Can Vector Space Bases Model Context?

Massimo Melucci

University of Padua
Department of Information Engineering
Via Gradenigo, 6/a – 35031 Padova – Italy
melo@dei.unipd.it

Abstract Current Information Retrieval models do not directly incorporate context, which is instead managed by means of techniques juxtaposed to indexing or retrieval. In this paper, direct modeling of context is addressed by proposing the full recover of the notion of vector space base, whose role has been understated in the formulation and investigation of the Vector Space Model. A base of a vector space models document or query content descriptors. Moreover the semantics of content descriptors depends on context. Therefore a *base* of a vector space is the construct to model *context*. Also, change of context can be modeled by linear transformations from one base to another. Matrix algebra can thus by employed to process context, its evolution and application. In particular, Relevance Feedback is shown to be an example of context change.

1 Introduction

Information Retrieval (IR) deals with the retrieval of all and only the documents which contain information relevant to any information need expressed by any user's query. A system matching that definition exists in principle. In practice a system is unable to answer any query with all and only relevant information because of the unsolvable task of understanding both the relevant information enclosed in documents and the information need expressed through any query submitted by any user.

The difficulty of IR is basically due to the fact that relevance is context-dependent, i.e. the information need evolves with the user, place, time, and other unobservable variables. Therefore, IR *is* context-dependent, and addressing context is fully recognizing the high complexity of IR.

In the early days of IR, some technology constraints made the task easier than the one should have to solve if context were considered instead. Indeed, the interaction paradigm usually understood in the past, and actually by current search engines as well, assumes one type of user placed in one context of which query is the single expression – the notion of location, time and other variables was simply not considered.

As consequence, classical models were defined by assuming that there is one user, one information need for each query, one location, one time, one history, one profile. A common approach to face IR in context has been based on ad-hoc techniques designed to capture time, space, histories, or profiles injected into models. The availability of sensors to get space location, log-files to implement history, metadata to describe profiles, clocks and calendar to get time makes this injection feasible. However, these data are only juxtaposed to models which remain independent of any notion of context.

The only variation considered in that “monolithic” theme was relevance feedback – as an information need could evolve after the user have seen and assessed some documents answering to a query, the automatic modification of the query could be performed by the system by using the feedback collected from the user who assessed the relevance of the seen documents. This way, the modified query is “closer” to relevant documents, and then more likely to retrieve other relevant documents, provided that the hypothesis that relevant documents are close one to each other.

In this paper it is believed that an important problem hampering the full development of IR in context is the lack of direct representation of the latter in the models. Thus, how to combine and integrate context in models in a way it is naturally described? A positive answer to this question would allow a direct modeling of the way time, space, histories, or profiles act on indexing and retrieval, and then would allow to intervene to set-up more effective systems.

The research reported in this paper has stemmed from the reflections on the use of the Vector Space Model for the retrieval of semantically annotated documents reported in [3] – a semantically annotated document contains key words labeled by classes from ontologies about their domain. From that research, it has been recognized the need of representing different meanings for a key word depending on the document that contain the key word. This paper has been also supported by reading [6], whereas the discovery of the concepts underlying the documents represented by the Vector Space Model was introduced in [2].

The paper has been structured as follows: The approach is described in Section 2. The Vector Space Model has been introduced in Section 3, whereas Section 4 describes a possible user-system interaction scenario which leads to the notion of context as interpreted in this paper. The illustration of the modeling approach is reported in Section 5, and Section 6 illustrates a notable example.

2 Approach

This paper investigates a classical model here adopted as a means of coping with (*i*) context in representation of documents and queries, and (*ii*) the resulting difficulties in determining the relevance of a document relative to a given query as context changes. Before developing yet another model it is believed that the classical models have left some reflections and intuitions still undeveloped despite decades of research.

The capabilities in modeling context of the Vector Space Model (VSM) for IR are investigated in this paper. That model gives an intuitive yet formal view of indexing and retrieval – it is a matter of fact that it attracted many researchers and newcomers of IR after it was introduced in [4] and [5]. Beside its seeming simplicity, it proved very effective in retrieving documents of different languages, subjects, size, and media. A number of weighing schemes and application were proposed and tested thus making it a sound framework.

