A Novel Web Text Mining Method Using the Discrete Cosine Transform

Laurence A.F. Park, Marimuthu Palaniswami, and Kotagiri Ramamohanarao

ARC Special Research Centre for Ultra-Broadband Information Networks
Department of Electrical & Electronic Engineering
The University of Melbourne
Parkville, Victoria, Australia 3010
lapark@ee.mu.oz.au
http://www.ee.mu.oz.au/cubin

Abstract. Fourier Domain Scoring (FDS) has been shown to give a 60% improvement in precision over the existing vector space methods, but its index requires a large storage space. We propose a new Web text mining method using the discrete cosine transform (DCT) to extract useful information from text documents and to provide improved document ranking, without having to store excessive data. While the new method preserves the performance of the FDS method, it gives a 40% improvement in precision over the established text mining methods when using only 20% of the storage space required by FDS.

1 Introduction

Text mining has been one of the great challenges to the knowledge discovery community and since the introduction of the Web, its importance has sky rocketed. The easiest way to search on the Web is to supply a set of query terms related to the information you want to find. A search engine will then proceed and try to find information that is associated to the query terms given. To classify a text based document using current vector space similarity measures, a search engine will compare the number of times the query terms appear, these results are then weighted and a relevance score is given to that document. Zobel and Moffat [7] showed that the most precise weighting scheme of the vector space measures is the BD-ACI-BCA method. This works very well on the TREC data set (where the queries are about 80 terms long), but as we have seen by results given by Web search engines, counting the words is sometimes not enough.

In [3] we showed how to utilise the spatial information in a document using Fourier Domain Scoring (FDS) to obtain more precise results. FDS not only records the number of times each word appears in the document, but also the positions of the words into entities called word signals. The Fourier transform is then applied to these word signals to obtain magnitude and phase information. This extra information can then be used to compare against other words.

Experiments [4] have shown that FDS gives similar results to the BD-ACI-BCA for long queries (containing about 80 terms), and a vast improvement of 60% greater precision for short queries (containing 1 to 5 terms).

T. Elomaa et al. (Eds.): PKDD, LNAI 2431, pp. 385–397, 2002.

We have shown through experimentation that using the Fourier transform on the word signals gave excellent results. However, the problem is that it requires more disk space to store the index (containing the extra spatial information) relative to the vector space methods and it requires more calculations to build the index and score the documents. With the intent to study the impact of reducing storage cost, we experimented in [4] using only a few frequency components but found the results were of a poorer quality. Therefore we propose a new method of document scoring using another transform which will give similar results to the Fourier transform, but not require as much information to obtain them.

When examining the properties of transforms, it is useful to note that one that keeps appearing in a wide range of disciplines is the Discrete Cosine Transform (DCT). The DCT decomposes a signal into the sum of cosine waves of different frequencies. Ahmed *et al.* [1] had first proposed the DCT to approximate the Karhunen-Loève transform. They showed that the DCT could easily be calculated by using the fast Fourier transform and that it also gave a close approximation to the KLT, which is used for signal compression. Some areas which have taken advantage of the compression property of the DCT are image (used in JPEG compression) and video compression (used in MPEG compression).

This paper is organised as follows. Section 2 will give a brief introduction to the Karhunen-Loève Transform, and its properties. Section 3 will introduce the discrete cosine transform and show its association to the KLT. Section 4 will explain the methods used in the document ranking experiments. Section 5 will outline the experiments performed and discuss some results. Finally, section 6 contains the conclusion.

2 Karhunen-Loève Transform

The Karhunen-Loève transform [2] (KLT, also known as Principle Component Analysis) adjusts the basis of a random signal, in a way as to diagonalise its covariance matrix. This is an important transform in signal compression, since it is able produce a transformed signal in which every element is linearly independent of each other, and will also order the basis functions in terms of importance to allow for easy least squares estimations.

