An Information Retrieval Model Based on Discrete Fourier Transform

Alberto Costa[1] and Massimo Melucci[2]

[1] LIX, École Polytechnique, F-91128 Palaiseau, France
costa@lix.polytechnique.fr
[2] Department of Information Engineering, University of Padua, I-35131 Padova, Italy
melo@dei.unipd.it

Abstract. Information Retrieval (IR) systems combine a variety of techniques stemming from logical, vector-space and probabilistic models. This variety of combinations has produced a significant increase in retrieval effectiveness since early 1990s. Nevertheless, the quest for new frameworks has not been less intense than the research in the optimization and experimentation of the most common retrieval models. This paper presents a new framework based on Discrete Fourier Transform (DFT) for IR. Basically, this model represents a query term as a sine curve and a query is the sum of sine curves, thus it acquires an elegant and sound mathematical form. The sinusoidal representation of the query is transformed from the time domain to the frequency domain through DFT. The result of the DFT is a spectrum. Each document of the collection corresponds to a set of filters and the retrieval operation corresponds to filtering the spectrum – for each document the spectrum is filtered and the result is a power. Hence, the documents are ranked by the power of the spectrum such that the more the document decreases the power of the spectrum, the higher the rank of the document. This paper is mainly theoretical and the retrieval algorithm is reported to suggest the feasibility of the proposed model. Some small-scale experiments carried out for testing the effectiveness of the algorithm indicate a performance comparable to the state-of-the-art.

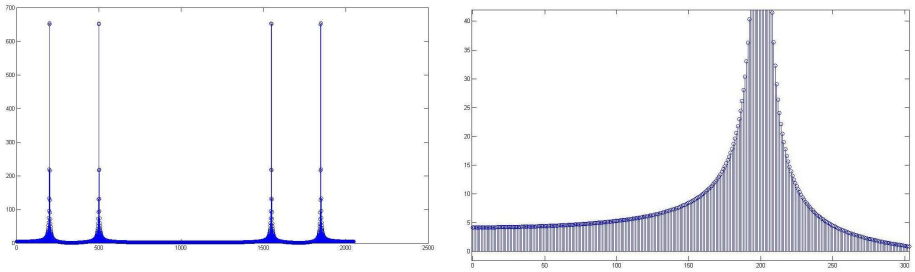
Keywords: Discrete Fourier Trasform, Digital Filters, Information Retrieval Models.

1 Introduction

Different views of Information Retrieval (IR) have been proposed and implemented in a series of models in the last four decades, the most famous being the logical models, the vector space models, and the probabilistic models. The variety of combinations of techniques originated from these models produced the significant increase in retrieval effectiveness observed within the Text Retrieval Conference (TREC) initiative [1]. However, the current retrieval technology is far from being optimal within every domain, medium, language or context. For this

reason, in addition to the numerous attempts to optimize and experiment the current retrieval models for different media, languages and contexts, the quest for new theoretical frameworks has been intense. [2, 3, 4, 5]

This paper illustrates a new framework which is based on Discrete Fourier Transform (DFT) and is called *Least Spectral Power Ranking* (LSPR). The LSPR framework provides the conceptual devices for designing and implementing an innovative class of retrieval systems. An intuitive description of the framework is provided in the remainder of this section while the other sections of the paper explain LSPR in detail.



(a) Spectrum for the query of the example. (b) Detail of the spectrum near point 200.

Fig. 1: Spectrum for a query (1a) and a detail (1b).

Basically in the LSPR model the query is viewed as a spectrum and each document as a set of filters, with one filter for each document term. Each term of the query is associated with a discrete time sinusoidal signal $A_i \sin(2\pi f_i nT)$, where A_i is the weight of the term i in the collection (e.g. IDF) and f_i is the signal frequency (not related with the frequency of a term in a document); an example is depicted in Figure 1a. How the signal frequency is chosen for each term is explained in Section 4.1. Thus, the query becomes a sum of sinusoidal signals for which the DFT can be computed. After the DFT module of the query has been computed, the spectrum looks like a curve which has peaks of amplitude proportional to A_i at the frequency f_i and the values of the curve monotonically decrease in the neighborhood of f_i as depicted in Figure 1b. The behaviour of the spectrum is crucial because the utilization of a sinusoidal signal combined with the variation in the amplitude of the spectrum causes variations in retrieval effectiveness (see Section 4.1 for more details) – the latter phenomenon has been observed in the experiments carried out during the research reported in this paper. In general, the DFT samples are complex numbers; computing the DFT module means substituting each complex sample with its module.