Nevertheless, the VSM has not been fully exploited in practice. An early attempt done to reevaluate it is reported in [7], yet the notion of context was completely ignored. A recent reconsideration was done in [6] by placing it in one mathematical framework.

Despite extensive study and experimentation, there are some potentialities to which further study should be deserved to the VSM. In particular, there is an issue stated below which was taken little seriously in the past, but looks promising, that is why it is addressed in this paper:

> A base of a vector space models document or query content descriptors. Moreover the semantics of content descriptors depends on context. Therefore a *base* of a vector space is the construct to model *context*. Also, change of context can be modeled by linear transformations from one base to another. Matrix algebra can thus by employed to process context, its evolution and application.

3 The Vector Space Model

Let V be a vector space in $\mathbb{R}^n$ and $T = \{\mathbf{t}_1, \ldots, \mathbf{t}_k\}$ be a set of k column vectors in V. The matrix $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_k]$ is the matrix corresponding to T. A set of vectors T is independent if $\sum_{i=1}^{k} x_i \mathbf{t}_i = \mathbf{0}$ only if $x_i = 0$ for every i. If $k = n$ and T is independent, T is a *base* for V. A base for V generates all the vectors of V, that is,

$$\mathbf{v} = \sum_{i=1}^{n} w_i \mathbf{t}_i = \mathbf{T} \cdot \mathbf{w}$$

for all $\mathbf{v} \in V$, where $\mathbf{w} = [w_1, \ldots, w_n]^\top$ are called coefficients.[1]

When a collection of documents is indexed, the IR system yields a set of unique descriptors $\{t_1, \ldots, t_n\}$. It is worth noting that this representation is valid apart from the media or the languages used to implement the documents. In this model, the set of unique descriptors is modeled as a base T. Every object, i.e. documents, fragments, or queries are modeled as vectors generated by T, that is

$$\mathbf{x} = \sum_{i=1}^{k} x_i \mathbf{t}_i = \mathbf{T} \cdot \mathbf{x}$$

where $\mathbf{x}$ models the object x, $\mathbf{t}_i$ models t_i and w_i is the coefficient of $\mathbf{t}_i$ in generating $\mathbf{v}$.

The retrieved documents are ranked by the inner product between query vectors and document vectors, that is by

$$\mathbf{d} \cdot \mathbf{q} = \mathbf{a}^\top \cdot (\mathbf{T}^\top \cdot \mathbf{T}) \cdot \mathbf{b}$$

where $\mathbf{q}$ is the vector generated by T and modeling a query q.

3.1 Remarks

It should be noted that the notion of orthogonality is stricter than the one of independence. Whereas orthogonal base vectors are independent one of each other, the converse

[1] $\mathbf{x}^\top$ means "transpose of" $\mathbf{x}$.

is not true – actually, one can build an independent base which is not orthogonal. For example, the set of versors $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ of $\mathbb{R}^3$, i.e. $\{[1,0,0]^\top\ [0,1,0]^\top\ [0,0,1]^\top\}$ are orthogonal and independent. Indeed, if the versors are organized as a matrix, the identity matrix $\mathbf{I}$ is obtained. To compute all the inner product, the matrix product $\mathbf{I}^\top \cdot \mathbf{I} = \mathbf{I} \cdot \mathbf{I} = \mathbf{I}$ is computed by thus showing that the inner product $\mathbf{e}_i^\top \cdot \mathbf{e}_j = 0$ if $i \neq j$ and 1 if $i = j$, the latter showing that the versors are orthogonal.

Let us show that the orthogonal set of versors is also independent. By absurd, let the set of versors be dependent, i.e. $\sum_i c_i \mathbf{e}_i = \mathbf{0}$ with the c_i's not all null. After multiplying both sides by $\mathbf{e}_1$, and recalling that $\mathbf{e}_1 \cdot \mathbf{e}_j = 0$ for $j > 1$, one can see that $\sum_i c_i \mathbf{e}_i = c_1(\mathbf{e}_1 \cdot \mathbf{e}_1) = 0$. But, $\mathbf{e}_1 \cdot \mathbf{e}_1 = 1$ since $\mathbf{e}_1 \neq \mathbf{0}$; therefore c_1 must be zero. After repeating the same procedure by using every $\mathbf{e}_i$ instead of $\mathbf{e}_1$, one can see that every c_i must be zero by thus showing that $\sum_i c_i \mathbf{e}_i = \mathbf{0}$ only if the c_i's all are null.