The KLT is of the form:

$$\tilde{y} = T\tilde{x} \tag{1}$$

where $\tilde{x}$ is the input vector, $T = [t_0\ t_1\ \ldots\ t_{N-1}]^T$ is the transformation matrix containing the basis vectors t_n, and $\tilde{y}$ is the transformed vector. The basis vectors t_n are found by solving:

$$\mathrm{cov}\,(X)\, t_n = \lambda_n t_n \tag{2}$$

where $\mathrm{cov}\,(X)$ is the covariance matrix of the matrix X consisting of input vectors $\tilde{x}$ and λ_n is a constant. We can see that equation 2 is the eigenvalue problem. Therefore the basis vectors t_n are the eigenvectors of the covariance matrix of X.

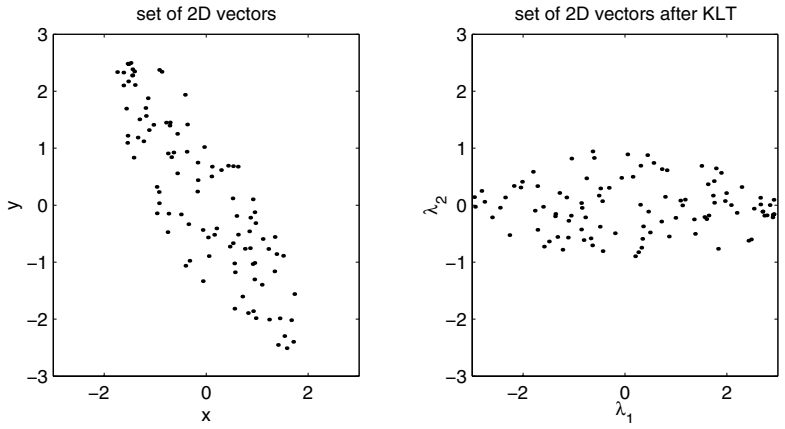

Fig. 1. Example of the Karhunen-Loève transform. The top plot displays 100 randomly generated points. The bottom plot shows the same points after performing the KLT

An example of the KLT can be seen in figure 1. We can see that the points have been mapped to a space where the x-axis is the dimension of greatest variance. The y-axis is the dimension of second greatest variance, which is also orthogonal to the x-axis. In this two dimensional case, the y axis is also the dimension of least variance.

To perform the KLT, we must take the whole data set into consideration. For large data sets, this can be computationally expensive. Many experiments have been performed to find the best estimate of the KLT which requires less calculations. It was shown in Ahmed *et al.* [1] that the DCT is a good approximation to the KLT for first order stationary Markov processes. A first order Markov process is defined as a random process $\{\ldots, X_{n-2}, X_{n-1}, X_n, X_{n+1}, X_{n+2}, \ldots\}$ such that:

$$\Pr(X_n = x_n | X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, \ldots) = \Pr(X_n = x_n | X_{n-1} = x_{n-1})$$

for each n. A stationary Markov process implies that the conditional probability $\Pr(X_n = x_n | X_{n-1} = x_{n-1})$ is independent of n. The signals we will be observing are word signals found in the FDS method. A weighted word signal ($\tilde{w}_{d,t}$) consists of the positions of term t in document d. Therefore, if we consider the weighted word count $w_{d,t,b}$ as a state and the bin position as the time (n), we can treat the word signal as a random process. Due to the nature of the English language, we will assume it is safe to identify a word signal as a first order stationary Markov process. The probability of term t appearing, after taking into account its previous appearances, should be similar to the probability when only taking into account its last appearance, independent of the bin position. Therefore by applying the DCT to a word signal, we are approximating the KLT of the word signal.