To obtain the ranked document list, the spectrum of the query is filtered by the documents, i.e. the filters. As a matter of fact, each document is represented as a set of filters, each of them is similar to the one in Figure 2: the breadth

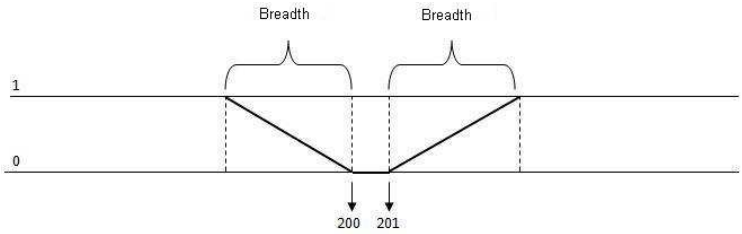
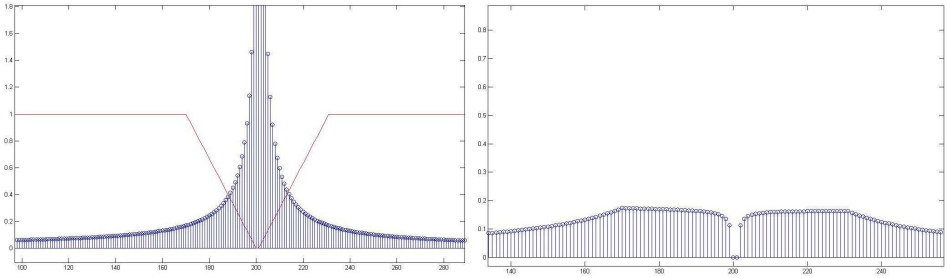


Fig. 2: Example of frequency response for the used filter.



(a) A fragment of a spectrum with a filter. (b) The obtained filtered spectrum.

Fig. 3: Representation of the filtering operation.

depends on the weight (e.g. TFIDF or BM25) of the term, while the position of the filter is set so that if a query has a term q_i, associated with a peak in the frequency f_i in the spectrum, and in the document there is q_i, the corresponding filter has ZL and ZR near f_i (as depicted in Figure 3a). The filtered spectrum is depicted in Figure 3b. The documents are then ranked by the power of the spectrum obtained as the sum of the components of the filtered spectrum. If a document is associated with a low power, it is considered highly relevant by the system, hence, the documents are ordered by increasing power. This is why the model is called *Least Spectral Power Ranking* (LSPR).

The paper is organized as follows. In Section 2 an overview of the main frameworks of the current IR models is presented. Section 3 presents the theory and the methods of DFT and digital filters, which is required to understand the rest of the paper, especially for those unfamilar with digital signal processing. Section 4 introduces LSPR. After that, Section 5 shows how LSPR works with an example (three documents and a query), while Section 6 suggests some future research directions.

2 Related Work

One of the quests in IR is whether a theory exists or not. The question is not without merit because if the answer were affirmative this discipline would have a solid framework in which the methodologies and experiments could be developed with the same efficacy as that of the models and experiments used in Physics to interpret the macro- or microscopic world. A model is the methodological instrument for formulating theories about IR. Currently, IR systems combine a variety of techniques stemming from logical, vector-space and probabilistic models. It is this variety of combinations investigated over the last four decades that has produced a significant increase in retrieval effectiveness in the last almost twenty years as reported in [1].

The models proposed in the literature aim at capturing some of the variables which affect the retrieval process. Although they are somehow complex, these models are however quite oversimplified because they deliberately ignore many other variables, especially those related to the user and to the context in which the end users operate. What the models proposed in the literature have in common is the use of a "metaphor" for describing them in a simplified manner, thus making their experimentation and dissemination possible.

The classical Boolean (or logical) model refers to set theory which views terms as document sets (operands) and queries as set operations [6] – non-classical logical models have been investigated by van Rijsbergen since 1980s [7]. The vector-space models view terms as basis vectors, documents and queries as vectors of the same space as conceived in the 1960s by Salton [8, 9] before being developed into Latent Semantic Indexing [10] which computes alternative basis vectors for expressing complex terms. The probabilistic models for IR refer to the theories of probability (there are many [11]) which view terms or documents as events (or random variables). There have been many contributions to the development of the probabilistic models for IR, to cite but a few [12, 13, 14, 15, 3, 5]. Specific classes of probabilistic models refer to the Bayesian networks [16] and more recently to statistical language models [17, 3] inspired by speech recognition. The recent book by van Rijsbergen attempts to unify the logical, vector-space and probabilistic models into a single framework – the one used to describe quantum mechanics [4] – which views documents and terms as vectors, retrieval operations as projectors and probability as a trace-based function.

