# Fourier Domain Scoring: A Novel Document Ranking Method

Laurence A.F. Park, *Member*, *IEEE*, Kotagiri Ramamohanarao, and Marimuthu Palaniswami, *Senior Member*, *IEEE*

**Abstract**—Current document retrieval methods use a vector space similarity measure to give scores of relevance to documents when related to a specific query. The central problem with these methods is that they neglect any spatial information within the documents in question. We present a new method, called Fourier Domain Scoring (FDS), which takes advantage of this spatial information, via the Fourier transform, to give a more accurate ordering of relevance to a document set. We show that FDS gives an improvement in precision over the vector space similarity measures for the common case of Web like queries, and it gives similar results to the vector space measures for longer queries.

**Index Terms**—Fourier domain scoring, information retrieval, search engine, vector space similarity measure, document ranking, Fourier transform, term signal.

## 1 Introduction

The birth of the World Wide Web just over a decade ago created many new areas of research and has brought back to life some that had been stagnant. The Web (which can be considered a huge database of documents) has become so popular that its content has grown to more than a billion documents. With so much information, the World Wide Web has the need for some organization. Today, this organization comes in the form of a Web search engine. The job of the search engine is to remove the chaos and clutter of the Web by allowing users to supply key terms so that the search engine can find information relevant to these key terms on the Web. Due to the importance of search engines, the interest in the field of information retrieval has risen.

There have been many methods for seeking out information on the Web. Many new techniques involve utilizing the HTML tags found on Web pages to obtain a higher understanding of the page content [1], [2], [3], [4], [5]. There have also been many page creators who have abused searching methods to obtain higher rankings in search engine results. A few search engines have branched out to analyzing pages in the form of text, PDF, Postscript, and a few Microsoft file formats as well as the traditional HTML.

To identify the topic of a document, the text within must be analyzed. This is why the text analysis method used in any search engine is critical. By obtaining a clear understanding of a page, we are able to find more relevant documents and make it harder for people to cheat the system.

- *L.A.F. Park is with the ARC Special Research Centre for Ultra-Broadband Information Networks, Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia 3010. E-mail: lapark@ee.mu.oz.au.*
- *K. Ramamohanarao is with the ARC Special Research Centre for Ultra-Broadband Information Networks, Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, Victoria, Australia 3010. E-mail: rao@cs.mu.oz.au.*
- *M. Palaniswami is with the Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria, Australia 3010. E-mail: swami@ee.mu.oz.au.*

*Manuscript received 1 Mar. 2002; revised 1 Oct. 2002; accepted 5 May 2003. For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 115992.*

Variants of the vector space model [6] have been the dominant methods used to analyze the text in information retrieval systems for many years. The concept behind the vector space model is to convert each document into a vector, so they can be easily compared to other document vectors (to find similarity amongst documents) and they can also be compared to query vectors (to find relevance to the query). The document vectors exist in a space where each dimension is one unique term from the document set. The size of the document vector along each dimension is a weighted count of that term in the document.

Many information retrieval systems employ the use of a vector space model to classify text. Some examples of current work which use the vector space model are SMART by Buckley et al. [7], [8], the Okapi Basic Search System by Robertson et al. [9], IRIS by Yang et al. [10], and INQUERY by Allan et al. [11]. Work by Zobel and Moffat [12] compared 720 combinations of the vector space weighting method on different data sets in the quest to find the best method. There was no consistent winner, but there were methods which appeared high in most experiments.

The problem with these techniques is that any spatial information contained in the documents is lost. Once the documents are converted into document vectors, the number of times each term appears is represented in the vector, but the positions of the terms (the flow of the document) is ignored.

We present a method entitled Fourier Domain Scoring (FDS), which can retain the document spatial information and use it to effectively rank documents. The difference between FDS and other vector space similarity measures is that, rather than storing only the count of a frequency term per document, FDS stores a term signal. The term signal shows how the term is spread throughout the document. If the spectrum of the query term signals are compared, we are able to observe which documents have a high occurrence of the query terms and which documents have the query terms appearing together. This information is obtained by comparing the magnitude and phase of the spectrum across different term signals, respectively.

1. Create index : For each term $t \in$ document set $D$
   - *A:* Create query term signals $\{\tilde{f}_{0,t}, \tilde{f}_{1,t}, \ldots, \tilde{f}_{D,t}\}$
   - *B:* Group term signals to produce inverted index
2. Query index : For each term $t \in T$ and document $d$ in collection
   - *C:* Weight signals $\tilde{\omega}_{d,t} = \Omega_d(\tilde{f}_{d,t})$
   - *D:* Transform signals into spectrums $\tilde{\nu}_{d,t} = \mathrm{DFT}(\tilde{\omega}_{d,t})$
   - *E:* For each spectral component $\nu_{d,t,b} \in \tilde{\nu}_{d,t}$
     - (a) Calculate component magnitudes $H^*_{d,b}$
     - (b) Calculate component phase precision $\Phi^*_{d,b}$
   - *F:* Combine component scores $s_{d,b} = \Phi^*_{d,b} H^*_{d,b}$ to obtain document score $s_d$

Fig. 1. Fourier Domain Scoring (FDS) algorithm.

FDS is a new text classification method which can be used in information retrieval systems by replacing the existing vector space similarity measures. Just as in information retrieval systems which use vector space similarity measures, other enhancements can be added on top of FDS to fully utilize the document environment and improve the results of the search (as in examining hypertext links on the Web, using a thesaurus, etc.).

Cases of systems which use the vector space similarity measures which could easily be replaced by FDS can be found everywhere. Some examples are as follows.

Google [13] records the position of every term in the document to give a higher ranking to document which contain query terms closer together. Systems such as PADRE [14] and Okapi [9] also use this style of proximity searching. To calculate the distances between every term would require a lot of calculations at query time, or a large space to store all of the previously calculated distances. FDS could easily be implemented in the Google search engine to store the precalculated phases of each term.

Ebert et al. [15] describe a method of visualizing clusters in documents. This system takes sequences of characters from the text to create vectors for the user to explore. It would be very interesting to replace these $n$-gram vectors with the Fourier transform of each character sequence or term sequence. This might provide a more compact representation and give better results.

The Latent Semantic Indexing method [16] uses singular value decomposition (SVD) to reduce the dimension of the index used by conventional search engines. By reducing the dimension, we are trying to remove noise generated by our use of language and trying to find the true semantic structure of the document. This method uses document vectors containing the count of each term as the elements, therefore, any spatial information is lost. If the document vectors were replaced by a document matrix generated by FDS, then SVD could easily be performed on each of the sets of spatial bins. This would produce a structural latent semantic index.

This paper is organized as follows: Section 2 will give an introduction to the Fourier Domain Scoring method and outline the basic steps needed to perform such a task on large document sets. Section 3 will show how the vector space methods are a special case of FDS. Section 4 will look into the new data extracted from the document and how it copes with large queries. Section 5 explains, in detail, the experiments performed using the TREC data set with short queries and compares the results with existing ranking methods.

## 2 Fourier Domain Scoring

Vector space similarity measures will only give a score based on the count of each term in the document. FDS tries to capture the location of the term through the document. Terms can then be compared to find if they are relevant to the document.

FDS compares the positions of the terms by comparing their phase and adjusts according to the appearance of the term by observing the magnitude. The basic structure of the FDS procedure is given in Fig. 1. Some of our initial work on this topic can be found in [17]. We discuss each of these steps in brief as it is essential to understand how FDS works before we can examine any of its properties.