3 The Discrete Cosine Transform

The Discrete Cosine Transform (DCT), like the Fourier Transform, converts a sequence of samples to the frequency domain. But unlike the Fourier transform, the basis is made up of cosine waves, therefore each basis wave has a set phase

The DCT is of the form:

$$\tilde{X} = \text{DCT}(\tilde{x}) \qquad X_k = \sum_{b=0}^{B-1} x_b \cos \frac{(2b+1)k\pi}{2B} \tag{3}$$

where $\tilde{X} = [X_0 \ X_1 \ \ldots \ X_{B-1}]$ and $\tilde{x} = [x_0 \ x_1 \ \ldots \ x_{B-1}]$ Therefore, a real positive signal (as in a word signal) maps to a real signal after the performing cosine transform. The DCT was introduced to solve the problems of pattern recognition and Wiener filtering [1].

To obtain significant features in a data set, a transform is usually applied, the features are selected and then the inverse transform is applied. The most significant features are the ones with the greatest variance. As shown, the KLT transforms a set of vectors, so that each component of the vector represents the direction of greatest variance in decreasing order. Therefore KLT is optimal for this task. We have seen that the DCT is a good approximation to the KLT for first order stationary Markov processes. Therefore the DCT should be a good choice for transforming to a space for easy feature selection (as in JPEG and MPEG).

4 Cosine Domain Scoring Method

In a recent paper on the FDS [4] method, we explained steps of the scoring method and proposed different ways to perform each step. The steps of performing the Cosine Domain Scoring (CDS) method on a document are:

1. Extract query term word signals $\tilde{f}_{d,t}$
2. Perform preweighting ($\tilde{f}_{d,t} \rightarrow \tilde{w}_{d,t}$)
3. Perform DCT ($\tilde{\eta}_{d,t} = \text{DCT}(\tilde{w}_{d,t})$)
4. Combine word spectrums ($\tilde{\eta}_{d,t} \rightarrow \tilde{s}_d$)
5. Combine word spectrum components ($\tilde{s}_d \rightarrow s_d$)

In this section we will look into the steps which differ from the FDS method.

4.1 Preweighting

When querying using a vector space method, weights are always applied to the term counts from documents to emphasise the significance of a term. The TBF×IDF and PTF×IDF weighting schemes [4] are both variants of the TF×IDF [6] which have been adjusted to suit the use of word signals. These are defined as:

$$\text{TBF}: \; w_{d,t,b} = 1 + \log_e f_{d,t,b} \tag{4}$$

where $f_{d,t,b}$ and $w_{d,t,b}$ are the count and weight of term t in spatial bin b of document d respectively.

$$\text{PTF}: \; w_{d,t,b} = (1 + \log_e f_{d,t}) \left(\frac{f_{d,t,b}}{f_{d,t}} \right) \tag{5}$$

where $f_{d,t}$ is the count of term t in document d. The preweighting of CDS will consist of one of these two methods or a variant of the BD-ACI-BCA weighting. The variant takes into account the word signals by replacing $w_{d,t}$ with:

$$w_{d,t,b} = r_{d,t,b} = 1 + \log_e f_{d,t,b} \tag{6}$$

The same values of W_d and W_q are used.

4.2 Combination of Word Spectrums

Once the DCT has been performed, we must combine all of the query word spectrums into one. In this experiment, this was done in two ways. The first called magnitude, the second called magnitude×selective phase precision. The combined word spectrum is defined as:

$$\tilde{s}_d = \left[s_{d,0} \; s_{d,1} \; \ldots \; s_{d,B-1} \right] \qquad s_{d,b} = \Phi_{d,b} \sum_{t \in T} H_{d,t,b}$$

where B is the number of spatial bins chosen, $H_{d,t,b}$ is the magnitude of the bth frequency component of the tth query term in the dth document and $\Phi_{d,b}$ is the phase precision of the bth frequency component in the dth document. The magnitude and phase precision values are extracted from the frequency components in the following way:

$$\eta_{d,t,b} = H_{d,t,b} \exp\left(i\theta_{d,t,b}\right) \tag{7}$$

where $\eta_{d,t,b}$ is the bth frequency component of the tth query term in the dth document. The phase vector is defined as follows:

$$\phi_{d,t,b} = \frac{\eta_{d,t,b}}{|\eta_{d,t,b}|} = \exp\left(i\theta_{d,t,b}\right)$$

