A General Matrix Framework for Modelling Information Retrieval

Thomas Rölleke Theodora Tsikrika Gabriella Kazai

Abstract

Content-oriented retrieval models are based on a document-term matrix, whereas link-oriented retrieval models are based on an adjacent (parent-child) matrix. Term frequency and inverse document frequency are key concepts in content-oriented retrieval, whereas pagerank, authorities and hubs are key concepts in link-oriented retrieval. We present in this paper a general matrix framework that covers content-oriented and link-oriented retrieval. Thereby, we include, in addition to content and links, the structure of documents, the retrieval quality and the semantics of indexing terms. The result of this paper is a general matrix framework for modelling information retrieval. The benefit of this framework lies in its high level of reusability and abstraction. The framework improves information retrieval in the sense that system construction becomes significantly more efficient, and thus, better and more personalised systems can be build at less costs.

1 Introduction

With the web and its search engines, ranking of retrieved objects becomes a focus in many application areas. More and more people face the task to build complex information systems that provide ranking functionality. The matrix framework presented in this paper contributes to the understanding of retrieval concepts, and it supports the construction of search systems since the matrix operations provide a high level of reusability and abstraction.

The matrix framework improves retrieval in the sense that system construction becomes more efficient, flexible and robust. For a search system engineer, the flexibility of tools is crucial, since the flexibility of retrieval and indexing functions yields the possibility to tune the effectiveness and efficiency of a system for the particular needs of an end user.

The literature background of this work is based on general IR literature such as [12], [6], [3], [4], and more specific literature such as [14], [13], [2], [8] and [7]. [14] and [13] and other publications of the authors on the generalised vector-space model and the probabilistic framework for information retrieval are major foundations and motivations for the matrix framework presented in our paper. Further, [2] on the duality of document indexing and relevance feedback, and [1] on probability distributions for exploiting term frequencies and capturing normalisation motivated our work to present a general matrix framework in which those methods can be applied to more than "just term frequencies". The extension of our matrix framework towards a probabilistic framework with probability distributions is one of the next research goals. The results and notations of [8] and [7] were input regarding link-oriented retrieval. All of the above literature addresses the formalisation of either content or links (structure), whereas in this paper we propose a general matrix framework for both content and structure. In addition, we include relevance feedback and evaluation.

Our paper is structured as follows: First, we introduce the matrix spaces in section 2. We consider a collection, a document and a query space, where we associate several matrices with each space. After having introduces the overall matrix framework, we address some matrices in more detail. Section 3 investigates tf-idf based on the content matrices. Section 4 addresses the collection and the document structure. In sections 5 and 6, we apply the matrix framework for modelling probabilistic relevance feedback and precision/recall based on the query result matrix. With the general framework presented here, tf-idf can be applied on, for example, the link matrix, concepts such as pagerank, authorities and hubs can be transferred to the term-term matrix of a collection. The mathematical and formal definition of the crucial concepts of IR in our matrix framework is a contribution on the way to a logical layer of IR.

2 The Matrix Spaces

Our aim is to provide a general model for IR based on matrices, since such a general model leads to a system implementation with high reusability, flexibility and robustness.

Table 1 shows three spaces: The collection space, the document space, and the query result space. A space contains several elements and has two dimensions. With each element of each space, we associate matrices. A content (document-term), structure (document-document) and semantics (term-term) matrix is associated with each collection, a content (location-term) matrix is associated with each document, and a result (document-class) matrix is associated with each query. The class dimension is also referred to as *assessment* dimension.

The document-term matrix DT_c is the content matrix of a collection. The location-term matrix LT_d is the content matrix of a document. The document-class matrix DA_q is the result matrix of a query. For each of these three basic matrices, we can compute the product of the matrix and its transposition. The results reflect the document similarity and term similarity in the collection space, the location similarity and term similarity in the document space, and the class degree and precision/recall in the query result space.