Let us now show that an independent set is not necessarily orthogonal, by using the following counterexample; the set of vectors $\{[3,1,1]^\top, [1,4,2]^\top, [1,3,5]^\top\}$ is independent yet not orthogonal.

In many applications of the VSM, the base vectors are assumed as orthogonal, and in particular they correspond to the versors of V. This means that every document or query vector $\mathbf{x}$ is generated by the set of versors, that is, $\mathbf{x} = \sum_{i=1}^k x_i \mathbf{t}_i = \sum_{i=1}^k x_i \mathbf{e}_i = [\sum_{i=1}^k x_i e_{i1}, \ldots\ \sum_{i=1}^k x_i e_{in}]^\top$. Since the j-th element of $\mathbf{e}_i$, i.e. e_{ij} is 1 if $i = j$ and 0 otherwise, $\mathbf{x} = [x_1 e_{i1}\ \ldots\ x_n e_{in}]^\top = [x_1\ \ldots, x_n]^\top$. As consequence, a document or query vector is a vector of coefficients. This is the reason why what has affected the effectiveness of the VSM in the experiments carried out in the past has been the choice of the coefficients, whereas the base has been ignored.

From these two facts, it is apparent that the notion of base vector is distinct from the one of coefficient. The latter is used to generate object vectors by combining base vectors, which represent descriptors. The distinction between the notion of coefficient and the one of base vector allows to associate multiple bases to the same set of coefficients, by thus decoupling them in modeling documents or queries. For example, let t_1, t_2, t_3 be three descriptors extracted from a document collection – these descriptors may be, say key words. Let $x_1 = 0.1, x_2 = 3.5, x_3 = 2.4$ be the weights computed for t_1, t_2, t_3, respectively, for a document d. To generate the vector $\mathbf{d}$ representing d, the set of versors can be used and $\mathbf{d} = \sum_i x_i \mathbf{e}_i = [0.1\ 3.5\ 2.4]^\top$ is obtained. If the independent set of vectors $\{\mathbf{t}_1 = [3,1,1]^\top, \mathbf{t}_2 = [1,4,2]^\top, \mathbf{t}_3 = [1,3,5]^\top\}$ were used instead, $\mathbf{d}$ would be $\sum_i x_i \mathbf{t}_i = [6.2\ 21.3\ 19.1]^\top$, which is a different representation of the same document.

By exploiting the difference between coefficient and descriptor, the approach taken in this paper is to leverage base vectors rather that coefficients as illustrated in the next Sections.

4 Scenario

Let a user be either a searcher accessing the system to retrieve documents relevant to his information need, or an author developing his documents. The user employs descriptors to express the semantic content of the documents or of the queries depending on whether he is an author or a searcher, respectively. The capability of a descriptor in

expressing the semantic content of documents or queries is given by its own semantics and relationships with other descriptors.

The descriptors are selected by the user on the basis of their own capabilities in expressing the semantic content of the documents and by their inter-relationships. Of course, it is assumed that the user chooses the best descriptors to express his query or his documents given his own knowledge of the domain. Moreover, the user chooses or does not choose a descriptor depending on the relationships with the other descriptors. For instance, a descriptor could be more likely to be used than another descriptor if the former tends to co-occur with other ones already employed in the same query or document.

