
<td>0.07</td>
</tr>
<tr>
<td>1420</td>
<td>Levels</td>
<td>0.18</td>
<td>1790</td>
<td>Fatty</td>
<td>0.07</td>
</tr>
<tr>
<td>1500</td>
<td>Low</td>
<td>0.09</td>
<td>1850</td>
<td>Normal</td>
<td>0.44</td>
</tr>
<tr>
<td>1520</td>
<td>Low</td>
<td>0.09</td>
<td>1850</td>
<td>Normal</td>
<td>0.44</td>
</tr>
<tr>
<td>1680</td>
<td>Fatal</td>
<td>0.18</td>
<td>1890</td>
<td>Glucose</td>
<td>0.07</td>
</tr>
<tr>
<td>1890</td>
<td>Glucose</td>
<td>0.09</td>
<td>2021</td>
<td>Free</td>
<td>0.29</td>
</tr>
<tr>
<td>2021</td>
<td>Free</td>
<td>0.09</td>
<td>2450</td>
<td>Acid</td>
<td>0.29</td>
</tr>
<tr>
<td>2390</td>
<td>Infant</td>
<td>0.19</td>
<td>2519</td>
<td>Toxin</td>
<td>0.29</td>
</tr>
<tr>
<td>2530</td>
<td>Lactoz</td>
<td>0.07</td>
<td>3033</td>
<td>Primary</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 2: Query from the MED test collection

<table>
<thead>
<tr>
<th>Term ID</th>
<th>Word</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1420</td>
<td>Levels</td>
<td>0.30</td>
</tr>
<tr>
<td>1500</td>
<td>Low</td>
<td>0.30</td>
</tr>
<tr>
<td>1520</td>
<td>High</td>
<td>0.30</td>
</tr>
<tr>
<td>1790</td>
<td>Fatty</td>
<td>0.25</td>
</tr>
<tr>
<td>1890</td>
<td>Glucose</td>
<td>0.30</td>
</tr>
<tr>
<td>2021</td>
<td>Free</td>
<td>0.30</td>
</tr>
<tr>
<td>2530</td>
<td>Lactoz</td>
<td>0.36</td>
</tr>
<tr>
<td>2591</td>
<td>Toxin</td>
<td>0.30</td>
</tr>
<tr>
<td>3031</td>
<td>Iron</td>
<td>0.24</td>
</tr>
</tbody>
</table>

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:

\[
(0.18 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) + (0.09 \times 0.30) = 0.1602
\]

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

\[
(0.07 \times 0.30) + (0.07 \times 0.30) + (0.07 \times 0.25) + (0.07 \times 0.30) + (0.29 \times 0.30) + (0.29 \times 0.30) + (0.29 \times 0.30) + (0.29 \times 0.30) = 0.2545
\]

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\(^5\). 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 methods\[^{10}\] local, global and length normalization are used for practical purposes. The term weighting is given by:

\[
L_q = G_j N_i;
\]

Where, \(L_q\) is the local weight for term \(i\) in document \(j\), \(G_j\) is the global weight for term \(i\) and \(N_i\) 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:

\[
L_i = \begin{cases} 
1, & \text{if } f_{ij} > 0 \\
0, & \text{if } f_{ij} = 0; \text{ and}
\end{cases}
\]

\[
L_q = f_i; \text{ Where, } f_i \text{ is the frequency of term } i \text{ in document } j.\]

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

<table>
<thead>
<tr>
<th>Formula</th>
<th>Name</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 if $f_i &gt; 0$</td>
<td>$f_i$</td>
<td>Binary</td>
</tr>
<tr>
<td>0 if $f_i = 0$</td>
<td>$f_i$</td>
<td>within-document frequency</td>
</tr>
<tr>
<td>$1 + \log f_i$ if $f_i &gt; 0$</td>
<td>$f_i$</td>
<td>Log</td>
</tr>
<tr>
<td>0 if $f_i = 0$</td>
<td>$f_i$</td>
<td>LOGA</td>
</tr>
<tr>
<td>$1 + \log f_i$ if $f_i &gt; 0$ or $f_i = 0$</td>
<td>$f_i$</td>
<td>Normalizing log</td>
</tr>
<tr>
<td>$0.5 + 0.5 \left( \frac{f_i}{x_j} \right)$</td>
<td>$f_i$</td>
<td>Augmented</td>
</tr>
<tr>
<td>0 if $f_i &gt; 0$</td>
<td>$f_i$</td>
<td>Normalized term frequency</td>
</tr>
<tr>
<td>0 if $f_i = 0$</td>
<td>$f_i$</td>
<td>ATF1</td>
</tr>
</tbody>
</table>

Table 5: Global terms weighting formulas

<table>
<thead>
<tr>
<th>Formula</th>
<th>Meaning (Name)</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log \left( \frac{N}{n_i} \right)$</td>
<td>Inverse document frequency</td>
<td>IDF1</td>
</tr>
<tr>
<td>$\log \left( \frac{N - n_i}{n_i} \right)$</td>
<td>Probabilistic inverse</td>
<td>IDF1</td>
</tr>
<tr>
<td>$1 + \frac{\sum_{i=1}^{n} \log f_i}{\log N}$</td>
<td>Entropy</td>
<td>ENPY</td>
</tr>
<tr>
<td>$\frac{F_i}{n_i}$</td>
<td>Global frequency</td>
<td>IGF</td>
</tr>
</tbody>
</table>