The DCT does not produce complex values when applied to a real signal, so we can either ignore the phase or treat the sign of the component as the phase. If we ignore the phase, this implies that we let $\Phi_{d,b} = 1$ for all d and b, we call this method *magnitude*. In the case where we do not ignore the phase, $\eta_{d,t,b}$ is real and so $\theta_{d,t,b}$ must be of the form πn, where n is an integer. This implies that we will have only $\phi_{d,t,b} \in \{-1, 1\}$. The selective phase precision equation [4] can be simplified to:

$$\begin{aligned} \text{Selective phase precision} := \bar{\Phi}_{d,b} &= \left| \frac{\sum_{t \in T: H_{d,t,b} \neq 0} \phi_{d,t,b}}{\#(T)} \right| \\ &= \left| \frac{\sum_{t \in T} \text{sgn}_0\left(\eta_{d,t,b}\right)}{\#(T)} \right| \end{aligned} \tag{8}$$

where T is the set of query terms, $\#(T)$ is the cardinality of the set T, and

$$\text{sgn}_x(y) = \begin{cases} 1 & \text{if } y \geq 0 \\ x & \text{if } y = 0 \\ -1 & \text{if } y < 0 \end{cases}$$

4.3 Combination of Spectral Components

After combining the word spectrums into one single score spectrum, we are left with B elements to combine in some way to produce a score. If using the Fourier transform, only the first $B/2+1$ elements are used since the rest are the complex conjugate of these. If using the DCT, all elements need to be considered, there is no dependence on any of these elements. Methods that will be considered are:

- Sum all components
- Sum first b components

where $0 < b < B$. By summing all of the components we will be able to utilise all of the information obtained from the DCT of the word spectrums. The second method (Sum first b components) will be considered due to the closeness of the DCT to the KLT. When the KLT is performed on a signal, we are adjusting the basis of the signals space such that the dimensions are ordered in terms of importance. If we consider only the first b components, we will be making a least squares approximation of the spectral vector for b dimensions. Therefore by performing the DCT and taking the first b components, we should have a close approximation to the B dimensional vector in the b dimensional space.

5 Experiments

The experiments were split into three groups. The first consisted of a general comparison of CDS methods using the already classified TREC documents and queries. The second compared the best CDS method with FDS 3.4.1[1] [4] method using short queries to simulate the Web environment. The third examined the ability to reduce the dimension of the word signals after performing the DCT. All experiments used the AP-2 document set from TREC, which is a collection of news paper articles from the Associated Press in the year 1988. The number of bins per word signal was set to eight. Case folding, stop word removal and stemming were performed before the index was created. The "staggered" form of the cosine transform is used since this is the standard for data compression and processing [5]. Each experiment compares the CDS method with the existing FDS, and the current best vector space method BD-ACI-BCA. The experiments are explained in more detail in the following sections.

[1] FDS 3.4.1 uses TBF×IDF preweighting, DFT, selective phase precision and adds all components

Table 1. Methods performed in experiment A

Method	Weighting	Combine word spectrums	Combine spectral components
CDS 1.1	none	magnitude	add all components
CDS 2.1	TBF×IDF	magnitude	add all components
CDS 3.1	PTF×IDF	magnitude	add all components
CDS 4.1	BD-ACI-BCA	magnitude	add all components
CDS 1.2	none	magnitude×selective phase precision	add all components
CDS 2.2	TBF×IDF	magnitude×selective phase precision	add all components
CDS 3.2	PTF×IDF	magnitude×selective phase precision	add all components
CDS 4.2	BD-ACI-BCA	magnitude×selective phase precision	add all components

5.1 Experiment A : Method Selection

To get an idea of the performance of each of the DCT methods, we will use the standard queries and relevance lists supplied by TREC. The queries applied were those from the TREC-1,2 and 3 conferences (queries 51 to 200). Each query is on average 80 words long. In [4], we have seen that when queries with t terms are given, where $t \gg$ number of words per bin, then the performance of FDS will approach the performance of a vector space measure. Due to the similarity between the FDS and CDS methods, this can also be said for CDS.