The LSPR framework proposed in this paper differs from those proposed in the past because the underlying mathematical basis is different, yet there are some points in common. In our opinion, the main difference is the view through which the terms or in general the information content of the documents and of the queries are represented. In LSPR, the terms are sine curves which are a mathematical representation of signals. As these curves are functions, they can be treated algebraically. This property allows us to express queries and documents as functions which can be transformed for capturing salient features. In this paper, the DFT is the transform investigated.

The LSPR framework employs the mathematical constructs proposed in [18, 19] which report a complex framework for document retrieval. In particular, [19]

addresses the problem of matching queries and documents in which query terms occur at different positions and then with a variety of patterns. As matching documents by considering all the possible patterns is a difficult task, the authors propose to transform the spatial view of the query term occurrences into another domain. Therefore, each term of each document is represented as a signal and each document is divided into components. The term signals are transformed into term spectra which are a function of the magnitude and of the phase of the term occurring within each component of each document (magnitude and phase are then indexed by term, document and component). Document scores can be obtained by combining the term spectra components across terms under the assumption that a relevant document would have a high occurrence of query terms (implying a high magnitude of components) and a similar position of each pattern of query terms (implying similar phase). Unfortunately, this framework did not produce significant improvements over the standard TFIDF-based vector-space model. What makes LSPR different from those models is that the spectrum and filtering are computed on the query and that the documents are the filters. Moreover, LSPR models the query terms as signals independently of the problem addressed in [19], the latter being a specific problem of text retrieval. Indeed, LSPR aims at making the sinusoidal signals a modeling paradigm (as the vector spaces, for example, are the paradigm of the vector-space model) rather than the approach of addressing the problem addressed in [19].

3 Background

3.1 Discrete Fourier Transform

The Fourier transform is essential in digital signal processing, because it allows us to analyze signals in the frequency domain in a more effective and efficient way than the time domain analysis [20, 21]. For continuous time signals the Fourier transform is employed, but if the signals are periodic the Fourier transform can be replaced with the Fourier series. In the discrete time domain, there is something similar and the transform related to discrete time periodic signals is the Discrete Fourier Transform (DFT). DFT is the only transform that can be computed numerically. Our interest in DFT derives from the need to analyze periodic sinusoidal discrete signals in the frequency domain. To this end the Fast Fourier Transform (FFT) is an $O(N \log N)$ algorithm [22, 23]. This approach is a "divide-and-conquer" algorithm and was conceived by Cooley and Tuckey in 1965.[3]

In the following, the DFT is more formally introduced. Let $x(nT)$ be a discrete time signal sample where T represents the distance between two samples. The DFT of this signal is

$$\tilde{X}(k) = T \sum_{n=0}^{N-1} x(nT)\, e^{-\mathrm{i}2\pi kFnT} \qquad F = \frac{F_c}{N} \tag{1}$$

[3] Actually, the first who spoke of this algorithm was Gauss, in 1805.

where $\mathrm{i} = \sqrt{-1}$, F_c is called sampling frequency, N is the number of samples and k identifies the k-th sample of the DFT. In the remainder of this paper, $F_c = 1/T$, therefore $F = \frac{1}{NT}$. In this way, Eq. 1 is

$$\tilde{X}(k) = T \sum_{n=0}^{N-1} x(nT)\, e^{-\mathrm{i}\frac{2\pi n}{N}k}. \tag{2}$$

In order to obtain the power of the spectrum, the module of Equation 2 has to be computed from the complex numbers of the DFT. When $x(nT)$ is the sampling of a real periodic function, the module of the DFT is symmetric with respect to $N/2$, hence, we only need to operate with half of the spectrum.

3.2 Spectral Leakage

As stated earlier, the signals used in LSRP are supposed to be periodic, sinusoidal and discrete. The general form for these signals is

$$A_i \sin(2\pi f_i nT). \tag{3}$$

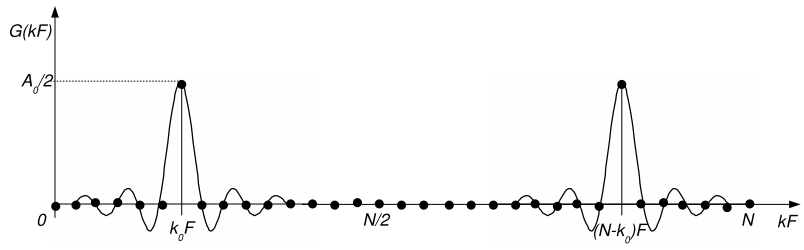
The module of the DFT for the signal (3) has a peak proportional to amplitude A_i at the frequency f_i (see Figure 4a), but if the frequency f_i of the signal is not a multiple of F, a distorted DFT is obtained; Figure 4b shows this phenomenon, which is called *spectral leakage*. Usually, in digital signal processing spectral leakage is avoided because it makes the recognition of the correct frequencies of the signals difficult.