When using a descriptor to express a query or a document, the user has his own semantics of the descriptor, which is different from the semantics of the same descriptor used by another user or by the same user in another place, time, need – in other words, the use of a descriptor depends on context. Therefore, context influences the selection of the descriptors, their semantics and inter-relationships. Some examples follow: After a series of queries about *computers*, the next query is likely not to include that key word because it can be assumed the user is refining its information need. If the user is searching for services close to *Padua*, the query might include *Venice* as well, since Padua is very close to Venice. If a document is general, highly specific terms are unlikely to be included by the user together with general terms since specific terms are more frequently used in technical documents. The evolution of queries given above depends on the evolution of the unerlying context, and reflects on the evolution of the vector base. For example, let $n = 2$ and t_1, t_2 be two descriptors, say *Padua* and *Venice*, respectively. A query q includes t_1 with weight 1; if versors are used as base vectors, $\mathbf{q} = [1\ 0]^\top$. Let us suppose the change of context has led to a new base, say the independent set of vectors $\{[1\ 2]^\top, [3\ 1]^\top\}$ is used as base, then, $\mathbf{q} = [1\ 3]^\top$.

Thus, context influences not only the choice of a descriptor, but also its semantics and the way it relates to other descriptors. This influence cannot be expressed by coefficients since these refer to one descriptor at a time and are computed by the system using statistical data about it. Context is actually expressed by the specific instance of the base vector which corresponds to the descriptor chosen to express an information need or to author a document – this notion is illustrated in the next Section by using matrices.

5 Modeling Context

If the VSM is employed to model the process illustrated in Section 4, descriptors are represented by base vectors and a base is used to generate every informative object vector. In a traditional IR setting, the model plays the role of saying *which* vectors are used to generate the vectors which represent the informative objects. Indeed, if the coefficient with which a base vector participates in generating the object vector is null, the vector does not participate in the generation. A non-null coefficient gives a measure of importance of the descriptor. Current non-context-aware IR models assume there is one vector for each descriptor.

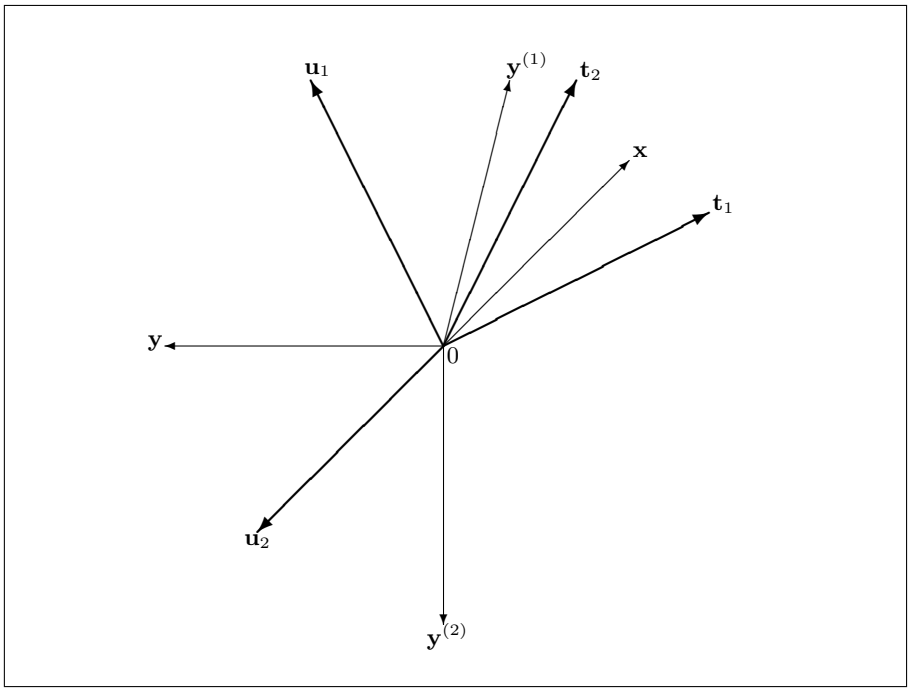

Figure1. An example of modeling context. Thick lines depict vector bases and thin lines depict generated object vectors.

If context is taken into account, its role is also the one of determining *how* the semantics of the descriptors used in that context is implemented. If, for instance, the user selects descriptor t, e.g. *computer* its base vector might be $\mathbf{t} = [1\ 0\ \frac{1}{2}\ 3]^\top$, $\mathbf{t} = [0\ 4\ \frac{3}{4}\ 1]^\top$, $\mathbf{t} = [3\ 1\ 0\ \frac{2}{3}]^\top$ or whatever – each of these vectors refer to the same descriptor and their numerical configuration depends on context. Thus, there are infinite vectors for each descriptor.