Therefore, in this experiment, we are looking for a method which will give similar (or better) performance than the BD-ACI-BCA method *This experiment is not to simulate the environment of the Web, but to examine the relative performance of each of the CDS methods using a standard document set and queries.* The methods used are displayed in table 1. The results can be seen in table 2 and figure 2. We can see that the methods CDS 2.2 and 4.1 perform well relative to the other CDS methods. Method CDS 2.2 gives a precision close to the FDS 3.4.1 method and BD-ACI-BCA. This gives a good indication that the DCT can be used in place of the DFT.

5.2 Experiment B : Web Queries

To simulate the Web environment, we will perform experiments using short queries (containing 1 to 5 words). The short queries were created by taking the title of the TREC queries used in experiment A. Due to this shortening of each query, the specifics of each query was also relaxed. Therefore the document relevance lists had to be recompiled. To create the new relevance lists, the top twenty documents classified by each method were collected and judged relative to each query. Only the top twenty were observed to emulate the level of patience of the typical Web search engine user. The methods compared are the CDS 2.2 (considered the best method from experiment A for low levels of recall), FDS 3.4.1 and BD-ACI-BCA. The results can be viewed in figure 3 and table 3. We can see that both FDS methods produce very similar results and show a 60%

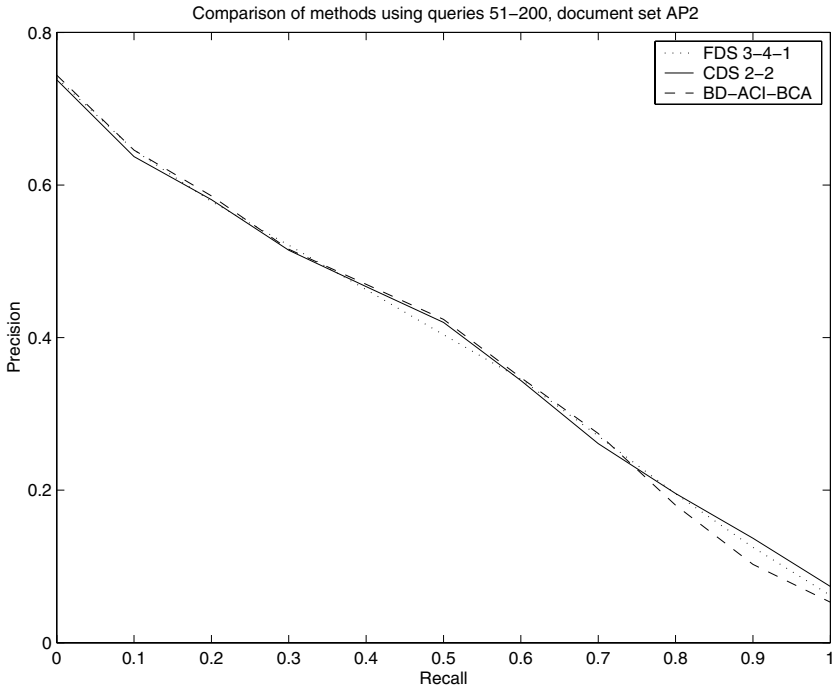

Fig. 2. Precision-recall plot for CDS 2.2, FDS 3.4.1 and BD-ACI-BCA using long query form of queries 51 to 200 and document set AP2

Table 2. Comparison of CDS methods using data set AP2 with the long form of queries 51 to 200. The largest CDS values per column are shown in italics