In contrast in the LSPR model, spectral leakage is kept because the presence of some components with reduced amplitude at the frequencies close to f_i makes it possible to better estimate the relevance of a document – the reason will be more clear in Section 4.1 – and introduces some constraints in the choice of F and f_i for the signals (3).

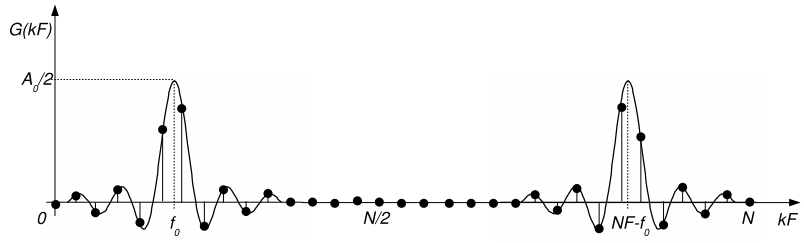
One of the most important theorems in digital signal processing is the sampling theorem (or Whittaker-Nyquist-Kotelnikov-Shannon theorem). There are a lot of equivalent definitions for this theorem. In this paper, the following is used:

Theorem 1 (Sampling theorem [20]). *Let $x(t)$ be a signal with no frequencies higher than f_M Hertz. Then $x(t)$ is uniquely determined by its samples $x(nT)$ if $F_c \geq 2f_M$.*

This theorem is important in the framework proposed in this paper because it gives some conditions for the choice of the frequencies f_i of the sinusoidal signals (3), and for the choice of the parameters of DFT. This will be explained better in Section 4.



(a) DFT of sinusoidal signal with f_i $(i = 0)$ multiple of F.



(b) DFT of sinusoidal signal with f_i $(i = 0)$ not multiple of F.

Fig. 4

3.3 Digital Filtering

One of the most important subjects in digital signal processing is digital filter design. A digital filter is a component that transforms an input signal $x(nT)$ to an output signal $y(nT)$ through the convolution of the input by the filter. Convolution is quite difficult to compute if the Fourier transform is not used. In contrast, after DFT has been applied to the signals and the filter, the computation is feasible. Let $\tilde{X}(k)$, $\tilde{Y}(k)$ and $\tilde{H}(k)$ be, respectively, the DFT of the input signal, the DFT of the output signal and the DFT of the filter; the relationship among these transforms can be written as:

$$\tilde{Y}(k) = \tilde{H}(k)\tilde{X}(k). \tag{4}$$

This equation performs filtering in the LSPR model with a simple multiplication between two functions.

There are a lot of filters which could be used in this context. In this paper, we used something similar to a *notch* filter which eliminates the spectrum near a predefined frequency (called *cut-off* frequency), and keeps the spectrum in corrispondence with the other frequencies unchanged. The design of a filter appropriate for IR is still under investigation. Figure 5 shows an example of the module for a notch filter.

Actually, in our tests we used a simplified filter, that is, a filter with a triangular form. This is better explained in Section 4.

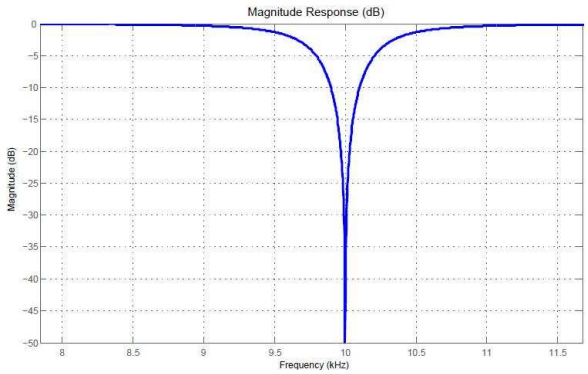


Fig. 5: Module of the transform for a notch filter with cut-off frequency of 10 kHz.

4 Least Spectral Power Ranking

In this section Least Spectral Power Ranking (LSPR) which implements the DFT-based framework is introduced. LSPR consists of three main steps:

1. transformation of a query into a spectrum,
2. transformation of a document into a notch filter set,
3. document ranking.