### 2.1 Collect Terms into Spatial Bins

Rather than mapping a document to a vector that contains the count of each term, FDS maps each document into a set of term vectors to build its index. These term vectors show the position of the term throughout the document, where the position of the element represents the position of the term in the document. To reduce the term vector size (and, hence, the calculations needed), the terms should be grouped into bins. If the number of bins is set to $B$, a document containing $W$ terms would have $W/B$ terms in each bin. Therefore, the first element of the term vector would contain the number of times the term appeared in the first $W/B$ terms. The second element would be the number of times the term appeared in the second $W/B$ terms, and so on. For example, the top half of Fig. 2 gives us the positions of the terms "travel" and "wales" throughout a document (signified by the vertical bars). If we choose $B = 8$, we obtain the term signal

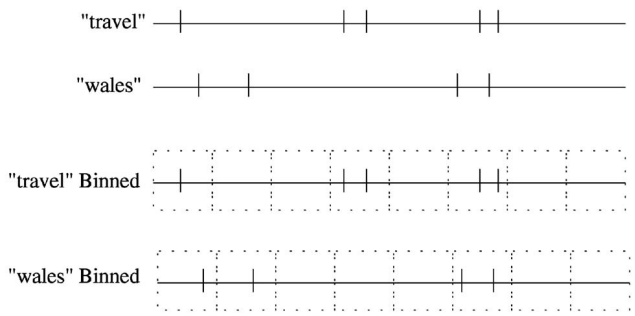



Fig. 2. A visual example of how the term signals are obtained. The top two lines, labeled "travel" and "wales," show the positions of the terms travel and wales in some document (the position is signified by the vertical stroke through the line). The bottom half shows the binned positions of the terms.

[1 0 0 2 0 2 0 0] for the term "travel" in that document and [1 1 0 0 0 2 0 0] for the term "wales" in that document (as shown in the bottom half of Fig. 2).

### 2.2 Create Inverted Index

Just as in any vector space model, an inverted index can be created to enable quick retrieval of term vectors. In this case, a little more information needs to be stored due to the documents being split into bins. The terms in each document were represented as:

$$< n >< b_1, f_1 >< b_2, f_2 > \ldots < b_n, f_n >, \quad (1)$$

where $n$ is the number of nonzero bins, $b_a$ is the bin number, and $f_a$ is the count of the term in bin $b_a$.

### 2.3 Weighting

The cosine similarity measure can be improved considerably by adding weighting to the document vectors before the score is calculated [18]. A few of the better performing schemes are BD-ACI-BCA, BI-ACI-BCA, and AB-AFD-BAA in Zobel and Moffat [12], and the Lnu.ltu method contained in SMART [8]. Both BD-ACI-BCA and Lnu.ltu are good all round weighting schemes, while AB-AFD-BAA is better for short queries and BI-ACI-BCA is more precise when using long queries.

Each of these schemes contain document weighting and query weighting. For example, BD-ACI-BCA refers to the following weights being used:

$$\begin{aligned} \Omega_d(f_{d,t}) &= \frac{1 + \log_e f_{d,t}}{(1-s) + s \cdot W_d / av_{d \in D} W_d} \\ \Omega_q(f_{q,t}) &= (1 + \log_e f_{q,t}) \log_e \left(1 + \frac{f^m}{f_t}\right), \end{aligned} \quad (2)$$

where $\Omega_d, \Omega_q$ are the document and query weights, respectively, $f_{d,t}$, $f_{q,t}$ are the count of term $t$ in document $d$ and query $q$, respectively, $s$ is the slope factor (set to 0.7), $W_d$, $av_{d \in D} W_d$ are the document vector norm and average document vector norm, respectively, $f_t$ is the number of documents having term $t$, and $f^m$ is the largest $f_t$. We can see that BD-ACI-BCA uses pivoted document normalization.

The document normalization is used to reduce the effect of repetition of a term in a document and to negate any effect the size of the document has on a query. The query normalization is used to reduce the effect of repetition of terms in the query and reduces the effect common terms have on the document score. FDS also requires these properties, therefore, preweighting (applying weights before performing the Fourier transform) can be performed to increase the accuracy of the relevance scores. Since we will be comparing FDS to the vector space methods mentioned, we will also use these for weighting. To use the weights on the term signals, we simply apply the same weighting scheme to the bin count, rather than the document count. For example, if we were using the BD-ACI-BCA weighting scheme with FDS, the weighted bin values would become:

$$\omega_{d,t,b} = \Omega_d(f_{d,t,b}) = \frac{1 + \log_e f_{d,t,b}}{(1-s) + s \cdot W_d / av_{d \in D} W_d}, \quad (3)$$

where $f_{d,t,b}$ is the count of term $t$ in bin $b$ of document $d$. The query weighting ($\Omega_d(f_{d,t})$, which is applied later) remains the same as in the vector space methods.

### 2.4 Perform Fourier Transform

The Fourier transform has been used in many areas of electrical engineering. It is designed to change the basis of a signal (or function) to linearly independent sinusoidal waves. The discrete form of the transform (DFT) is of the form:

$$\nu_{d,t,\beta} = \sum_{b=0}^{B-1} \omega_{d,t,b} \exp\left(\frac{-i 2\pi \beta b}{B}\right). \quad (4)$$

In our case, the signal will be the weighted term signal consisting of elements $\omega_{d,t,b}$, where $b \in \{0, 1, \ldots, B-1\}$. Since each $\nu_{d,t,b}$ is the projection of the term signal $\tilde{\omega}_{d,t}$ onto a sinusoidal wave of frequency $\beta$, the signal $\tilde{\nu}_{d,t}$ is the spectrum of the given term signal. The spectral component number $\beta$ is an element of the set $\{0, 1, \ldots, B-1\}$.

Therefore, the Fourier transform maps a signal from the time or spatial domain to the frequency domain. Once the spectrum of a signal can be seen, we are able to identify the major frequency components which give the signal its shape. The Fourier transform produces the following mapping:

$$\{\omega_{d,t,b}\} \rightarrow \{\nu_{d,t,b}\} = \{H_{d,t,b} \exp(i \phi_{d,t,b})\},$$

where $\omega_{d,t,b}$ is the weight of term $t$ in bin $b$ of document $d$, $\nu_{d,t,b}$ is the $b$th frequency component of term $t$ in document $d$, $H_{d,t,b}$ and $\phi_{d,t,b}$ are the magnitude and phase of frequency component $\nu_{d,t,b}$, respectively (and are therefore real values), and $i$ is $\sqrt{-1}$.

The Nyquist-Shannon sampling theorem [19] states that the highest frequency component to be found in a real signal is equal to half of the sampling rate. This implies that, if we choose $B$ bins for the term signal, we will only need to examine frequency components 0 to $\frac{B}{2}$ to analyze all of the information found in the spatial term signal.

By performing the Fourier transform on the spatial term bins, we are able to see how the term flows through the document. Each frequency component contains magnitude and phase information which can be interpreted as the effect and shift of the component, respectively. The effect gives us an idea of the shape of the term signal. If a lower frequency component magnitude is large with respect to the other components, then the term should appear clustered in a few places in the document. If a higher frequency component magnitude is large with respect to the other components, then the term clusters would be scattered through the document. The shift represents the position of the term throughout the document and is measured in radians. The shift is useful when comparing term signals. If two signals have the same phase, then they are said to be "in phase," meaning they appear together. But, if two signals have opposite phase (different by $\pi$ radians), then they are "out of phase." Therefore, if two or more term signals are in phase, this implies that they appear together throughout the document. If two of more terms are not in phase, it implies that they appear in the document, but not together most of the time.