Method	Precision at Recall					Average	R-
	0%	10%	20%	30%	40%	Precision	Precision
BD-ACI-BCA	0.7441	0.6458	0.5858	0.5159	0.4698	0.3792	0.4039
FDS 3.4.1	0.7404	0.6457	0.5783	0.5211	0.4628	0.3816	0.4015
CDS 1.1	0.6826	0.5631	0.4801	0.4049	0.3490	0.2817	0.3149
CDS 2.1	0.6889	0.5967	0.5383	0.4659	0.4252	0.3451	0.3603
CDS 3.1	0.6418	0.5320	0.4521	0.3797	0.3440	0.2859	0.3183
CDS 4.1	0.7326	*0.6462*	*0.5800*	0.5143	0.4511	0.3707	0.3938
CDS 1.2	0.6926	0.5772	0.5012	0.4331	0.3676	0.3047	0.3338
CDS 2.2	0.7343	0.6420	0.5767	*0.5228*	*0.4648*	*0.3808*	*0.4026*
CDS 3.2	0.7093	0.6031	0.5282	0.4569	0.4058	0.3428	0.3607
CDS 4.2	*0.7438*	0.6298	0.5672	0.5010	0.4419	0.3619	0.3804

improvement over BD-ACI-BCA. For some queries CDS 2.2 performs slightly better than FDS 3.4.1, for some it performs slightly less. From these results, we can see that we would get approximately the same results whether using CDS 2.2 or FDS 3.4.1.

Table 3. This table shows the short queries applied to the AP2 document set. We can see that the CDS and FDS methods give more relevant documents out of the top 20 returned by each method

Query term	Relevant documents in top 20		
	BD-ACI-BCA	CDS 2.2	FDS 3.4.1
Airbus Subsidies	10	14	14
Satellite Launch Contracts	7	13	12
Rail Strikes	8	18	17
Weather Related Fatalities	6	10	11
Information Retrieval Systems	3	6	7
Attempts to Revive the SALT II Treaty	8	14	12
Bank Failures	16	18	18
U.S. Army Acquisition of Advanced Weapons Systems	2	3	4
International Military Equipment Sales	6	10	11
Fiber Optics Equipment Manufacturers	5	8	8
Total	71	114	114

5.3 Experiment C : Reduction of Dimension

FDS requires B+2 elements to be stored per word signal ($\frac{B}{2}+1$ elements for both magnitude and phase). CDS uses the DCT which produces real values, therefore only B elements need to be stored. This is still a large amount of data when we consider that the vector space methods only require that one element is to be stored. This is where the dimension reduction properties of the DCT are useful.

It is safe to assume that the CDS word signals are first order stationary Markov processes and hence the DCT is a good approximation of the KLT. Therefore, we should be able to perform a reduction of dimensionality and still obtain results comparable to those without the reduction. By performing the reduction, we do not have to store as much in the index and we do not have to perform as many calculations. Although the reduction may cause a degradation in the quality of results, it should be graceful due to the DCT approximation of the KLT. We performed experiments on the reduced data using both the long and short queries. The results can be seen in tables 4 and 5 respectively. We can see in both cases that the precision is reduced only by a small margin when the number of elements stored are reduced. Reducing the number of components has little effect on the precision of the top 20 documents for these ten short queries.