Before these three main steps some preprocessing is performed. The preprocessing phase includes all the operations usually done by every IR system, such as stemming and stopwords removal. In this first implementation of the model, we used the well-known TF-IDF weighting scheme for the documents in the collection. Actually, we used the normalized version of TF-IDF: if we call w_{ij} the TF-IDF weight for the term i in the document j, the normalized version is

$$\hat{\mathrm{w}}_{ij} = \frac{\mathrm{w}_{ij}}{\sqrt{\sum_i \mathrm{w}_{ij}^2}}. \tag{5}$$

This expression has been used to implement the strength of the filter. The IDF weighting scheme has been used for implementing the amplitude of the peaks in the spectrum. Other weighting schemes can be used as well.

4.1 Query to Spectrum Transformation

The first step is to set the parameters of the DFT (such as F and N) and of the sinusoidal signals. We choose to set $F = 2$ Hz, and to set every frequency of the signals (3) as an odd number: in this way it is not possible for a frequency to be an integer multiple of F, and this guarantees the spectral leakage. This choice

allows us to write equation (3) in another way. Recalling that $T = \frac{1}{NF}$:

$$A_i \sin(2\pi f_i n T) = A_i \sin\left(\frac{\pi f_i n}{N}\right). \tag{6}$$

Before explaining how we choose the frequency of each signal, the parameter N has to be set. It is important to remember that the spectrum is associated with the query, and that for each query term there is a sinusoidal signal. In this way the query is a sum of sinusoidal signals. The DFT of this sum is computed using N samples (N is both the number of samples of the sinusoidal signals used as input for the DFT and the number of points of the spectrum). For each term of the query there are 300 points of the spectrum and the peak of the relative signal is near point 200. These values are chosen because some experiments showed that the amplitude of the spectrum was less than 1% of the peak value 100 points after and before the peak. The use of 300 points assures that each sinusoidal signal is related to a disjoint interval of the spectrum; for example, if the query has 2 terms, there are $2 \times 300 = 600$ points. The number of points is the same for each term. Methods for choosing the number of points depending on the terms will be studied in the future. The FFT algorithm requires that this value is rounded at the closest power of two, and that this value has to be doubled due to the symmetry of the DFT (this last operation also has the advantage of fulfilling the conditions of Theorem 1); for example, 600 is rounded to 1024 and finally $N = 2048 = 1024 \times 2$. $F_c = NF = 2048 \times 2 = 4096$ Hz, where the maximum frequency f_M among the sinusoidal signals is 1001 Hz, as shown below, so the condition of Theorem 1 ($F_c \geq 2f_M$) is verified.

In order to have the peak of the sinusoidal signal near to point 200 of the relative 300-point interval and to guarantee spectral leakage, its frequency is set to

$$(300 \cdot (i-1) + 200)\, F + 1 \text{ Hz} \tag{7}$$

where i refers to the i-th term of the query. Finally, the amplitude of the sinusoidal signal is the IDF weight of the term.

Now we can show with an example how this method works. Suppose there is a query with 2 terms; for the sake of clarity, suppose also that the amplitudes of the two original terms are equal, and we call this value A. When $N = 2048$, the frequencies of the 2 signals are, respectively, 401 Hz and 1001 Hz using formula (7). The input for the FFT algorithm is a N-points vector S, and it is calculated as

$$S[n] = A \sin\left(\frac{401\pi n}{2048}\right) + A \sin\left(\frac{1001\pi n}{2048}\right), \qquad n = 1, 2 \ldots, N. \tag{8}$$

The FFT algorithm returns the DFT of this vector and the module is something similar to Figure 1a. In this figure we can see that the spectrum is symmetric with respect to $N/2$ (point 1024, which corresponds to 2048 Hz), and the peaks are near the points 200 and 500. Figure 1b shows more clearly the spectrum near point 200. Another important aspect showed by Figure 1b is the side

effect of the spectral leakage. The decrease in importance follows the decrease in amplitude of the spectrum: this is an important hypothesis, and the efficacy of the presented model is based on it. Without spectral leakage, the figure would show a single peak at point 200 and the function would be zero at the other points, removing the decrease in the amplitude and making it impossible to estimate the decrease in importance, thus making futile the effect of the filtering operations.

4.2 Document to Filter Transformation

Each document is transformed into a filter set (one for each document term). As stated before, in LSPR, each filter is similar to a "notch" with triangular form. To be precise, there are two points, ZL (*Zero Left*) and ZR (*Zero Right*), where the module is 0, and there is a parameter called *breadth* such that the value of the filter before ZL−breadth and after ZR+breadth is 1. Finally, to connect ZL (ZR) with ZL−breadth (ZR+breadth) points, there is a linear function. An example of this filter is represented in Figure 2, where Figure 3 shows a fragment of a spectrum with a filter, and the obtained filtered spectrum.

Two issues needs some explainations: the selection of the breadth (which represents the strength of the filter) and the position of the filter.