In general, the vector $\mathbf{d}$ of the document d written in its own context is generated by the base $\mathbf{T}$ which corresponds to the document context T, which is in its turn not necessarily equal to the base $\mathbf{U}$ that generates, say a query q, or another document. If relevance is estimated by the usual inner product, documents are ranked by

$$\begin{aligned}\mathbf{d} \cdot \mathbf{q} &= \mathbf{a}^\top \cdot \mathbf{T}^\top \cdot \mathbf{U} \cdot \mathbf{b} \\ &= \mathbf{a}^\top \cdot (\mathbf{T}^\top \cdot \mathbf{T}) \cdot \mathbf{c}\end{aligned}$$

where $\mathbf{d} = \mathbf{T} \cdot \mathbf{a}$ and $\mathbf{q} = \mathbf{U} \cdot \mathbf{b}$, and $\mathbf{c} = \mathbf{C} \cdot \mathbf{b}$ is the result of the rotation of $\mathbf{b}$ to the context of d. In other terms, $\mathbf{c}$ is the coefficient that a user would have used in place of $\mathbf{b}$ if the context were been represented by $\mathbf{T}$ instead by $\mathbf{U}$. Let us give the following numerical example which is also depicted in Figure 1. Document x is represented by

vector $\mathbf{x}$ and is generated by base $\mathbf{T}$ as follows:

$$\mathbf{T} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad \mathbf{a} = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad \mathbf{x} = \mathbf{T} \cdot \mathbf{a} = \begin{bmatrix} \frac{3}{2} \\ \frac{3}{2} \end{bmatrix}$$

It is worth noting that the axis, i.e. the base vectors of document x are given by $\mathbf{t}_1, \mathbf{t}_2$ which are not orthogonal, but independent of one to each other. The coefficients a_1, a_2 linearly combine $\mathbf{t}_1, \mathbf{t}_2$ which thus generate $\mathbf{x}$. Query y is similarly represented by vector $\mathbf{y}$, but is generated by base $\mathbf{U}$ and coefficients b_1, b_2 instead:

$$\mathbf{U} = \begin{bmatrix} -1 & -1 \\ 2 & -1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix} \quad \mathbf{y} = \mathbf{U} \cdot \mathbf{b} = \begin{bmatrix} -\frac{3}{2} \\ 0 \end{bmatrix}$$

It is interesting noting queries $y^{(1)}$ and $y^{(2)}$. The former is generated by $\mathbf{U}$ as follows:

$$\mathbf{b}^{(1)} = \begin{bmatrix} \frac{1}{2} \\ -1 \end{bmatrix} \quad \mathbf{y}^{(1)} = \mathbf{U} \cdot \mathbf{b}^{(1)} = \begin{bmatrix} \frac{1}{2} \\ 2 \end{bmatrix}$$

whereas the latter is generated by $\mathbf{T}$ but using the same coefficients used to generate $y^{(1)}$:

$$\mathbf{b}^{(2)} = \mathbf{b}^{(1)} \quad \mathbf{y}^{(2)} = \mathbf{T} \cdot \mathbf{b}^{(2)} = \begin{bmatrix} 2 \\ \frac{5}{2} \end{bmatrix}$$

Both query vectors have been generated by the same coefficients, but $\mathbf{y}^{(2)}$ is closer to $\mathbf{x}$ since it has been generated by the same base, where $\mathbf{y}^{(1)}$ has been generated by $\mathbf{U}$. The vector $\mathbf{y}$ is not more close to $\mathbf{x}$ than $\mathbf{y}^{(1)}$ and $\mathbf{y}^{(2)}$ because its context is far away from the context of x. This example explains how the knowledge of the context can be crucial to get the "right" document ranking.