The tables 2 and 3 show the modelling of a link (parent-child) structure for the dimensions of the collection space and the document space. Here, we find the link-structure known from web retrieval (structure of a collection, matrix PC_{Dc}) in the collection space and the document structure is to be found in the document space (matrix PC_{Ld}). For the document structure, we choose the terminology *location*, where location shall cover concepts such as *section*, *paragraph*, and *position*.

The tables introduce a carefully chosen notation. We use a capital letter for indicating the dimensions of each matrix, and we use a lower case letter as subscript for indicating the space. The matrices that present the link structure among a dimension are named *PC* for parent-child. The parent-child matrices carry a subscript that indicates the dimension and the space. For the scope of this paper, we restrict to the links of the elements in the dimensions of the collection space and the document space.

The carefully chosen notation for indicating matrix dimensions and spaces is a main result of this paper. (The notation has one weakness that might disturb the reader: T is used for the term dimension and for the matrix transposition.) The notation allows for a general IR model with high abstraction. It allows us to show the duality between query term expansion based on term co-occurrence and pagerank, as we will show next.

Consider the matrices and meanings of eigenvectors depicted in the tables 4 and 5.

The matrix TT_c is the term similarity matrix, i. e. it reflects the co-occurrence of terms. If we multiply a query vector of terms with the term similarity matrix, then similar terms are considered in the modified query vector.

$$\vec{q}_{modified} = TT_c \cdot \vec{q}$$

Query elements (terms) that had a value of 0 in the input query might have in the modified query a value greater than 0 because similar terms do occur in the query. For the eigenvectors of TT_c, we obtain:

$$\lambda \cdot \vec{q} = TT_c \cdot \vec{q}$$

The factor λ is a scalar that scales the vector. The eigenvector of TT_c is a query (a document, respectively) that reflects the information in TT_c in the sense that if a term occurs, then the similar terms of the term also do occur. The eigenvector of TT_c reflects term co-occurrence.

Next, we look at an equation that models pagerank. Let $link(x, y)$ be 1 if document x links to document y, and 0 otherwise. Let $pr(x)$ be the pagerank of a document (page). (We work here with a simplified form of the pagerank formulae where assume that the pagerank values contain a normalisation with respect to outgoing links and we omit a starting value.) We obtain the pagerank $pr(y)$ of a document y as follows:

$$pr(y) = \sum_x link(x, y) \cdot pr(x)$$

The summation can be expressed using the parent-child matrix PC_{Dc}.

$$\vec{y} = {PC_{Dc}}^T \cdot \vec{x}$$

The elements of ${PC_{Dc}}^T$ reflect whether document x links to document y, and the vector $\vec{y}$ contains the pagerank values of all documents.

For example, for the structure in figure 1, we obtain:

$${PC_{Dc}}^T = \left(\begin{array}{c|ccc} & d_1 & d_2 & d_3 \\ \hline d_1 & 0 & 0 & 0 \\ d_2 & 1 & 0 & 0 \\ d_3 & 1 & 1 & 0 \end{array}\right)$$

Collection space	Document space	Query result space
DT_c: Documents × Terms	LT_d: Locations × Terms	DA_q: Documents × Classes
$DD_c = DT_c \cdot {DT_c}^T$ Document sim. (term degree)	$LL_d = LT_d \cdot {LT_d}^T$ Location sim. (term degree)	$DD_q = DA_q \cdot {DA_q}^T$ Class degree
$TT_c = {DT_c}^T \cdot DT_c$ Term sim. (document degree)	$TT_d = {LT_d}^T \cdot LT_d$ Term sim. (location degree)	$AA_q = {DA_q}^T \cdot DA_q$ Precision/Recall

Table 1: The matrix spaces: collection, document and query result

Links in the dimensions of the collection space	
Document links (collection structure)	Term links (collection semantics)
PC_{Dc}: Parents × Children	PC_{Tc}: Parents × Children
$PP_{Dc} = PC_{Dc} \cdot {PC_{Dc}}^T$: Out-degree of documents	$PP_{Tc} = PC_{Tc} \cdot {PC_{Tc}}^T$: Generality of terms
$CC_{Dc} = {PC_{Dc}}^T \cdot PC_{Dc}$: In-degree of documents	$CC_{Tc} = {PC_{Tc}}^T \cdot PC_{Tc}$: Specificity of terms