The former is the weight TF-IDF of the document term normalized by the document length ($\hat{w}_{ij}$), multiplied by a constant called *selectivity*. The value of this constant was set to 24, because some experiments showed that for values greater or smaller than 24, the MAP decreased. Probably, this value depends on the collection, but the interesting fact is that 24 is a sort of maximum for a function that relates selectivity and MAP. This reminds us of BM25 thus suggesting some possible future research directions.

The latter is computed as follow: If the term was occurring in the query, ZL and ZR were points 200 and 201 of the relative interval; this means that for example if this term is the second of the query, ZL=500 and ZR=501, while ZL=200 and ZR=201 for the first query term, as in Figure 2. Otherwise, the filter is not created.

4.3 Document Ranking

Suppose a query is given as input to the system by the user and consider the set of documents retrieved by the system; they are reranked by a score computed after applying Equation (4). As a matter of fact, the module obtained by the query-to-spectrum transformation is filtered by the filter set obtained after the document-to-filter transformation phase. Finally, the score obtained by each document is the sum of the module of the spectrum (the "power") after filtering, hence, every document is associated with a score. The documents are ordered by increasing power because the more the filters decrease the power of the spectrum, the more the system consideres the document and the query related. Figure 6 summarizes LSPR.

Algorithm: LSPR
Input: query Q, collection of documents
Output: ranked document list

```
1  Compute the vector S, as described in Section 4.1
2  spectrum ← |DFT(S)|
3  I ← [] (initialize I)
4  ZL ← [] (initialize ZL)
5  for each term t of the query Q
6       do
7          Retrieve the posting list of t and call it L(t)
8          I ← [I, index of t − 1]
9          ZL ← [ZL, 300 · (index of t − 1) + 200]
10         for each X ∈ L(t)
11              do
12                 if it is the first time that document X is selected
13                   then
14                        f_spectrum ← spectrum
15                 end if
16                 for i = 1 to |I|
17                      do
18                         breadth ← Round(selectivity · weight of q_i in X)
19                         f_spectrum ← Filter(f_spectrum, ZL[i], breadth, I[i] − 1)
20                 end for
21                 power[X] ← Σ_{k=1}^{|f_spectrum|} f_spectrum[k]
22         end for
23 end for
24 Order the documents by increasing power and save the list in result
25 return result
```

Fig. 6: An algorithm to summarize LSPR.

In this algorithm, the notation $X \leftarrow []$ represents the initialization of a vector, while $X \leftarrow [X, i]$ is used to add a new element i to the vector X. Basically, the rows from 1 to 4 represent the initialization phase; the rows from 5 to 9 set the values of ZL for the filters (as explained in Section 4.2), and the rows from 10 to 22 describe the filtering operation. At the end, the power is computed and the algorithm returns the ranking list. The Filter function used within the algorithm has as inputs the spectrum, the ZL point of the filter, the breadth of the filter and the index i of the term of the query, whereas it returns as output the filtered spectrum. Basically, this function does the multiplications of the module of the spectrum (e.g. that in Figure 1a) by the frequency response of the filter (e.g. that in Figure 2), as shown in Figure 3.

4.4 Some Preliminary Results

LSPR was tested by using the CACM test collection and compared with two baselines:

- A basic vector-space model.
- The Divergence From Randomness (DFR) model implemented by Desktop Terrier v.1.1.

After stopword removal and stemming, the retrieved list were measured by `trec_eval`. The Mean Average Precision (MAP) of the vector-space model was 0.242, while the MAP of DFR was 0.329. The MAP of LSPR was 0.348, thus indicating a performance comparable to the state-of-the-art.

5 Example

To give a global idea about how LSPR works, in this section we present an example with a collection of three documents and a query.

Suppose we have the following documents in the collection (for the sake of clarity, we consider them after stopword removal, and without stemming):

- D1 : (retrieval, data, author, book);
- D2 : (information, computer, system, storage, information, data);
- D3 : (information, retrieval, system, relevance, MAP, precision, recall, relevance).

Suppose also that the query is Q : (information, retrieval, relevance).

Table 1 represents the index of the collection, with the normalized TF-IDF weighting schema, and the query with the IDF weighting schema.