In general, document representation and ranking is dependent on context. In fact, if $\mathbf{T}_1$ and $\mathbf{T}_2$ are the bases which, respectively, generate documents $\mathbf{d}_1$ and $\mathbf{d}_2$ with the same coefficients $\mathbf{a}$, and if a query is generated by an arbitrary context matrix $\mathbf{U}$ with coefficients $\mathbf{b}$, $\mathbf{a}^\top \cdot (\mathbf{T}_1^\top \cdot \mathbf{U}) \cdot \mathbf{b} \neq \mathbf{a}^\top \cdot (\mathbf{T}_2^\top \cdot \mathbf{U}) \cdot \mathbf{b}$.

5.1 Context Change

As descriptors are represented as base vectors, different contexts reflect on different bases. If, for instance, the user is expressing a query using the context represented by $\mathbf{t}_1 = [1\ 2]^\top, \mathbf{t}_2 = [3\ 1]^\top$, a difference in context should reflect as a different base, say $\mathbf{t}_1 = [1\ 0]^\top, \mathbf{t}_2 = [0\ 1]^\top$. This means that a base changes as context changes, i.e. there are more than one base for a collection of documents, nor does exist one base for all the queries, but there are as many bases as contexts are.

Context changes reflect on the descriptors used to describe document or query contents. Given a base $\mathbf{U}_0$, a change of the context U_0 leads to a new context U_1 represented by matrix $\mathbf{U}_1$. Since bases are expressed by matrices, a context change transforms the matrix $\mathbf{U}_0$, which expresses context U_0, to the matrix $\mathbf{U}_1$, which expresses context U_1.

Given two bases $\mathbf{U}_1$ and $\mathbf{U}_0$, there exists a unique matrix $\mathbf{C}$ such that $\mathbf{U}_1 = \mathbf{C} \cdot \mathbf{U}_0$ [1]. Note that $\mathbf{U}_i$ is a base matrix, therefore its column vectors are independent,

and then $\mathbf{U}_i^{-1}$ exists. The matrix $\mathbf{C}$ is the context change matrix and represents a linear transformation C mapping context U_0 to context U_1. As $\mathbf{U}_1$ is uniquely given by $\mathbf{U}_0$ and $\mathbf{C}$, and $\mathbf{C}$ is unique for each $\mathbf{U}_0$ and $\mathbf{U}_1$, there is a one-to-one mapping between context change matrix, linear transformation, and context change. In other words, one needs to only find the right matrix out to compute context change.

5.2 Remarks

In using vector bases as conceptual and logical tool for context modeling, some reflections are needed around the elements of base vectors. What follows is only a partial list of remarks and a close examination is of course matter of future research.

A problem is related to the definition of the elements of the base vectors, i.e. how to decide the scalars which make up the numerical configuration of the base vectors. An option would be that they can be seen as the coefficients which could have been used to combine the versors and generate the base vectors. In that case, a semantics of the versors is needed. In the Generalized VSM, an attempt was made to estimate the concepts, named "minterms", which underlies a base. The assumption was that the minterms are the 2^n versors of a vector space and the coefficients are computed by considering document similarities, yet context were not considered. For details, the reader is suggested to refer to [8].

Given a query context, if there is only one document context for the collection, then document context is not influential in ranking retrieved documents. Indeed, it is easy to see that, given a query base $\mathbf{U}$, if $\mathbf{T}_i = \mathbf{T}$ for all $i = 1, \ldots, m$, where m is the number of documents, there exists a unique $\mathbf{C}$ such that $\mathbf{U} = \mathbf{C} \cdot \mathbf{T}_i$ for each i. Thus, there exists a single $\mathbf{c} = \mathbf{C} \cdot \mathbf{b}$ and the retrieval function is based on $\mathbf{d}_i \cdot \mathbf{q} = \mathbf{a}_i^\top \cdot (\mathbf{T}^\top \cdot \mathbf{T}) \cdot \mathbf{c}$ which depends on $\mathbf{a}_i$ only.

It should be pointed out that orthogonality *versus* non-orthogonality is a different notion of the notion of context *versus* non-context. Indeed, let $\mathbf{d}_1$ and $\mathbf{d}_2$ be two document vectors generated by the orthonormal base represented by $\mathbf{I}$. If these documents are matched against a query q, and $\mathbf{d}_1^\top \cdot \mathbf{q}$ and $\mathbf{d}_2^\top \cdot \mathbf{q}$ are computed, document ranking is different from the one computed if the documents have been generated by another base $\mathbf{T}$.