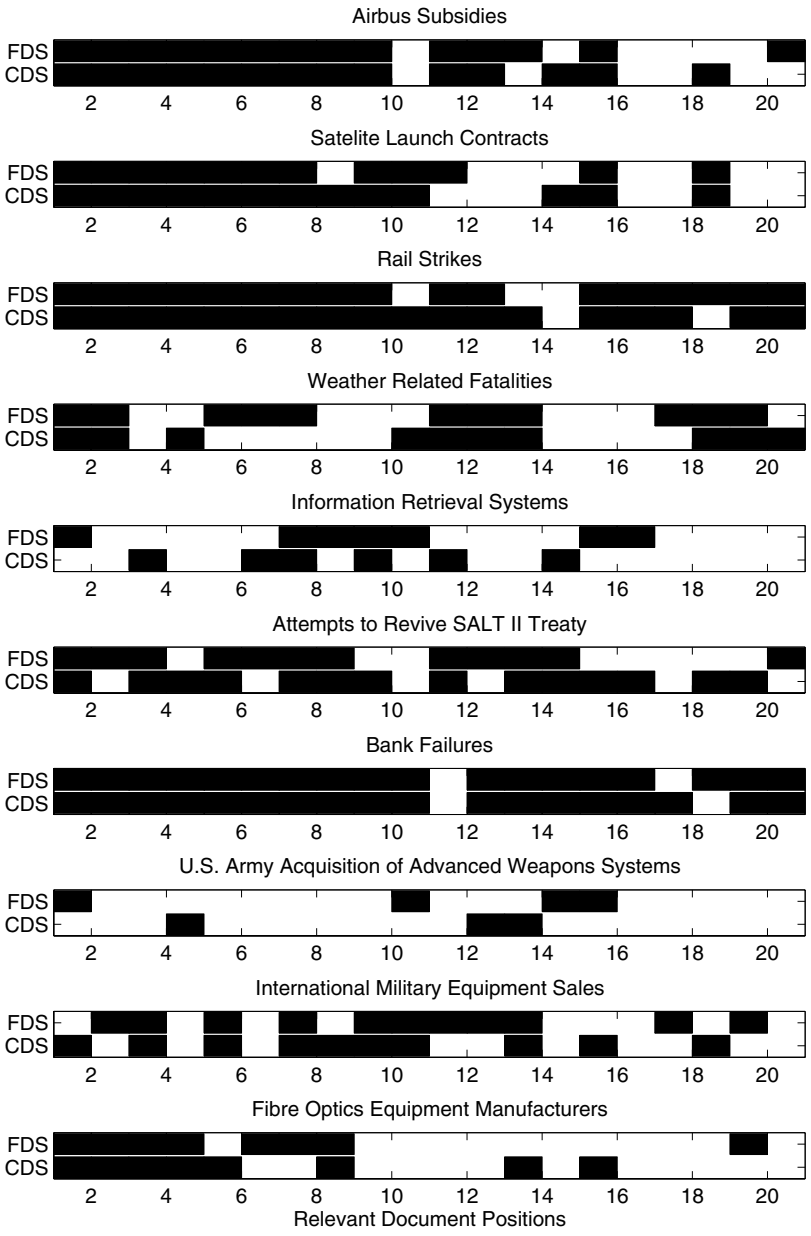

Fig. 3. This set of charts shows the positions of relevant documents from the queries in table 3. The documents are ranked with FDS 3.4.1 and CDS 2.2. A relevant document is identified by a black block. Both methods provide similar results

Table 4. Reduction of dimension results for long queries using CDS 2.2. The largest value in each column is shown in italics. The comp column refers to the number of components added to obtain the document score

Comp.	Precision at Recall					Average	R-
	0%	10%	20%	30%	40%	Precision	Precision
1	0.7082	0.6204	0.5568	0.5002	0.4433	0.3492	0.3728
2	0.7139	0.6197	0.5570	0.5052	0.4443	0.3572	0.3770
3	0.7221	0.6201	0.5616	0.5103	0.4429	0.3625	0.3749
4	0.7222	0.6200	0.5580	0.5117	0.4516	0.3664	0.3807
5	0.7270	0.6239	0.5688	0.5176	0.4564	0.3743	0.3961
6	*0.7414*	0.6408	0.5698	*0.5258*	0.4557	0.3769	0.3955
7	0.7385	*0.6421*	0.5695	*0.5258*	0.4597	0.3788	0.3978
8	0.7343	0.6420	*0.5767*	0.5228	*0.4648*	*0.3808*	*0.4026*

In some cases we can see that by choosing a smaller number of components, we obtain a higher precision. If we use only 2 components (20% of size of FDS) we still obtain a 40% improvement in precision over BD-ACI-BCA for short queries.