For example, if we consider the term signals for the terms "travel" and "wales" at the top of Fig. 3, we can see that they appear together in bins 0 and 5. If we perform the DFT on each signal, we can examine the magnitude and phase of the spectral components. If a component $c$ from term 1 has a similar phase to component $c$ from term 2, these two components are considered in phase, which implies that the terms 1 and 2 appear together in that region

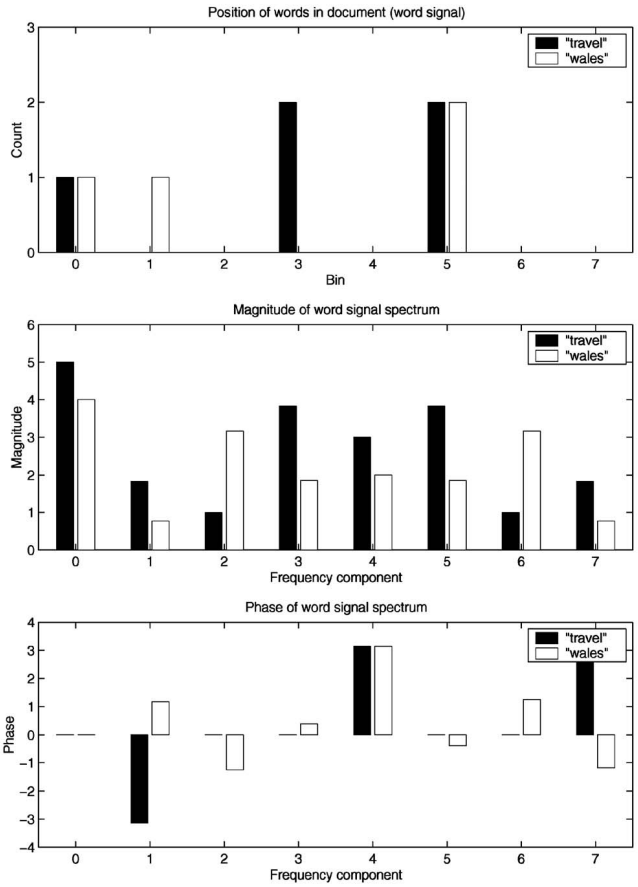



Fig. 3. An example of before and after the Fourier transform. We can see the binned positions of term 1 and term 2 from a certain document in the top plot. The bottom two plots show the information obtained after computing the Fourier transform on each of the binned term signals. Notice the reflection of the spectral information about component 4 (as stated in the Nyquist-Shannon sampling theorem).

of the spectrum. This is the case for components 0, 3, and 4 when comparing the terms "travel" and "wales." If the magnitude for those components are high (relative to the other components), it implies that these are major frequency components of the term signal. This is also the case with components 0, 3, and 4 when comparing the terms "travel" and "wales." Therefore, this document can be considered relevant to the query "travel" and "wales." An example of the effect of the Fourier transform on a term signal can be seen in Fig. 3.

## 2.5 Combining Term Spectra

Once we have performed the Fourier transform on the $B$ length vectors, we have a set of $\frac{B}{2}+1$ independent complex numbers for each query term, known as the term spectra. From intuition, a relevant document should have large magnitudes and the corresponding phases from each term should be similar (in phase), so we should deal with the magnitude and phase separately. We will combine the spectra to obtain the magnitude $(H_{d,b})$ and phase precision $(\Phi_{d,b})$ for each component $b$ in document $d$.

### 2.5.1 Magnitude

To calculate the effect of the query terms on the document, we can either treat the spectral components as complex values or just examine the magnitude of each.

The first method treats each set of frequency components as though it is a separate index. Each spectral component is multiplied by the weight of the corresponding query term. The resulting complex values are then added and the magnitude is taken. If the complex values are in phase for a particular component, the resulting score component $(s_{d,b})$ will be large. If the spectral components are out of phase, they will cancel each other out and result in a small score vector. The magnitude $(H^v_{d,b})$ of each component $b$ in each document $d$ is:

$$\text{Sum vectors} := H^v_{d,b} = \left| \sum_{t \in T} \nu_{d,t,b} \cdot \Omega_q(f_{q,t}) \right|, \tag{5}$$

where $\nu_{d,t,b}$ is the complex value of frequency component $b$ of term $t$ in document $d$ and $\Omega_q(f_{q,t})$ is the query term $t$ weight.

The motivation behind this idea is to give a high score to the documents which contain all similar term spectrums for all terms in the query. A higher score will also be given to the documents which contain more occurrences of the query terms (which would lead to higher magnitude of the spectral components).

If we take into account only the magnitude of the complex spectral components (and deal with the phase later), we can add the magnitudes to obtain the occurrence of the term through the document:

$$\text{Sum magnitudes} := H^m_{d,b} = \sum_{t \in T} \left| \nu_{d,t,b} \cdot \Omega_q(f_{q,t}) \right|. \tag{6}$$

### 2.5.2 Phase Precision

If we examine the phase information of the term signal spectrum, we are also examining the position of the term in the spatial domain. We want terms with similar position, therefore, we want to find the spectral components which have similar phase. This can be performed by treating each spectral component as a unit vector with its own phase and finding the average phase which we call phase precision. When finding the phase precision, we are left with the choice of what to do with spectral components of zero magnitude since the phase of these components mean nothing. The choices we have made are to either average over the nonzero components and to average over all components.

*Nonzero phase precision* is given by the following equation:

$$\text{Nonzero phase precision} := \Phi^n_{d,b} = \left| \frac{\sum_{t \in T: H_{d,t,b} \neq 0} \phi_{d,t,b}}{\#(\hat{T}_{d,b})} \right|, \tag{7}$$

where

$$\nu_{d,t,b} = H_{d,t,b} \exp(i\theta_{d,t,b}) \tag{8}$$

$$\phi_{d,t,b} = \frac{\nu_{d,t,b}}{|\nu_{d,t,b}|} = \exp(i\theta_{d,t,b}) \tag{9}$$

and $\theta_{d,t,b}$ is the phase of the $b$th frequency component of the $t$th query term in the $d$th document $\phi_{d,t,b}$ is the unit phase of the $b$th component of the $t$th query term in the $d$th document, $\#(x)$ is a function which gives the cardinality of the set $x$, and $\hat{T}_{d,b}$ is the set of query terms which do not have zero magnitude for frequency component $b$ in document $d$.

*Zero phase precision* is again similar to nonzero phase precision in the way that it only includes the frequency

TABLE 1
Calculation of the Different Magnitude and Phase Precision Values of the Terms *Mariquita*, *Travels*, and *Wales*

| Term | $\Omega_q(f_{q,t})$ | Frequency Component (magnitude, phase) | | | | |
|---|---|---|---|---|---|---|
| | | 0 | 1 | 2 | 3 | 4 |
| mariquita | 1.8 | (3.0, 0) | (1.0, −0.7) | (3.0, −1.5) | (1.0, −2.3) | (3.0, 3.1) |
| travels | 1.0 | (1, 0) | (1, −2.3) | (1, 1.5) | (1, −0.7) | (1, 3.1) |
| wales | 1.1 | (2.0, 0) | (1.4, 3.1) | (0, 0) | (1.4, 0) | (2.0, 3.1) |
| $H^v_{d,b}$ | | 8.6 | 2.2 | 4.4 | 2.2 | 8.6 |
| $H^m_{d,b}$ | | 8.6 | 4.3 | 6.4 | 4.3 | 8.6 |
| $\Phi^n_{d,b}$ | | 1.0 | 0.5 | 0 | 0.5 | 1.0 |
| $\Phi^z_{d,b}$ | | 1.0 | 0.5 | 0 | 0.5 | 1.0 |
| $\Phi^1_{d,b}$ | | 1 | 1 | 1 | 1 | 1 |

components with nonzero magnitude, but it averages over the total number of query terms. This total averaging is performed so that if any query terms do not have a frequency component $b$, then the score will be reduced.

$$\text{Zero phase precision} := \Phi^z_{d,b} = \left| \frac{\sum_{t \in T: H_{d,t,b} \neq 0} \phi_{d,t,b}}{\#(T)} \right|, \tag{10}$$

where $T$ is the set of query terms.

*No phase precision* ignores any phase information. This is best used when the phase has already been taken into account when creating the magnitude vector.

$$\text{No phase precision} := \Phi^1_{d,b} = 1. \tag{11}$$

Once the magnitude and phase precision have been obtained, we are able to combine them to create the document score vector. Each component of the score vector is calculated by multiplying the corresponding magnitude and phase precision values.

$$s_{d,b} = H^*_{d,b} \Phi^*_{d,b}, \tag{12}$$

where $H^*_{d,b}$ and $\Phi^*_{d,b}$ represent the selected magnitude and phase precision methods selected, respectively.

To assist in understanding the term spectrum combination process, we have provided a sample set of term signal spectrums and calculations of each of the magnitude and phase precision methods in Table 1.

## 2.6 Combining Spectral Components

Once the term spectrums have been combined into score vectors, we are left with a vector with $B/2 + 1$ elements, containing how relevant each query term was to that frequency component. This section will explain a few methods to help us obtain a single score from this vector and allow us to rank a set of documents in terms of relevance to the query terms. Methods which were experimented with were:

1. Sum all components.
2. Sum largest score vector elements.
3. Sum largest phase precision components.
4. Sum largest magnitude components.
5. Sum threshold phase precision components.