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

$$L_q = \begin{cases} 
1 + \log f_i \quad \text{if } f_i > 0 \\
0, \quad \text{if } f_i = 0 
\end{cases}$$

$$L_q = \begin{cases} 
1 + \log f_i \quad \text{if } f_i > 0 \\
1 + \log a_i \quad \text{if } f_i = 0 
\end{cases}$$

Where, $a_i$ is the average frequency of the terms that appear in document $j$. Because LOGN is normalized by the $(1 + \log a_i)$ 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)

$$L_q = \begin{cases} 
.5 + .5 \left( \frac{f_i}{x_j} \right) \quad \text{if } f_i > 0 \quad \text{and} \\
0, \quad \text{if } f_i = 0 
\end{cases}$$

Where, $x_i$ 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 $L_q$ 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, IDF2 and IDF3 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. IDF2 is the logarithm of the inverse of the probability that term $i$ appears in a random document. IDF3 is the logarithm of the inverse of the odds that term $i$ appears in random document. IDF2 and IDF3 are similar that they both award high weight for terms appearing in few documents and low weight for terms appearing in many documents in the collection; however, they differ because IDF3 actually awards negative weight for terms appearing in more than half of the documents in the collection and the lowest weight of IDF2 is one. In addition, we have used the Entropy weight (ENPY)

$$G_i = 1 + \frac{n_i \log \frac{f_i}{\log N}}{\sum_{j=1}^{n} \log N},$$

Where, $F_i$ is the frequency of term $i$ throughout the entire collection. 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

$$(IGF)$$

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

<table>
<thead>
<tr>
<th>Formula</th>
<th>Name</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.2 + 0.8 \frac{f_i}{x_i}$ if $f_i &gt; 0$ or if $x_i = 0$</td>
<td>Changed-coefficient ATPF</td>
<td>ATFC</td>
</tr>
<tr>
<td>$0.9 + 0.1 \frac{f_i}{x_i}$ if $f_i &gt; 0$ or if $x_i = 0$</td>
<td>Augment</td>
<td>ATFA</td>
</tr>
<tr>
<td>$0.2 + 0.8 \log(f_i + 1)$ if $f_i &gt; 0$ or if $f_i = 0$</td>
<td>Augmented log frequency</td>
<td>LOGG</td>
</tr>
<tr>
<td>$\frac{f_i}{\sqrt{x_i} - 0.5} + 1$ if $f_i &gt; 0$ or if $x_i = 0$</td>
<td>Square root</td>
<td>SQRT</td>
</tr>
</tbody>
</table>

Table 7: Diversify global terms weighted formulas

<table>
<thead>
<tr>
<th>Formula</th>
<th>Name</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\log \left( \frac{f_i}{x_i} + 1 \right)$</td>
<td>Log-global frequency IDF</td>
<td>IGLF</td>
</tr>
<tr>
<td>$\frac{f_i}{x_i} + 1$</td>
<td>Incremental global frequency IDF</td>
<td>IGGF</td>
</tr>
<tr>
<td>$\frac{f_i}{\sqrt{x_i} - 0.9}$</td>
<td>Square root global frequency IDF</td>
<td>IGFS</td>
</tr>
</tbody>
</table>

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

$$N_i = \frac{1}{\sqrt{\sum_{i=1}^{m} (G_i L_i)^2}}.$$

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:

$$N_i = \frac{1}{(1 - \text{slope}) + \text{slope} L_i}.$$

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 diversifying weighting formulas are changed coefficient ATFI (ATFC) and augmented average term frequency (ATFA), given, respectively by:

$$L_i = \begin{cases} 
0.2 + 0.8 \frac{f_i}{x_i} & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$

$$L_i = \begin{cases} 
0.9 + 0.1 \frac{f_i}{x_i} & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$

ATFC is developed using a general version of ATFI

$$L_i = \begin{cases} 
K + (1 - K) \frac{f_i}{x_i} & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$

Changed-coefficient ATFI 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 ATFI but it is normalized differently. ATFI 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,

$$L_i = \begin{cases} 
0.2 + 0.8 \log(f_i + 1) & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$

We simply modified ($f_i/x_i$) to log($f_i+1$) because log seems to be a better local weight than within-document frequency. Note that now $L_i$ can be greater than one. Another modify new local weight is square root (SQRT), given by;

$$L_i = \begin{cases} 
\sqrt{f_i - 0.5} + 1 & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$

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 $\sqrt{f_i}$ would have a similar shape. As $f_i$ 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:

$$L_i = \begin{cases} 
0.2 + 0.8 \frac{f_i}{x_i} & \text{if } f_i > 0 \text{ or } x_i = 0; \\
0 & \text{else.}
\end{cases}$$
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:

\[ G_i = \sqrt[\frac{F_i}{n_i} - 0.9] \]

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 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, CISTI, 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
t. 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