Table 5. Short queries applied to the AP2 document set using the reduced dimension CDS 2.2 method. The column Dx refers to the CDS 2.2 method using the first x components

Query term	Number of Relevant documents in top 20							
	D1	D2	D3	D4	D5	D6	D7	D8
Airbus Subsidies	10	11	12	12	12	13	14	14
Satellite Launch Contracts	13	14	14	14	14	14	14	13
Rail Strikes	16	17	17	18	18	18	19	18
Weather Related Fatalities	10	10	10	9	9	8	9	10
Information Retrieval Systems	6	6	7	6	7	7	7	6
Attempts to Revive the SALT II Treaty	4	5	7	9	10	12	12	14
Bank Failures	19	19	17	19	19	17	18	18
U.S. Army Acquisition of Advanced Weapons Systems	1	3	4	4	4	4	4	3
International Military Equipment Sales	8	9	8	9	9	11	11	10
Fiber Optics Equipment Manufacturers	5	8	8	8	8	8	8	8
Total	92	102	104	108	110	112	116	114

6 Conclusion

We have introduced the new method called Cosine Domain Scoring (CDS) which uses the Discrete Cosine Transform to perform document ranking. Since each

word signal can be classified as a first order stationary Markov process, the results further illustrate the fact that the DCT is a close approximation to the Karhunen-Loève transform (KLT).

Results were given for three different experiments. The first experiment showed that CDS 2.2 produced the most precise results for long queries out of the CDS methods given. The second showed that using CDS resulted in comparable results to those of FDS. The third experiment displayed that by reducing the dimension of the transformed word signals, we not only reduce the number of calculations and space needed to store the index, but we also produce results with approximately the same precision. The experiment showed that if only 2 components were used, we obtain precision 40% higher than that of BD-ACI-BCA and require only 20% of the storage needed by FDS.

From these experiments, we have concluded that replacing the DFT with the DCT gives us similar results by only using a fraction of the components. The DCT's relationship to the KLT has allowed us to obtain a deeper understanding of the components produced by the transform. This allows us to give results just as good as FDS, requiring fewer calculations, and allowing us to store the index in a more compact manner.

Acknowledgements

We would like to thank the ARC Special Research Centre for Ultra-Broadband Information Networks for their support and funding of this research.

References

1. N. Ahmed, T. Natarajan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 23:90–93, January 1974. 386, 387, 388
2. Okan Ersoy. *Fourier-Related Transforms, Fast Algorithms and Applications.* Prentice-Hall, Upper Saddle River, NJ 07458, 1997. 386
3. Laurence A. F. Park, Marimuthu Palaniswami, and Ramamohanarao Kotagiri. Internet document filtering using fourier domain scoring. In Luc de Raedt and Arno Siebes, editors, *Principles of Data Mining and Knowledge Discovery*, number 2168 in Lecture Notes in Artificial Intelligence, pages 362–373. Springer-Verlag, September 2001. 385
4. Laurence A. F. Park, Kotagiri Ramamohanarao, and Marimuthu Palaniswami. Fourier domain scoring : A novel document ranking method. *IEEE Transactions on Knowledge and Data Engineering*, Submitted February 2002. http://www.ee.mu.oz.au/pgrad/lapark/fds_compare3.pdf. 385, 386, 388, 389, 390, 391
5. William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C, The art of scientific computing.* Cambridge University Press, 2nd edition, 1997. 390
6. Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing gigabytes : compressing and indexing documents and images.* Morgan Kaufmann Publishers, 1999. 388

7. Justin Zobel and Alistair Moffat. Exploring the similarity space. In *ACM SIGIR Forum*, volume 32, pages 18–34, Spring 1998. 385