	D1	D2	D3	Q
author	0.663	0	0	0
book	0.663	0	0	0
computer	0	0.596	0	0
data	0.245	0.220	0	0
information	0	0.440	0.136	0.585
MAP	0	0	0.367	0
precision	0	0	0.367	0
recall	0	0	0.367	0
relevance	0	0	0.735	1.585
retrieval	0.245	0	0.136	0.585
storage	0	0.596	0	0
system	0	0.220	0.136	0

Table 1: Normalized TF-IDF index matrix and IDF weights for query

At this point, we show how LSPR creates the spectrum of the query, as described in section 4.1. First, the frequencies of the terms "information", "retrieval" and "relevance" are set respectively to 401 Hz, 1001 Hz and 1601 Hz, while N is 2048. Now, the vector S is computed as

$$S[n] = 0.585 \sin\left(\frac{401\pi n}{2048}\right) + 0.585 \sin\left(\frac{1001\pi n}{2048}\right) + 1.585 \sin\left(\frac{1601\pi n}{2048}\right) \quad (9)$$

where $n = 1, 2 \ldots, N$. The vector S is given as input for the FFT algorithm, and we obtain the spectrum shown in Figure 7 (actually, due to the symmetry, we consider only half of the spectrum).

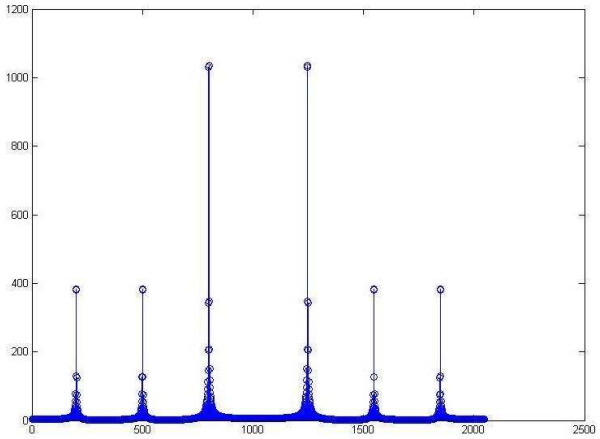
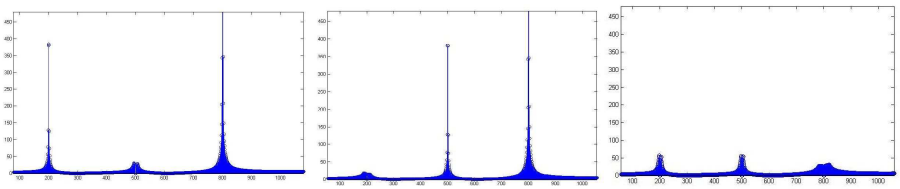


Fig. 7: Spectrum for the query

Now, the documents are transformed into filters, as explained in section 4.2. Since D1 contains only the term "retrieval" of the query, with weight 0.245, it is transofmed into a filter with ZL=500 (which corresponds to the peak of the term "retrieval" in the spectrum) and with breadth=Round$(24 \cdot 0.245)$= 6. Similarly, D2 corresponds to a filter ("information") with ZL=200 and breadth=Round$(24 \cdot 0.440)$=11. Finally, D3 is transformed into a set of three filters; the first one ("information") has ZL=200 and breadth=Round$(24 \cdot 0.136)$=3, the second one ("retrieval") has ZL=500 and breadth=Round$(24 \cdot 0.136)$=3, and the third one has ZL=800 and breadth=Round$(24 \cdot 0.735)$=18. At the end, after the filtering operations on the spectrum represented in Figure 7, we obtain the filtered spectra of Figure 8.

The power (sum of the components of the spectrum) of the spectrum of Figure 7 is 13007.091, while the power of the spectra of Figure 8 are respectively 11836.613, 11649.498 and 6919.414; thus, according to the ranking rule of LSPR, the first document retrieved is D3, the second is D2 and the third is D1.



(a) D1 (b) D2 (c) D3

Fig. 8: Spectra after the filtering operations performed on the three documents.

6 Conclusions and Future Work

This paper describes an IR model, called LSPR, that uses DFT. The paper is mainly theoretical because the focus of the paper was on the model whereas future work will focus on large scale experimentation. Nevertheless, some experiments were performed on a small scale. Preliminary results show that this model works well, and its efficacy is comparable with the efficacy of the state-of-the-art.

LSPR was a first implementation of the framework based on DFT and a variety of implementations can be obtained through the series of parameters illustrated in the paper. It is our opinion that there is large room for improvement by setting appropriate values for sample sizes, thresholds and frequencies. At first glance, such an approach may be difficult, but the sound mathematical framework of DFT may provide useful guidance and significant improvements.

Future work has three main objectives. First, efficiency needs to be improved. In this first implementation, since the effectiveness evaluation of the model was our focus, efficiency was not in the priority list. In order to test the model with larger collections these aspects are very important, hence, an optimization of the algorithm is necessary.