Table 2: Structure and semantics in a collection

Links in the dimensions of the document space	
Location links (document structure)	Term links (document semantics)
PC_{Ld}: Parents × Children	PC_{Td}: Parents × Children
$PP_{Ld} = PC_{Ld} \cdot {PC_{Ld}}^T$: Out-degree of locations	$PP_{Td} = PC_{Td} \cdot {PC_{Td}}^T$: Generality of terms
$CC_{Ld} = {PC_{Ld}}^T \cdot PC_{Ld}$: In-degree of locations	$CC_{Td} = {PC_{Td}}^T \cdot PC_{Td}$: Specificity of terms

Table 3: Structure and semantics in a document

Matrix	Matrix elements	Eigenvector meaning
DD_c	Number of common terms	a term that reflects document co-containment
TT_c	Number of common documents	a document that reflects term co-occurrence
LL_d	Number of common terms	a term that reflects location co-containment
TT_d	Number of common locations	a location that reflects term co-occurrence
DD_q	Number of common classes	a class that reflects document co-membership
AA_q	Number of common documents	a document that reflects class co-containment

Table 4: Space matrices and their eigenvectors

Matrix	Matrix elements	Eigenvector meaning
PC_{Dc}	$PC_{Dc} = [a_{ij}]$, where document i links (points) to document j	pagerank based on out-going links; hub-oriented
${PC_{Dc}}^T$	${PC_{Dc}}^T = [a_{ij}]$, where document i is referenced by document j	pagerank based on in-coming links; authority-oriented
PP_{Dc}	Number of common child documents; parent similarity; out-degree	a document that reflects common parents; $\lambda \cdot \vec{x} = PC_{Dc} \cdot {PC_{Dc}}^T \cdot \vec{x}$; a hub
CC_{Dc}	Number of common parent documents; child similarity; in-degree	a document that reflects common children; $\lambda \cdot \vec{x} = {PC_{Dc}}^T \cdot PC_{Dc} \cdot \vec{x}$; an authority

Table 5: Document dimension matrices and their eigenvectors

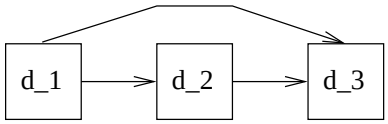

Figure 1: Three linked documents

For the eigenvector $\vec{x}$ of ${PC_{Dc}}^T$, we have:

$$\lambda \cdot \vec{x} = {PC_{Dc}}^T \cdot \vec{x}$$

The eigenvector of ${PC_{Dc}}^T$ contains the pagerank values of each document. Here, the fix point meaning of an eigenvector in system analysis becomes evident: an eigenvector is an input vector to a system represented by a matrix such that the system does not change the vector. This fix point view helps to understand the eigenvector meaning.

The tables 4 and 5 show the meaning of the eigenvectors of several matrices. With our matrix framework, we show here the correspondence of the eigenvector operation, and the notion of pagerank can be transferred, for example, to the PC_{Tc} matrix that reflects the links (the semantics) of a collection.

We take a closer look at the eigenvectors of PP_{Dc} and CC_{Dc}. PP_{Dc} is the parent similarity and corresponds in the collection space to DD_c, the document similarity. CC_{Dc} is the child similarity (in-degree) and corresponds in the collection space to TT_c, the term similarity. With this correspondence, we consider containment (document contains term) to be dual to linking (parent links to child), and occurrence (term occurs in documents) to be dual to referencing (child is referenced by parent).