### 2.6.1 Sum All Components

Since each element of the score vector represents the content of the query terms in that specific frequency component, a document which contains high values in each element should be more relevant than one that contains lower values. Therefore, the most obvious operation to perform is summation.

$$S_d = \sum_{b=1}^{B/2+1} s_{d,b}. \tag{13}$$

### 2.6.2 Sum Largest Score Vector Elements

It could also be stated that if there are high values contained in any of the elements of the score vector, then the the document should be considered relevant the the query terms. To implement this, only the components with the two greatest values were added to create the score.

$$S_d = s_{d,b_1} + s_{d,b_2} \tag{14}$$
$$s_{d,b_1}, s_{d,b_2} \geq \max_{\forall b \neq b_1, b_2} (s_{d,b}).$$

The larger of the two components will be the zeroth component (DC component). The DC component magnitude of a term signal spectrum will be equal to the sum of all bins of the term signal (and, hence, the value used in vector space measures). This largest component is a real value and, therefore, has zero phase (leading to a precision of 1 if all terms are present).

### 2.6.3 Sum Largest Phase Precision Components

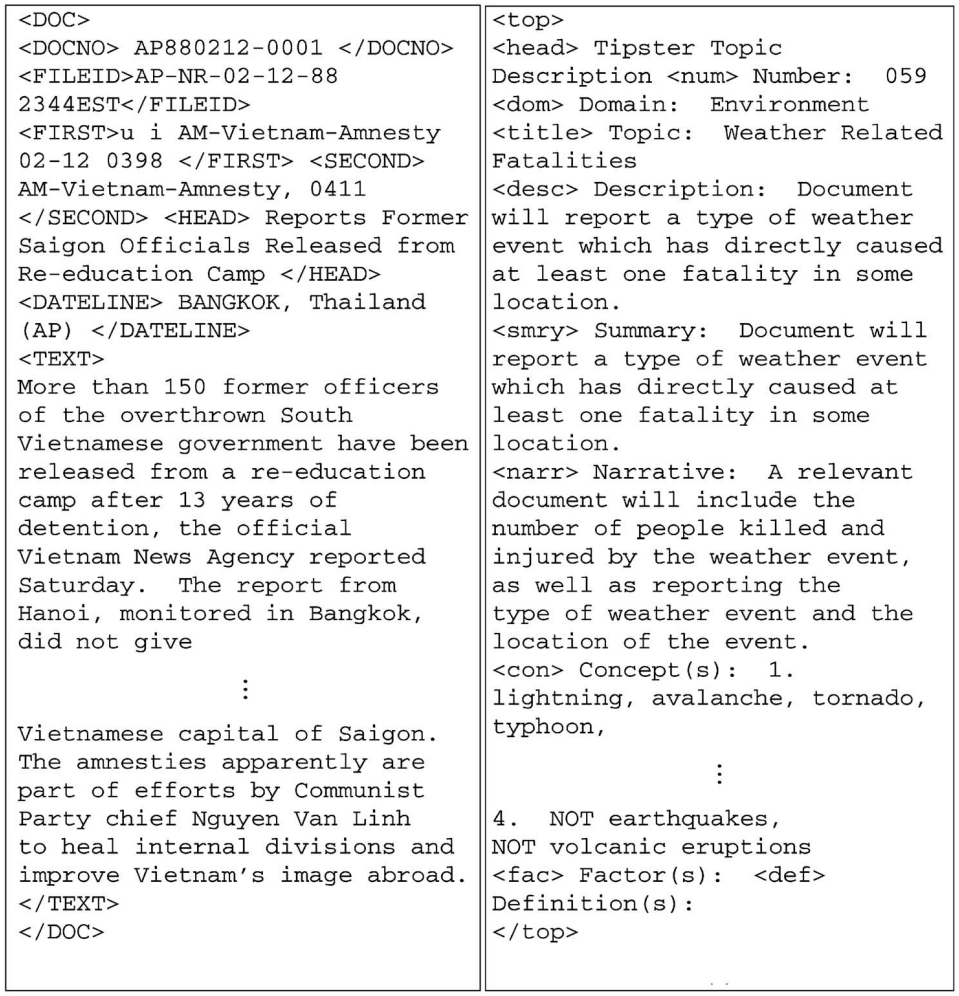
Rather than finding elements of the score vector which have high values, we might want to find the elements which have the largest phase precision component. Components which have high phase precision will identify a sinusoidal pattern of all of the query terms appearing together through the document. Again, the component with the largest phase precision will be the zeroth component. For this reason, the two components with the highest phase precision will be summed to give the score. The score equation will be as in (14), but the components $b_1$ and $b_2$ are chosen according to the following equation:

$$\Phi^\star_{d,b_1}, \Phi^\star_{d,b_2} \geq \max_{\forall b \neq b_1, b_2} (\Phi^\star_{d,b}),$$

where $\Phi^\star_{d,b}$ represents which ever of the three phase precision methods we choose.

### 2.6.4 Sum Largest Magnitude Components

Another score calculation method is to include only the score vector elements with larger magnitude components. The magnitude of the frequency components of the query term vectors gives us some understanding of how many query terms appeared in the pattern of the sinusoidal wave (to the corresponding frequency component). As in the last



```
<DOC>
<DOCNO> AP880212-0001 </DOCNO>
<FILEID>AP-NR-02-12-88
2344EST</FILEID>
<FIRST>u i AM-Vietnam-Amnesty
02-12 0398 </FIRST> <SECOND>
AM-Vietnam-Amnesty, 0411
</SECOND> <HEAD> Reports Former
Saigon Officials Released from
Re-education Camp </HEAD>
<DATELINE> BANGKOK, Thailand
(AP) </DATELINE>
<TEXT>
More than 150 former officers
of the overthrown South
Vietnamese government have been
released from a re-education
camp after 13 years of
detention, the official
Vietnam News Agency reported
Saturday.  The report from
Hanoi, monitored in Bangkok,
did not give
                ⋮
Vietnamese capital of Saigon.
The amnesties apparently are
part of efforts by Communist
Party chief Nguyen Van Linh
to heal internal divisions and
improve Vietnam's image abroad.
</TEXT>
</DOC>
```

(a)

```
<top>
<head> Tipster Topic
Description <num> Number:  059
<dom> Domain:  Environment
<title> Topic:  Weather Related
Fatalities
<desc> Description:  Document
will report a type of weather
event which has directly caused
at least one fatality in some
location.
<smry> Summary:  Document will
report a type of weather event
which has directly caused at
least one fatality in some
location.
<narr> Narrative:  A relevant
document will include the
number of people killed and
injured by the weather event,
as well as reporting the
type of weather event and the
location of the event.
<con> Concept(s):  1.
lightning, avalanche, tornado,
typhoon,
                ⋮
4.  NOT earthquakes,
NOT volcanic eruptions
<fac> Factor(s):  <def>
Definition(s):
</top>
```

(b)

Fig. 4. (a) A typical document from the TREC document set. (b) A sample query from the TREC collection.

scoring method, the components with the two greatest magnitudes will be taken to create the score for each document. Again, the component with the greatest magnitude will be the zeroth component.

$$H^*_{d,b_1}, H^*_{d,b_2} \geq \max_{\forall b \neq b_1, b_2} (H_{d,b}).$$

The score components added will be $s_{b_1}$ and $s_{b_2}$.

#### 2.6.5 Sum Threshold Phase Precision Components

The final score calculation method is to include only the score vector elements with a phase precision greater than some predefined constant. The idea behind this method is that the components with low phase precision will only disturb the quality of the final score, if the phase precision is low, it must mean that the set of query terms were not in phase for that component, and is therefore considered noise.

$$S_d = \sum_{b \in \{c | \Phi^\star_{d,c} > P\}} s_{d,b},$$

where $P$ is the threshold chosen.

## 3 Comparison to Vector Space Method

If we examine the FDS method, we can see that the vector space method is a special case of FDS where $B = 1$. To show this, we will step through the FDS method with $B = 1$.