Second, we have to look for better configuration than those reported in this paper. To obtain this, we will tune the parameters of the model. For example, in this implementation we use the TF-IDF weighting scheme (and IDF for the query), but other schemes such as Okapi BM25 can be tested. Another important issue is the similarity between the terms: using some clustering algorithm [24], we can estimate how two terms are related; these informations can be used to increase the efficacy of the model, for example by associating other different filters with the similar terms. In addition, the internal parameter of LSPR, such as the number of points of DFT and the selectivity will be investigated.

Finally, the other feature of LSPR is that the functions are defined over the complex field, the latter being a characteristic of Geometry of IR by van Rijsbergen [4] in which no restrictions were placed on the scalars. The use of the complex field is a question of representation and many operations in that framework are always real (thus permitting ranking). The fact that scalars are complex and the descriptors are represented as sine curves, i.e. signals, helps enlarge the research in IR to other media than text by leveraging the extra representation

power given by complex numbers. Moreover, the functions used in the Fourier transform are mutually orthogonal. This property recalls the techniques developed within the vector space models (e.g. Latent Semantic Analysis) and the framework developed by [4]. Further investigation will then be carried out on these connections.

If the model leads to good results with larger experimental collections, we can test it in other contexts, such as web information retrieval. Actually, we are working on a modified version of LSPR for the recommender systems.

Acknowledgements

We would like to thank Alberto Caccin, Albijon Hoxaj, Marco Lonardi, Enrico Martinelli and Giuseppe Soldo for the tests with Terrier. Moreover, we want to thank Emanuele Di Buccio for his precious suggestions.

One of the authors (A. C.) is grateful to Digiteo Project 2009-55D "ARM" for financial support.

References

[1] Voorhees, E., Harman, D., eds.: TREC: Experiment and Evaluation in Information Retrieval. The MIT Press, Cambridge, MA, USA (2005)
[2] Robertson, S.: Salton award lecture: On theoretical argument in information retrieval. SIGIR Forum **34**(1) (2000) 1–10
[3] Croft, W., Lafferty, J., eds.: Language Modeling for Information Retrieval. Springer (2003)
[4] van Rijsbergen, C.: The Geometry of Information Retrieval. Cambridge University Press, UK (2004)
[5] Fuhr, N.: A probability ranking principle for interactive information retrieval. Journal of Information Retrieval **11**(3) (2008) 251–265
[6] Cooper, W.: Getting beyond Boole. Information Processing & Management **24** (1988) 243–248
[7] van Rijsbergen, C.: A non-classical logic for Information Retrieval. The Computer Journal **29**(6) (1986) 481–485
[8] Salton, G.: Automatic information organization and retrieval. Mc Graw Hill, New York, NY (1968)
[9] Salton, G.: Mathematics and information retrieval. Journal of Documentation **35**(1) (1979) 1–29
[10] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41**(6) (1990) 391–407
[11] Fine, T.: Theories of probability. Academic Press (1973)
[12] Maron, M., Kuhns, J.: On relevance, probabilistic indexing and retrieval. Journal of the ACM **7** (1960) 216–244
[13] Robertson, S., Sparck Jones, K.: Relevance weighting of search terms. Journal of the American Society for Information Science **27** (May 1976) 129–146
[14] Robertson, S.: The probability ranking principle in information retrieval. Journal of Documentation **33**(4) (1977) 294–304

[15] Robertson, S., Walker, S.: Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), Dublin, Ireland (1994) 232–241
[16] Turtle, H., Croft, W.: Inference networks for document Retrieval. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), Brussels, Belgium (September 1990)
[17] Ponte, J., Croft, W.: A language modeling approach to information retrieval. In: Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR), Melbourne, Australia, ACM Press, New York (August 1998) 275–281
[18] Park, L.A.F., Ramamohanarao, K., Palaniswami, M.: Fourier domain scoring: A novel document ranking method. IEEE Trans. on Knowl. and Data Eng. **16**(5) (2004) 529–539
[19] Park, L.A.F., Ramamohanarao, K., Palaniswami, M.: A novel document retrieval method using the discrete wavelet transform. ACM Trans. Inf. Syst. **23**(3) (2005) 267–298
[20] Oppenheim, A.V., Willsky, A.S., Nawab, S.H.: Signals & systems. Second edn. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1996)
[21] Mitra, S.K.: Digital Signal Processing: A Computer-Based Approach. Third edn. McGraw-Hill, New York (2006)
[22] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms. Second edn. MIT Press, McGraw-Hill Book Company (2000)
[23] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes in C: the art of scientific computing. Second edn. Cambridge University Press (1992)
[24] Croft, B., Metzler, D., Strohman, T.: Search Engines: Information Retrieval in Practice. First edn. Addison Wesley (2009)