As explained above, TT_c can be used for query (document, respectively) expansion. With $TT_c \cdot T_q$, we add terms to vector T_q that are similar to the terms in T_q. Analogously, with $CC_{Dc} \cdot C_d$, we add children to vector C_d that are similar to the children in C_d. On the other hand, with $PP_{Dc} \cdot P_d$, we add parents to vector P_d that are similar to the parents in P_d. An eigenvector of TT_c reflects the term similarity. An eigenvector of CC_{Dc} reflects the child similarity (in-degree, authority), and an eigenvector of PP_{Dc} reflects the parent similarity (out-degree, hub).

In the next sections, we present the usage of our framework for classical retrieval parameters such as term frequency and document frequency. We start with the content matrices of the collection and the document space.

3 Content

The content of the collection is represented by the document-term matrix of the collection space and the content of a document is represented by the location-term matrix of the document space.

This section investigates how the inverse document frequency of terms can be described by using the collection content matrix (3.1) and how the location (term) frequency of terms can be described by using the document content matrix (3.2).

3.1 Collection space

In a collection space c, each of the two dimensions (documents and terms) is modelled as a vector.

The vector of documents in the collection is defined as $D_c = [d_i]_{N\times 1}$, where $d_i \geq 0$ is the document weight. This weight can be used to define the importance of a document in the collection. It can be estimated by taking into account the source of the document, the size of the document, the number of incoming and outgoing links (in the case of hyperlinked documents) or other available evidence. The L_1- norm of the document vector is defined as $\|D_c\|_1 \equiv \sum_{i=1}^{N} |d_i|$. In the simple case where the document weight denotes just the presence (occurrence) of the document in the collection:

$$d_i = \begin{cases} 1 & \text{if } d_i \in c \\ 0 & \text{if } d_i \notin c \end{cases}$$

The L_1- norm represents the number of documents in the collection: $N_{D_c} = \|D_c\|_1$

Similarly the vector of terms in the collection is defined as: $T_c = [t_i]_{M\times 1}$, $t_i \geq 0$ and the L_1- norm of the term vector is defined as $\|T_c\|_1 \equiv \sum_{i=1}^{M} |t_i|$. In the simple case where

$$t_i = \begin{cases} 1 & \text{if } t_i \in c \\ 0 & \text{if } t_i \notin c \end{cases}$$

Then the L_1- norm represents the number of terms in the collection: then $N_{T_c} = \|T_c\|_1$.

Let $DT_c = [dt_{ij}]_{N\times M}$, be the matrix of document-term information in the collection, where the rows correspond to documents and columns to terms. We define each matrix element:

$$dt_{ij} = \begin{cases} 1 & \text{if } t_j \in d_i \\ 0 & \text{if } t_j \notin d_i \end{cases}$$

For example, let us consider the collection containing documents d_1, d_2 and d_3 and terms t_1, t_2, t_3 and t_4. The document and term vectors are:

$$D_c = \left(\begin{array}{c|c} & c \\ \hline d_1 & 1 \\ d_2 & 1 \\ d_3 & 1 \end{array}\right)$$

$$T_c = \left(\begin{array}{c|c} & c \\ \hline t_1 & 1 \\ t_2 & 1 \\ t_3 & 1 \\ t_4 & 1 \end{array}\right)$$

and the number of documents and terms in the collection are $N_{D_c} = 3$ and $N_{T_c} = 4$ respectively.

Let the document-term matrix of our collection be:

$$DT_c = \left(\begin{array}{c|cccc} & t_1 & t_2 & t_3 & t_4 \\ \hline d_1 & 1 & 1 & 1 & 1 \\ d_2 & 0 & 1 & 1 & 0 \\ d_3 & 1 & 1 & 0 & 0 \end{array}\right)$$

Our aim is to define the inverse document frequency of a term in a collection based on the content matrix of the collection.