**Theorem 1.** *Vector space methods are a special case of FDS where* $B = 1$.

**Proof.** The first step is to gather the document terms into bins. $B = 1$, so the whole document is considered to be a bin, therefore:

$$f_{d,t,0} = f_{d,t} \tag{15}$$

when the document weighting is performed, we apply the weighting to each bin, which is the document in this case:

$$\tilde{\omega}_{d,t} = \Omega_d(\tilde{f}_{d,t}) \tag{16}$$

$$= \Omega_d(f_{d,t,0}) \tag{17}$$

$$= \Omega_d(f_{d,t}). \tag{18}$$

The Fourier transform of a signal of length one is itself. The Magnitude vector ($H^*_{d,b}$) will be the same for both sum vectors and sum magnitudes, since $\nu_{d,t,0}$ is real and positive:

TABLE 2
Results of FDS Sensitivity Analysis with Respect to the Number of Bins per Term Signal, on Document Set AP2WSJ2, Using Queries 51-200

| Bins | Prec.5 | Prec.10 | Prec.15 | Prec.20 |
|---|---|---|---|---|
| 2 | 0.4573 | 0.4387 | 0.4076 | 0.3950 |
| 4 | 0.4733 | 0.4480 | 0.4227 | 0.4100 |
| 8 | *0.4947* | *0.4673* | *0.4493* | *0.4220* |
| 16 | 0.4933 | 0.4587 | 0.4351 | 0.4167 |
| 32 | 0.4853 | 0.4573 | 0.4453 | 0.4217 |

*AB-AFD-BAA preweighting was used with sum magnitudes and zero phase precision. Greatest precision is italicized.*

$$H^*_{d,0} = \sum_{t \in T} \nu_{d,t,0} \cdot \Omega_q(f_{q,t}) \tag{19}$$

$$= \sum_{t \in T} \omega_{d,t,0} \cdot \Omega_q(f_{q,t}) \tag{20}$$

$$= \sum_{t \in T} \Omega_q(f_{d,t}) \cdot \Omega_q(f_{q,t}). \tag{21}$$

The nonzero phase precision will be 1 because $\nu_{d,t,0}$ is real and positive, and the zero phase precision will be a fraction of the terms found in the document. Therefore, the vector space method is a special case of the FDS method where $B = 1$. □

## 4 Performance with Large Queries

Given that the DC component of a term signal spectrum is the most dominant component, how do the rest of the components affect the score? All of the methods are based on the idea that if the terms appear in phase, then the document should be more relevant than a document out of phase. This can be easily seen for a few query terms, but what about when there are many query terms (as in the TREC query shown in Fig. 4b)?

Since the DC ($b = 0$) component is the largest component and always has zero phase, any other component will also have to have a common phase across the set of query terms in order to be comparable to the DC component. For a component to have common phase across the entire set of query terms, the query terms must occur in the same positional bins. If the number of query terms is greater than the number of terms per bin, the query terms have no chance of appearing in the same bins. For each new query term added, the commonality between phases will reduce. Therefore, if the number of query terms is much greater than the number of terms per bin, then the AC ($b \neq 0$) components will be insignificant when compared to the DC component.

If we look at the basic expression for magnitude $\times$ phase precision (12), we can see that the phase precision acts as a weighting term. If we consider component $b$ (not equal to 0), we can see that the phase precision will remain at value 1 as long as all terms in the query are in the same positional bins in the document. Once the number of query terms passes the number of terms per bin, it is impossible for all terms in the query appear in the same positional bins. Therefore, the phase precision will reduce with every new term added to the query term set. Therefore:

$$\#(T) \gg B \Rightarrow \Phi_{d,b} < \epsilon \quad \forall b \in \{1, 2, \ldots, B-1\}, \tag{22}$$

TABLE 3
Method Names Are of the Form $\mathrm{fds}$-$x$-$y$-$z$.$\mathrm{bn}$, where the Values of $x, y, z, n$ Associate to the Description in This Table

| Label | Value | Description |
|---|---|---|
| $x$ | 5 | BD-ACI-BCA preweighting |
| | 7 | AB-AFD-BAA preweighting |
| | 8 | BI-ACI-BCA preweighting |
| | 9 | Lnu.ltu preweighting |
| $y$ | 1 | Sum vectors with No phase precision |
| | 3 | Sum magnitude with Non-zero phase precision |
| | 4 | Sum magnitude with Zero phase precision |
| $z$ | 1 | Sum components in order |
| | 2 | Sum largest phase precision components |
| | 3 | Sum largest magnitude components |
| | 4 | Sum largest score components |
| | 5 | Sum threshold phase precision components |
| $n$ | 1-5 | Number of score components added |

where $\epsilon$ is a small and positive. This implies that if a large number of query terms are used, the DC component will be the dominant factor in the score.

$$\lim_{\#(T) \to \infty} S_d = s_{d,0}. \tag{23}$$

We have also found in the previous section (Section 3) that the DC component is related to the value used in vector space similarity measures. Based on these two observations, it follows that if the number of query terms is large, FDS will be comparable to the vector space similarity measure using the weighting method chosen. So, if we choose the BD-ACI-BCA weighting for FDS, it should be similar to the BD-ACI-BCA vector space measure when large queries are used.

## 5 Experiments

Initial results have shown that FDS improves the accuracy of search results on small document sets (containing about 100 to 1,000 documents) [17]. In these experiments, we will show that FDS can effectively improve the results of large document sets. By doing so, we will also show that using FDS in a Web search engine would be a great enhancement.

Before the documents were indexed, preprocessing was performed. This preprocessing consisted of case folding, removing stop words (the stop word list contained about 400 common English words), and stemming was performed using Porter's stemming algorithm [20]. The FDS methods used eight[1] bins to record the approximate position of the terms throughout the document. The number of bins chosen gives a trade off between accuracy and speed. Experiments on term signals of different lengths are shown in Table 2. For this document set (AP2WSJ2), eight bins provides the best precision.

### 5.1 Document and Query Set

The document sets used are part of the TREC English document collection. The TREC documents come from many sources and are split on to several disks to allow the users to compare specific sets. The documents used in these experiments are from the Associated Press Newswire (1988) disk 2, the Wall Street Journal disk 2 (AP2WSJ2), the Federal Register disk 2 (1989), and articles from Ziff-Davis disk 2

1. A power of 2 is needed to perform the FFT.

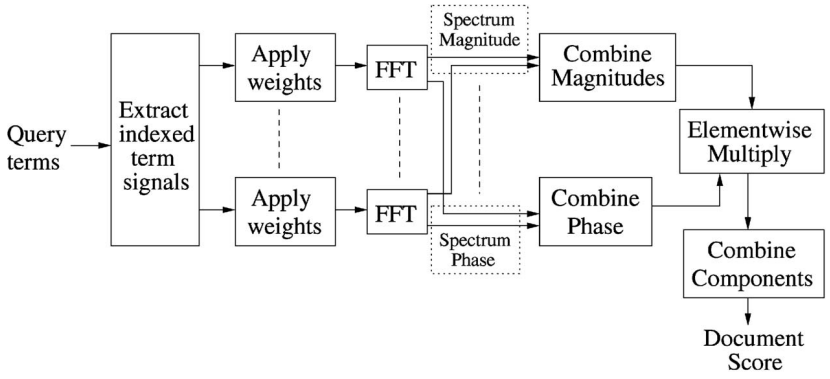



Fig. 5. The document score calculation process. One term signal is extracted from the index for each query term in each document.