It should be also pointed out that orthogonality does not mean absence of context, there are on the contrary infinite bases and contexts related to an diagonal matrix. Indeed, if $\mathbf{I}$ is the identity matrix used to represent orthogonality between any pair of base vectors, there exist more than one base $\mathbf{T}$, and then more than one context, such that $\mathbf{T}^\top \cdot \mathbf{T} = \mathbf{I}$.

6 Relevance Feedback

Relevance Feedback (RF) is a form of query context change. Let us describe RF in the VSM. Provided a query q, the generation of vector $\mathbf{q}$ by base T is given by $\mathbf{T} \cdot \mathbf{b}$. RF computes a new query vector after observing $0 \leq r \leq N$ relevant documents $\{d_1, \ldots, d_r\}$ and of $N - r$ non-relevant documents $\{d_{r+1}, \ldots, d_n\}$. Let q^+ be the

new query and $\mathbf{q}^+$ be the vector which represents it. By using the classical relevance feedback formulation provided with the VSM, the vector is expressed as:

$$\mathbf{q}^+ = \mathbf{q} + \alpha \sum_{i=1}^{r} \mathbf{d}_i - \beta \sum_{j=r+1}^{n} \mathbf{d}_j$$

where α and β are two weights whose values are decided after some training experiments. After substituting document and query vectors with the corresponding generation formulas:

$$\begin{aligned}
\mathbf{q}^+ &= \sum_{k=1}^{n} b_k \mathbf{t}_k + \alpha \sum_{i=1}^{r} \sum_{k=1}^{n} a_{ik} \mathbf{t}_k - \beta \sum_{j=r+1}^{n} \sum_{k=1}^{n} a_{jk} \mathbf{t}_k \\
&= \sum_{k=1}^{n} b_k \mathbf{t}_k + \sum_{k=1}^{n} \alpha \left(\sum_{i=1}^{r} a_{ik} \right) \mathbf{t}_k - \sum_{k=1}^{n} \beta \left(\sum_{j=r+1}^{n} a_{jk} \right) \mathbf{t}_k \\
&= \sum_{k=1}^{n} \left(b_k + \alpha \sum_{i=1}^{r} a_{ik} - \beta \sum_{j=r+1}^{n} a_{jk} \right) \mathbf{t}_k \\
&= \sum_{k=1}^{n} b_k^+ \mathbf{t}_k
\end{aligned}$$

where

$$\begin{aligned}
b_k^+ &= b_k + \alpha \sum_{i=1}^{r} a_{ik} - \beta \sum_{j=r+1}^{n} a_{jk} \\
&= b_k + r_k - s_k
\end{aligned}$$

and $r_k = \alpha \sum_{i=1}^{r} a_{ik}$ and $s_k = \beta \sum_{j=r+1}^{n} a_{jk}$. Using a matrix notation, one can write:

$$\mathbf{b}^+ = \mathbf{b} + \mathbf{r} - \mathbf{s}$$

where $\mathbf{b} = [b_1, \ldots, b_n]^\top$, $\mathbf{r} = [r_1, \ldots, r_n]^\top$, $\mathbf{s} = [s_1, \ldots, s_n]^\top$. There exists $\mathbf{C}$ such that $\mathbf{b}^+ = \mathbf{C} \cdot \mathbf{b}$ such that the column vectors of $\mathbf{C}$ form an indepedent set. Such a matrix can be built as follows: The elements of $\mathbf{b}$ are permuted so that the n-th element is not zero. The elements of the other vectors are permuted accordingly. For each k, if $b_k \neq 0$, then $c_{kk} = 1 + (r_k - n_k)/b_k$ and the elements of the same row of c_{kk} are set to zero. If $b_k = 0$, then let h be the index of the first non-zero element b_h such that $h > k$; such an element exists because the $b_n \neq 0$ after permutation. Then, set c_{kh} to $(r_k - n_k)/b_h$, c_{kk} to an arbitrary constant $\epsilon \neq 0$ and the other elements of the same row of c_{kk} are set to zero. The result is a triangular matrix $\mathbf{C}$ with non-null diagonal elements. Thus, $b_k^+ = \sum_{i=1}^{n} c_{ki} b_i = \sum_{i=k}^{n} c_{ki} b_i = (b_k + r_k - n_k)$ if $b_k \neq 0$, and $b_k^+ = \epsilon b_k + (r_k - n_k)/b_h = (r_k - n_k)/b_h$, otherwise.