The common definition of the inverse document frequency $idf(t, c)$ based on the document frequency $df(t, c)$ of a term in a collection is:

$$df(t, c) := \frac{n_d(t, c)}{N_d(c)} \tag{1}$$

$$idf(t, c) := -\log df(t, c) \tag{2}$$

where $N_d(c)$ is the number of documents in the collection and $n_d(t, c)$ is the number of documents in which term t occurs

Following these definitions, we describe idf in our matrix framework, by defining a vector of the terms in the collection where each element is the number of documents containing the term:

$$nd\text{-}T_c = D_c^T \cdot DT_c = [nd_{t_i}]_{1\times M}$$

For our example, we obtain

$$nd\text{-}T_c = \left(\begin{array}{c|cccc} & t_1 & t_2 & t_3 & t_4 \\ \hline nd & 2 & 3 & 2 & 1 \end{array}\right)$$

By normalising each element of $nd\text{-}T_c$ with the number of documents in the collection, we obtain a vector of the terms in the collection where each element is the document frequency (*df*) of the term:

$$df\text{-}T_c = \frac{nd\text{-}T_c}{N_{D_c}}$$

For our example, this yields:

$$df\text{-}T_c = \left(\begin{array}{c|cccc} & t_1 & t_2 & t_3 & t_4 \\ \hline df & \frac{2}{3} & \frac{3}{3} & \frac{2}{3} & \frac{1}{3} \end{array}\right)$$

Next, we apply the negative logarithm on each matrix element of $df\text{-}T_c$ to obtain the inverse document frequencies. Let *apply(f,M)* be a function which applies the function *f* to each matrix element. We obtain the $idf\text{-}T_c$ vector which is the vector of the terms in the collection where each element is the inverse document frequency of the term:

$$idf\text{-}T_c = apply(-\log, df\text{-}T_c)$$

For our example, we obtain:

$$idf\text{-}T_c = \left(\begin{array}{c|ccc} & t_1 & t_2 & ... \\ \hline idf & -\log\frac{2}{3} & -\log\frac{3}{3} & ... \end{array}\right)$$

Next, we investigate — analog to the document frequency of a term — the term frequency of a document. (Note that we investigate the term frequency of a *document*, not the term frequency of a *term*. The latter one is dealt with in section 3.2.)

The definition of the inverse term frequency $itf(d, c)$ of a document is based on the term frequency $tf(d, c)$ of a document in a collection.

$$tf(d, c) := \frac{n_t(d, c)}{N_t(c)} \tag{3}$$

$$itf(d, c) := -\log tf(d, c) \tag{4}$$

Note the correspondence between definition 1 (document frequency value) and definition 3 (term frequency value), definition 2 (inverse document frequency value) and definition 4 (inverse term frequency value).

Therefore, we can analogously define a vector of the documents in the collection where each element is the number of terms occurring in the document:

$$nt\text{-}D_c = DT_c \cdot T_c = [nt_{d_i}]_{N\times 1}$$

a vector of the documents in the collection where each element is the term frequency of the document:

$$tf\text{-}D_c = \frac{nt\text{-}D_c}{N_{T_c}}$$

and a vector of the documents in the collection where each element is the inverse term frequency of the document:

$$itf\text{-}D_c = apply(-\log, tf\text{-}D_c)$$

For our example, we obtain:

$$itf\text{-}T_c = \left(\begin{array}{c|ccc} & d_1 & d_2 & d_3 \\ \hline itf & -\log\frac{4}{4} & -\log\frac{2}{4} & -\log\frac{2}{4} \end{array}\right)$$

The inverse document frequency reflects the so-called *discriminative power* (occurrence) of a term, the inverse term frequency reflects the *specific power* (length) of a document.

Note the perfect mathematical analogy between document and term frequency. However, there is a terminological misfit with the common term frequency definition (common is the usage of term frequency for a term in a document) and the term frequency of a document. The term frequency defined in this section is the term frequency of a document in a collection, whereas the classical term frequency corresponds to the *location* frequency of a term in a document, as we point out in section 3.2.

3.2 Document space

Similarly to the description of the collection space, each of the two dimensions (location-term) of the document space d is modelled as a vector.