TABLE 4
Best 20 Document Retrieval Methods Ordered by Precision after (a) 5 and (b) 10 Documents Were Retrieved for Titles of Queries 51-200 on AP2WSJ2

| Method | Precision 5 |
|---|---|
| fds-7-4-2.b2 | 0.5027 |
| fds-5-4-4.b2 | 0.5027 |
| fds-7-4-4.b2 | 0.4973 |
| fds-5-4-3.b2 | 0.4960 |
| fds-9-4-1.b5 | 0.4960 |
| fds-7-4-1.b5 | 0.4947 |
| fds-7-4-3.b2 | 0.4933 |
| fds-5-4-2.b2 | 0.4933 |
| fds-7-4-1.b4 | 0.4933 |
| fds-9-4-2.b2 | 0.4920 |
| fds-7-4-1.b3 | 0.4880 |
| vsm:AB-AFD-BAA | 0.4880 |
| fds-5-4-1.b5 | 0.4867 |
| fds-9-4-4.b2 | 0.4867 |
| fds-9-4-1.b4 | 0.4867 |
| fds-7-4-1.b2 | 0.4867 |
| fds-9-4-3.b2 | 0.4853 |
| fds-7-4-5.b2 | 0.4813 |
| fds-5-4-1.b4 | 0.4800 |
| fds-9-4-1.b3 | 0.4747 |
| vsm:Lnu.ltu | 0.4693 |
| vsm:BD-ACI-BCA | 0.4440 |
| vsm:BI-ACI-BCA | 0.4347 |

(a)

| Method | Precision 10 |
|---|---|
| fds-7-4-4.b2 | 0.4687 |
| fds-7-4-1.b5 | 0.4673 |
| fds-7-4-3.b2 | 0.4667 |
| fds-5-4-1.b5 | 0.4647 |
| fds-5-4-3.b2 | 0.4627 |
| fds-7-4-2.b2 | 0.4620 |
| fds-5-4-2.b2 | 0.4620 |
| fds-9-4-1.b5 | 0.4613 |
| fds-7-4-1.b4 | 0.4613 |
| fds-5-4-4.b2 | 0.4600 |
| fds-9-4-4.b2 | 0.4587 |
| fds-9-4-3.b2 | 0.4587 |
| fds-9-4-2.b2 | 0.4580 |
| fds-7-4-1.b3 | 0.4580 |
| fds-9-4-1.b4 | 0.4567 |
| fds-5-4-1.b3 | 0.4560 |
| fds-5-4-1.b4 | 0.4547 |
| fds-7-4-5.b2 | 0.4547 |
| fds-7-4-1.b2 | 0.4547 |
| fds-9-4-1.b3 | 0.4547 |
| vsm:AB-AFD-BAA | 0.4493 |
| vsm:Lnu.ltu | 0.4493 |
| vsm:BD-ACI-BCA | 0.4247 |
| vsm:BI-ACI-BCA | 0.4100 |

(b)

*Vector space methods not in the best 20 are shown below the bar.*

(FR2ZIFF2). This information can be found at the TREC Web site [21]. Each document in the set is separated by SGML tags and comes in the form shown in Fig. 4a. The AP2WSJ and FR2ZIFF2 collections contain 154,439 and 76,780 news articles, respectively.

The queries used were those corresponding to the ad-hoc tasks in the TREC-1 (queries 51 to 100), TREC-2 (queries 101 to 150), and TREC-3 (queries 151 to 200) conferences. These can also be found on the TREC web site [21]. A typical query is shown in Fig. 4b. As we can see by the sample query shown, a typical TREC query is not equivalent to a typical Web search engine query. We have also shown in Section 4 that the FDS method is better suited for short Web like queries since the only score component which will contribute for large queries is the zeroth. To extract Web like queries from the query set, we chose to use only the terms which appeared in the title section of the query. The titles only consist of a few terms which are very similar to those which a typical Web user would issue when searching for articles on the same topic.

To evaluate the methods performed on each document set, we examined the precision[2] at 10 and 20 documents retrieved. This gives a good idea as to which methods would display the correct results on the first and second pages of results if used as a Web search engine.

### 5.2 Methods Performed

The FDS methods performed were combinations of the document weighting, magnitude, and phase precision, and summing components shown in Table 3. The document score calculation process is shown in Fig. 5. Note that the $x$ value in Table 3 refers to the "apply weights" box in Fig. 5, the $y$ value in Table 3 refers to the "combine magnitudes" and "combine phase" boxes in Fig. 5 and the $z$ value in Table 3 refers to the "combine components" box in Fig. 5. These methods were compared to the vector space method (denoted as vsm: *weighting scheme* in the tabulated results)

2. Precision is the proportion of relevant documents to total documents retrieved.

TABLE 5
Best 20 Document Retrieval Methods Ordered by Precision after (a) 15 and (b) 20 Documents Were Retrieved for Titles of Queries 51-200 on AP2WSJ2

| Method | Precision 15 |
|---|---|
| fds-7-4-1.b5 | 0.4493 |
| fds-7-4-1.b4 | 0.4458 |
| fds-5-4-1.b5 | 0.4440 |
| fds-9-4-1.b4 | 0.4440 |
| fds-9-4-4.b2 | 0.4436 |
| fds-7-4-4.b2 | 0.4431 |
| fds-5-4-1.b4 | 0.4413 |
| vsm:AB-AFD-BAA | 0.4404 |
| fds-7-4-2.b2 | 0.4396 |
| fds-9-4-1.b5 | 0.4391 |
| fds-5-4-4.b2 | 0.4382 |
| fds-9-4-2.b2 | 0.4378 |
| fds-7-4-3.b2 | 0.4369 |
| fds-9-4-3.b2 | 0.4369 |
| fds-5-4-2.b2 | 0.4364 |
| fds-9-4-1.b3 | 0.4360 |
| vsm:Lnu.ltu | 0.4356 |
| fds-7-4-1.b2 | 0.4351 |
| fds-7-4-1.b3 | 0.4320 |
| fds-5-4-1.b3 | 0.4320 |
| vsm:BD-ACI-BCA | 0.4142 |
| vsm:BI-ACI-BCA | 0.3862 |

(a)

| Method | Precision 20 |
|---|---|
| fds-7-4-4.b2 | 0.4247 |
| fds-9-4-1.b5 | 0.4227 |
| fds-9-4-4.b2 | 0.4220 |
| fds-7-4-2.b2 | 0.4220 |
| fds-7-4-1.b5 | 0.4220 |
| vsm:AB-AFD-BAA | 0.4217 |
| fds-7-4-3.b2 | 0.4210 |
| fds-7-4-1.b4 | 0.4207 |
| fds-5-4-1.b4 | 0.4197 |
| fds-5-4-3.b2 | 0.4193 |
| fds-5-4-1.b5 | 0.4193 |
| fds-9-4-1.b4 | 0.4190 |
| vsm:Lnu.ltu | 0.4180 |
| fds-5-4-4.b2 | 0.4177 |
| fds-5-4-2.b2 | 0.4167 |
| fds-5-4-1.b3 | 0.4167 |
| fds-9-4-3.b2 | 0.4147 |
| fds-7-4-1.b3 | 0.4147 |
| fds-9-4-2.b2 | 0.4127 |
| fds-9-4-1.b3 | 0.4117 |
| vsm:BD-ACI-BCA | 0.3953 |
| vsm:BI-ACI-BCA | 0.3657 |

(b)

*Vector space methods not in the best 20 are shown below the bar.*

TABLE 6
Best 20 Document Retrieval Methods Ordered by Precision after (a) 5 and (b) 10 Documents Were Retrieved for Titles of Queries 51-200 on FR2ZIFF2

| Method | Precision 5 |
|---|---|
| fds-9-4-1.b4 | 0.2119 |
| fds-9-4-5.b2 | 0.2079 |
| fds-9-4-1.b5 | 0.2059 |
| fds-9-4-1.b3 | 0.2020 |
| vsm:Lnu.ltu | 0.2000 |
| fds-9-4-3.b2 | 0.1980 |
| fds-9-4-4.b2 | 0.1980 |
| vsm:AB-AFD-BAA | 0.1960 |
| fds-7-4-1.b4 | 0.1960 |
| fds-5-4-5.b2 | 0.1960 |
| fds-7-4-1.b5 | 0.1960 |
| fds-7-4-5.b2 | 0.1960 |
| fds-5-4-1.b4 | 0.1941 |
| fds-9-4-2.b2 | 0.1901 |
| fds-5-4-1.b2 | 0.1901 |
| fds-9-4-1.b2 | 0.1881 |
| fds-5-4-1.b3 | 0.1861 |
| fds-5-4-1.b5 | 0.1822 |
| fds-7-1-1.b5 | 0.1762 |
| vsm:BD-ACI-BCA | 0.1762 |
| vsm:BI-ACI-BCA | 0.1267 |