The ϵs in the diagonal of $\mathbf{C}$ correspond to the null elements of $\mathbf{b}$ and, therefore, do not affect the computation of $\mathbf{b}^+$. However, the ϵ's are necessary to make the column

vectors independent. Indeed, since $\epsilon \neq 0$ and $b_k = 0$ or $c_{kk} \neq 0$ and $b_k \neq 0$ for every k, then $\det(\mathbf{C}) \neq 0$, because $\mathbf{C}$ is triangular, and the column vectors are independent.

Note that if $\mathbf{q}$ is generated by $\mathbf{T}$, then $\mathbf{q}^+$ is generated by $\mathbf{T} \cdot \mathbf{C}$ because $\mathbf{q}^+ = \mathbf{T} \cdot \mathbf{b}^+ = (\mathbf{T} \cdot \mathbf{C}) \cdot \mathbf{b}$. Therefore, the context is provided by the partition of the collection in the set of relevant documents and its complement. Context change is represented by $\mathbf{C}$ which shows that RF equals to re-computing a query in the context of the partition relevant/non-relevant documents. In particular, this context change matrix is obtained by increasing the weight of "positive" descriptors and by decreasing the one of "negative" descriptors. Every technique, such as query translation-based Cross-Language IR or word sense disambiguation, which is implemented as a transformation of a query to another might be seen as a context change.

7 Conclusions and Future Research

In this paper it is argued that the VSM can be extended to incorporate the notion of context. A base of a vector space can operate at this aim by modeling context directly. The correspondence between Relevance Feedback, which is an instance of context change, and linear transformation between bases witnesses the feasibility of the proposed approach.

The developments provide a proposal not only for the computation of a measure of relevance in case of context-aware search, but also for the incorporation of context into the model. The major strength of this proposal derives from the fact that it naturally employ a sound mathematical framework.

Current research is devoted to the computation of the base vectors, the semantics of the base vectors, the mapping between contexts and bases, the implementation of the framework in various domains, e.g. browsing or query expansion, other that to the efficiency issues relative to matrix computation. A great deal of attention, and consequently of research, shall be paid to evaluation, which is inevitably a long term objective.

References

1. Apostol, T.: 1969, *Calculus*. New York: J. Wiley & Sons, 2nd edition.
2. Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman: 1990, 'Indexing by Latent Semantic Analysis'. *Journal of the American Society for Information Science* **41**(6), 391–407.
3. Melucci, M. and J. Rehder: 2003, 'Using Semantic Annotations for Automatic Hypertext Link Generation in Scientific Texts'. In: N. Ashish and C. Goble (eds.): *Proceedings of the Workshop on Semantic Web Technologies for Searching and Retrieving Scientific Data*. Sanibel Island, Florida, USA. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-83/.
4. Salton, G.: 1989, *Automatic Text Processing*. Addison-Wesley.
5. Salton, G., A. Wong, and C. S. Yang: 1975, 'A Vector Space Model for Automatic Indexing'. *Communications of the ACM* **18**(11), 613–620.
6. van Rijsbergen, C.: 2004, *The Geometry of Information Retrieval*. UK: Cambridge University Press.

7. Wong, S. and V. Raghavan: 1984, 'Vector Space Model of Information Retrieval – A Reevaluation'. In: *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*. Cambridge, England, pp. 167–185.
8. Wong, S., W. Ziarko, V. Raghavan, and P. Wong: 1987, 'On Modeling of Information Retrieval Concepts in Vector Spaces'. *ACM Transactions on Database Systems* **12**(2), 299–321.