The vector of locations in the document is defined as $L_d = [l_i]_{R\times 1}, l_i \geq 0$ and the vector of terms in the document as $T_d = [t_i]_{S\times 1}, t_i \geq 0$

Let $LT_d = [lt_{ij}]_{R\times S}$ be the matrix of location-term information in the document, where rows correspond to locations and columns to terms. Each matrix element is defined as:

$$lt_{ij} = \begin{cases} 1 & \text{if } t_j \in l_i \\ 0 & \text{if } t_j \notin l_i \end{cases}$$

Let a document with content such as "sailing boats greece sailing" be given. The location and term vectors of this document are then defined as follows:

$$L_d = \left(\begin{array}{c|c} & d \\ \hline l_1 & 1 \\ l_2 & 1 \\ l_3 & 1 \\ l_4 & 1 \end{array}\right)$$

$$T_d = \left(\begin{array}{c|c} & d \\ \hline t_1 & 1 \\ t_2 & 1 \\ t_3 & 1 \end{array}\right)$$

and the location-term matrix representing the document content is:

$$LT_d = \left(\begin{array}{c|ccc} & t_1 & t_2 & t_3 \\ \hline l_1 & 1 & 0 & 0 \\ l_2 & 0 & 0 & 1 \\ l_3 & 0 & 1 & 0 \\ l_4 & 1 & 0 & 0 \end{array}\right)$$

Our aim is to define the location frequency of a term in a document using the matrices of the document space. Note that this corresponds to the classical IR notion of term frequency and this becomes clear as we first present the classical term frequency definition and then introduce the location frequency definition.

The term frequency of a term in a document is commonly defined as:

$$tf(t,d) := \frac{occ(t,d)}{occ(t_{max},d)} \tag{5}$$

$$0 \leq tf(t,d) \leq 1, occ(t,d) \geq 0,$$
$$\forall t : occ(t,d) \leq occ(t_{max},d)$$

The tf-value is sometimes lifted to the interval $0.5 \leq tf(t,d) \leq 1$ with

$$tf(t,d) := \frac{1}{2} \cdot \left(1 + \frac{occ(t,d)}{occ(t_{max},d)}\right)$$

In general, the lifting to the interval $a \leq tf(t,d) \leq 1.0$ can be described with

$$tf(t,d) := a + (1-a) \cdot \frac{occ(t,d)}{occ(t_{max},d)}$$

We mention the lifting here to be comprehensive, however, for the matrix-based definition we do not consider it further, since the lifting factor is heuristic and depends on the actual collection and retrieval function. See [11] and related publications for the usage of term frequencies.

Following these definitions, we introduce the matrix-based definition of the location frequency of a term in a document. Firstly, we define a vector of the terms in the document where each element is the number of locations (nl) containing the term:

$$nl\text{-}T_d = L_d^T \cdot LT_d = [nl_{t_i}]_{1\times S}$$

For our example, we obtain:

$$nl\text{-}T_d = \left(\begin{array}{c|ccc} & t_1 & t_2 & t_3 \\ \hline nl & 2 & 1 & 1 \end{array}\right)$$

The next step is to define a vector of the terms in the document where each element is the location frequency (lf) of the term:

$$lf\text{-}T_d = \frac{nl\text{-}T_d}{\|nl\text{-}T_d\|_\infty}$$

($\|.\|_\infty$: $L_{\infty-norm}$ $\|\vec{x}\|_\infty \equiv \max_i |x_i|$)

Whereas the document frequency of a term was defined by normalising with the number of documents in the collection, the location frequency of a term is defined by normalising with the maximal location frequency in the document.

For our example, we obtain:

$$lf\text{-}T_d = \left(\begin{array}{c|ccc} & t_1 & t_2 & t_3 \\ \hline lf & \frac{2}{4} & \frac{1}{4} & \frac{1}{4} \end{array}\right)$$

Also, as previously done in the collection space, where we defined the term frequency of a document, we can define in the document space the term frequency of a location.

This can be achieved by defining the vector of the locations in the document where each element is the number of terms occurring in the location:

$$nt\text{-}L_d = LT_d \cdot T_d = [nt_{l_i}]_{R\times 1}$$

and the vector of the locations in the document where each element is the term frequency of the location:

$$tf\text{-}L_d = \frac{nt\text{-}L_d}{\|nt_{L_d}\|_\infty}$$

Next, we describe the matrices related to structure.