(a)

| Method | Precision 10 |
|---|---|
| fds-9-4-1.b5 | 0.1713 |
| fds-9-4-1.b3 | 0.1644 |
| fds-9-4-5.b2 | 0.1644 |
| fds-9-4-1.b2 | 0.1624 |
| fds-9-4-1.b4 | 0.1624 |
| fds-9-4-2.b2 | 0.1614 |
| vsm:Lnu.ltu | 0.1614 |
| fds-9-4-3.b2 | 0.1614 |
| fds-9-4-4.b2 | 0.1614 |
| vsm:AB-AFD-BAA | 0.1594 |
| fds-7-1-1.b4 | 0.1584 |
| fds-5-4-1.b3 | 0.1584 |
| fds-5-4-1.b5 | 0.1584 |
| fds-7-4-1.b4 | 0.1574 |
| fds-5-4-5.b2 | 0.1574 |
| fds-5-4-1.b4 | 0.1574 |
| fds-7-4-5.b2 | 0.1554 |
| fds-5-4-1.b2 | 0.1554 |
| fds-7-4-1.b5 | 0.1545 |
| fds-7-1-1.b5 | 0.1535 |
| vsm:BD-ACI-BCA | 0.1376 |
| vsm:BI-ACI-BCA | 0.0901 |

(b)

*Vector space methods not in the best 20 are shown below the bar.*

using the same weighting schemes (BD-ACI-BCA, AB-AFD-BAA, BI-ACI-BCA, and the SMART Lnu.ltu weighting). Tables 4 and 5 show the results for the top 20 methods using AP2WSJ2 when ranked by precision after 5 and 10 documents, and 15 and 20 documents, respectively. Tables 6 and 7 show the same using the FR2ZIFF2 collection. Any vector space methods which did not make the top 20 are shown at the bottom of the table.

The results show us that using FDS does boost the precision of a document search when using Web like queries. We can see from the results in all of the short query tables that, by using FDS, we are able to increase the precision of the underlying vector space method chosen. For example, fds-9-4-1.b5 uses SMART's Lnu.ltu weighting, we can see in the tables that it appears above SMART, therefore, it has increased the precision. We can also see that most of the methods appearing in the top results use sum

TABLE 7
Best 20 Document Retrieval Methods Ordered by Precision after (a) 15 and (b) 20 Documents Were Retrieved for Titles of Queries 51-200 on FR2ZIFF2

| Method | Precision 15 |
|---|---|
| fds-9-4-1.b4 | 0.1505 |
| fds-9-4-1.b5 | 0.1465 |
| vsm:Lnu.ltu | 0.1459 |
| fds-9-4-1.b3 | 0.1452 |
| fds-5-4-5.b2 | 0.1432 |
| fds-5-4-1.b1 | 0.1426 |
| fds-9-4-5.b2 | 0.1426 |
| fds-9-4-2.b2 | 0.1426 |
| fds-5-4-2.b2 | 0.1419 |
| fds-5-4-3.b2 | 0.1419 |
| fds-7-1-1.b5 | 0.1419 |
| fds-5-4-1.b5 | 0.1413 |
| fds-9-4-1.b2 | 0.1413 |
| fds-9-4-4.b2 | 0.1406 |
| fds-5-4-4.b2 | 0.1399 |
| fds-5-4-1.b4 | 0.1399 |
| fds-9-1-1.b5 | 0.1386 |
| fds-5-4-1.b2 | 0.1386 |
| fds-5-4-1.b3 | 0.1380 |
| fds-7-1-1.b4 | 0.1360 |
| vsm:AB-AFD-BAA | 0.1360 |
| vsm:BD-ACI-BCA | 0.1201 |
| vsm:BI-ACI-BCA | 0.0812 |

(a)

| Method | Precision 20 |
|---|---|
| fds-9-4-1.b5 | 0.1366 |
| vsm:Lnu.ltu | 0.1342 |
| fds-5-4-4.b2 | 0.1337 |
| fds-9-4-1.b4 | 0.1332 |
| fds-5-4-1.b3 | 0.1327 |
| fds-5-4-2.b2 | 0.1327 |
| fds-5-4-1.b2 | 0.1322 |
| fds-5-4-1.b1 | 0.1312 |
| fds-5-4-3.b2 | 0.1307 |
| fds-9-4-1.b3 | 0.1302 |
| fds-5-4-1.b4 | 0.1302 |
| fds-5-4-1.b5 | 0.1287 |
| fds-9-4-5.b2 | 0.1282 |
| fds-5-4-5.b2 | 0.1277 |
| fds-9-1-1.b5 | 0.1272 |
| fds-9-4-4.b2 | 0.1267 |
| fds-7-1-1.b5 | 0.1257 |
| fds-9-4-1.b2 | 0.1252 |
| fds-9-4-2.b2 | 0.1252 |
| fds-7-1-1.b4 | 0.1252 |
| vsm:AB-AFD-BAA | 0.1238 |
| vsm:BD-ACI-BCA | 0.1139 |
| vsm:BI-ACI-BCA | 0.0748 |

(b)

*Vector space methods not in the best 20 are shown below the bar.*

magnitudes with zero phase precision ($x.4.z.bn$), and about half of the results use sum components ($x.y.1.bn$) in order where a higher value of $n$ usually means a higher precision.

To show the effect of long queries, we have also included a few results using the whole query (not just the title) in Table 8. We can see from these results that FDS drops down to the level of the vector space methods and, therefore, is not as effective in this case.

TABLE 8
Best 20 Document Retrieval Methods Ordered by Precision after 10 Documents Retrieved for Long Queries 51-200 on AP2WSJ2

| Method | 5 | 10 | 15 | 20 |
|---|---|---|---|---|
| vsm:BD-ACI-BCA | 0.6040 | 0.5740 | 0.5453 | 0.5257 |
| vsm:Lnu.ltu | 0.5800 | 0.5587 | 0.5396 | 0.5207 |
| fds-5-4-1.b5 | 0.5987 | 0.5500 | 0.5200 | 0.5027 |
| fds-5-4-1.b4 | 0.5933 | 0.5473 | 0.5182 | 0.5003 |
| fds-5-4-1.b3 | 0.5840 | 0.5427 | 0.5178 | 0.4947 |
| fds-5-4-1.b2 | 0.5733 | 0.5420 | 0.5187 | 0.4917 |
| fds-5-4-1.b1 | 0.5560 | 0.5300 | 0.5049 | 0.4790 |
| vsm:BI-ACI-BCA | 0.5613 | 0.5233 | 0.5018 | 0.4783 |
| fds-5-1-1.b1 | 0.5347 | 0.5213 | 0.4956 | 0.4727 |
| fds-9-4-1.b4 | 0.5640 | 0.5173 | 0.4867 | 0.4737 |
| fds-9-4-1.b5 | 0.5640 | 0.5173 | 0.4898 | 0.4690 |
| fds-9-4-1.b3 | 0.5560 | 0.5133 | 0.4853 | 0.4670 |
| fds-9-4-1.b2 | 0.5413 | 0.5073 | 0.4853 | 0.4693 |
| fds-8-4-1.b5 | 0.5427 | 0.5007 | 0.4769 | 0.4483 |
| fds-9-4-1.b1 | 0.5307 | 0.4980 | 0.4729 | 0.4560 |
| fds-8-4-1.b4 | 0.5480 | 0.4967 | 0.4680 | 0.4470 |
| fds-8-4-1.b3 | 0.5333 | 0.4913 | 0.4676 | 0.4403 |
| fds-8-4-1.b2 | 0.5293 | 0.4833 | 0.4631 | 0.4397 |
| fds-8-4-1.b1 | 0.5013 | 0.4733 | 0.4493 | 0.4250 |
| fds-7-4-1.b5 | 0.5013 | 0.4620 | 0.4449 | 0.4223 |
| vsm:AB-AFD-BAA | 0.4507 | 0.4147 | 0.4062 | 0.3833 |

*Vector space methods not in the best 20 are shown below the bar.*

## 6 Conclusion

Internet search engines have become an essential tool for locating resources and information on the Web. Due to the large majority of Web information being textual, a need arises for a superior text search engine. We have presented a novel and simple document ranking method called Fourier Domain Scoring (FDS), which provides more precise results than the currently used vector space similarity methods. We also showed that the existing vector space similarity methods are a special case of FDS. The results show that FDS is a superior method because it makes use of the spatial information within a document rather than the count of each query term.

Results were shown for two different large datasets supplied by TREC. The experiments were set up to simulate text retrieval on the Web and compared to existing popular vector space methods. We showed that FDS produced more precise results than the vector space methods for both data sets when using Web like queries, but reverted back to vector space method quality when using long queries.

This implies that FDS would enhance the performance of a Web search engine if used in the place of a vector space measure as the text classification method.

## Acknowledgments

The authors would like to thank the ARC Special Research Centre for Ultra-Broadband Information Networks for their support and funding of this research. They would also like to thank Alistair Moffat and his research team for use of their resources and help. Finally, they wish to thank the anonymous reviewers for their suggestions and comments for improving the quality of the presentation of their paper. This work was supported by the Australian Research Council.

## REFERENCES


[1] M. Marchiori, "The Quest for Correct Information on the Web: Hyper Search Engines," *Computer Networks and ISDN Systems,* vol. 29, pp. 1225-1235, 1997.

[2] D. Siaw, W. Ngu, and X. Wu, "Site Helper: A Localised Agent That Helps Incremental Exploration of the World Wide Web," *Computer Networks and ISDN Systems,* vol. 29, pp. 1249-1255, 1997.

[3] S.J. Carruère and R. Kazman, "Webquery: Searching and Visualising the Web through Connectivity," *Computer Networks and ISDN Systems,* vol. 29, pp. 1257-1267, 1997.

[4] E. Spertus, "Parasite: Mining Structural Information on the Web," *Computer Networks and ISDN Systems,* vol. 29, pp. 1205-1215, 1997.

[5] O. Etzioni, "Moving up the Information Food Chain," *AI Magazine,* vol. 18, pp. 11-18, Am. Assoc. for Artificial Intelligence, Summer 1997.

[6] I.H. Witten, A. Moffat, and T.C. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images.* Morgan Kaufmann Publishers, 1999.

[7] C. Buckley and J. Walz, "SMART in TREC 8," *Proc. Eighth Text Retrieval Conf.,* pp. 577-582, Nov. 1999.

[8] C. Buckley, A. Singhal, M. Mitra, and G. Salton, "New Retrieval Approaches Using Smart: TREC 4," *Proc. Fourth Text Retrieval Conf.,* pp. 25-48, Nov. 1995.

[9] S.E. Robertson and S. Walker, "Okapi/Keenbow at TREC-8," *Proc. Eighth Text Retrieval Conf.,* pp. 151-162, Nov. 1999.

[10] K. Yang and K. Maglaughlin, "IRIS at TREC-8," *Proc. Eighth Text Retrieval Conf.,* pp. 645-656, Nov. 1999.

[11] J. Allan, J. Callan, F.-F. Feng, and D. Malin, "Inquery and Trec-8," *Proc. Eighth Text Retrieval Conf.,* pp. 637-644, Nov. 1999.

[12] J. Zobel and A. Moffat, "Exploring the Similarity Space," *Proc. ACM SIGIR Forum,* vol. 32, pp. 18-34, Spring 1998.

[13] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems,* vol. 30, nos. 1-7, pp. 107-117, Apr. 1998.

[14] D. Hawking and P. Thistlewaite, "Proximity Operators—So Near and Yet So Far," *Proc. Fourth Text Retrieval Conf.,* pp. 131-144, Nov. 1995.

[15] D.S. Ebert, A. Zwa, and E.L. Miller, "Two-Handed Volumetric Document Corpus Mangement," *IEEE Computer Graphics and Applications,* vol. 17, no. 4, pp. 60-62, July/Aug. 1997.

[16] M.W. Berry, S.T. Dumais, and G.W. O'Brien, "Using Linear Algebra for Intelligent Information Retrieval," Technical Report, Computer Science Dept., The Univ. of Tennessee, Knoxville, Dec. 1994.

[17] L.A.F. Park, M. Palaniswami, and R. Kotagiri, "Internet Document Filtering Using Fourier Domain Scoring," *Principles of Data Mining and Knowledge Discovery,* L. de Raedt and A. Siebes, eds., pp. 362-373, Springer-Verlag, pp. 362-373 Sept. 2001.

[18] S.T. Dumais, "Improving the Retrieval of Information from External Sources," *Behaviour Research Methods, Instruments & Computers,* vol. 23, no. 2, pp. 229-236, 1991.

[19] Wikipedia, Nyquist-Shannon Sampling Theorem, http://www.wikipedia.org/wiki/Nyquist-Shannon_sampling_theorem, Feb. 2003.

[20] M.F. Porter, "An Algorithm for Suffix Stripping," *Program,* vol. 14, no. 3, pp. 130-137, 1980.

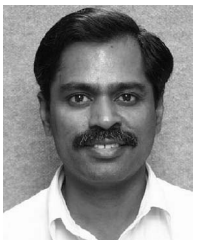
[21] *Proc. Text Retrieval Conf.,* Nat'l Inst. of Standards and Technology, http://trec.nist.gov/, 2001.

[22] *Proc. Eighth Text Retrieval Conf.,* E.M. Voorhees and D.K. Harman, eds., Nat'l Inst. of Standards and Technology special publication 500-246, Dept. of Commerce, Nov. 1999.

[23] *Proc. Fourth Text Retrieval Conf.,* D. Harman, ed., Nat'l Inst. of Standards and Technology special publication 500-236, Nov. 1995.


**Laurence A.F. Park** received the BE (Hons) and BSc degrees from the University of Melbourne, Australia in 1999. He is currently a research student in the ARC Special Research Centre for Ultra-Broadband Information Networks and the Electrical and Electronic Engineering Department at The University of Melbourne, working on his PhD dissertation on Web information retrieval. He is jointly supervised by Professor Palaniswami and Professor Rao Kotagiri. His other interests include music and image retrieval and analysis. He is a member of the IEEE and IEEE Computer Society.

**Kotagiri Ramamohanarao** received the BE degree from Andhra University in 1972, the ME degree from the Indian Institute of Science in 1974, and the PhD degree from Monash University in 1980. He joined the Computer Science Department at the University of Melbourne as a lecturer in 1980, was awarded the Alexander von Humboldt Fellowship in 1983, was promoted to senior lecturer in 1984 and to reader in 1988, and was appointed as a professor of computer science in 1989. He was head of the School of Electrical Engineering and Computer Science at the University of Melbourne, Australia, was codirector of the Key Center for Knowledge-Based Systems, and served as a member of the Australian Research Council Engineering Panel. He is on the editorial boards of the *Computer Journal* and the *Journal for Universal Computer Science*, is a fellow of the Institute of Engineers Australia, and has been on program committees and an invited speaker at several international conferences. He formerly served as research director for the Cooperative Research Center for Intelligent Decision Systems and, until 1996, was a member of the editorial board of the *Very Large Databases Journal*.



**Marimuthu Palaniswami** received the BE (Hons) degree from the University of Madras, the MEngSc degree from the University of Melbourne, and the PhD degree from the University of Newcastle, Australia. He is an associate professor and deputy director for the Centre for Networked Decision and Sensor Systems at the University of Melbourne, Australia. His research interests are in the fields of computational intelligence, computer networks, nonlinear dynamics, intelligent control, and Internet computing. He has published more than 180 conference and journal papers in these topics. He was an Associate Editor of the *IEEE Transactions on Neural Networks* and is on the editorial board of a few computing and electrical engineering journals. He served as a technical program cochair for the IEEE International Conference on Neural Networks, 1995, and has served on the programm committees of a number of international conferences. His invited presentations include several key note lectures and invited tutorials in the areas of machine learning, biomedical engineering, and control. He has completed several industry sponsored projects for National Australia Bank, Broken Hill Propriety Limited, Defence Science and Technology Organisation, Integrated Control Systems Pty Ltd, and Signal Processing Associates Pty Ltd. He has also been supported with several Australian Research Council grants, industry research and development grants, and industry research contracts. He was also a recipient of a foreign specialist award from the Ministry of Education, Japan. He is a senior member of the IEEE.