4 Structure and Semantics

In the collection space and in the document space, the links among the dimensions constitute the structure and the semantics in a collection and a document, respectively.

PC_{Dc} is the matrix that reflects the collection structure (links among documents), and PC_{Ld} is the matrix that reflects the document structure (links among document parts). The matrices $PP_{Dc} = PC_{Dc} \cdot {PC_{Dc}}^T$ and $CC_{Dc} = {PC_{Dc}}^T \cdot PC_{Dc}$ reflect the document parent and child similarity, as pointed out in section 2. Thereby, PP_{Dc} is also referred to as *co-citation degree*, i. e. the degree to which two documents cite the same children. CC_{Dc} is referred to as *bibliographic coupling degree*, i. e. the degree to which two documents are cited by the same parents.

In a dual way, we can consider PP_{Ld} and CC_{Ld} in a document. These parameters are potentially useful in structured document retrieval where we face the task of estimating probabilities for document parts. The probability estimation could take the "hub" and "authority" feature of document parts into account.

The semantics in the collection space is reflected in PC_{Tc}, from which we derive PP_{Tc} and CC_{Tc}, reflect the term parent and term child similarities. The eigenvector meanings given in table 5 apply in a dual way to PP_{Tc} and CC_{Tc}. An "authority" term is a term with a high number of incoming links, i. e. an authority term is a specialisation of several general terms. For example, "business technology transfer manager" is an authority, since this compound is a specialisation of several general terms. A "hub" term is a term with many outgoing links (many specialisation). For example, a name such as "Smith" could be a hub term, since it expands to many compounds that are distinctive in the first name. With this "hub" view on terms, terms with several meanings (homonymy) and smallest parts of a word with a meaning (morphemes) are hub candidates. Hub terms tend to be general (broad) terms whereas authority terms tend to be specific (narrow) terms. The combination of this hub and authority view on terms with the term similarity matrix TT_c is an interesting challenge since it adds a term characteristics to the otherwise purely occurrence-based similarity measure.

Next, we consider the modelling of precision and recall in our matrix framework.

5 Precision and Recall

Precision and recall are defined as follows:

$$precision := \frac{retrieved \cap relevant}{retrieved}$$

$$recall := \frac{retrieved \cap relevant}{relevant}$$

The description of precision/recall in our matrix framework is based on the document assessment matrix of a query result. Let *ret* stand for *retrieved* and let *rel* stand for *relevant*. Let d_1, d_2, and d_4 be retrieved documents, and let d_1 and d_3 be relevant documents. This is represented in matrix DA_q as

follows:

$$DA_q = \left(\begin{array}{c|cc} & ret & rel \\ \hline d_1 & 1 & 1 \\ d_2 & 1 & 0 \\ d_3 & 0 & 1 \\ d_4 & 1 & 0 \end{array} \right)$$

Analog to the operation for obtaining a term-term matrix from the document-term matrix, we perform the operation for obtaining an assessment-assessment matrix from the document-assessment matrix.

$$AA_q = {DA_q}^T \cdot DA_q = \left(\begin{array}{c|cc} & ret & rel \\ \hline ret & 3 & 1 \\ rel & 1 & 2 \end{array} \right)$$

The precision/recall values can be derived directly from the matrix elements.

$$precision = \frac{AA_q(ret, rel)}{AA_q(ret, ret)}$$

$$recall = \frac{AA_q(ret, rel)}{AA_q(rel, rel)}$$

The above definition captures the set view on retrieval results. However, a retrieval system returns a list of documents rather than a set of documents, and we require to capture the list information in evaluation.

Let

$$DA_{q_1} = \left(\begin{array}{c|cc} & ret & rel \\ \hline d_1 & 0.7 & 1 \\ d_2 & 0.5 & 0 \\ d_3 & 0.0 & 1 \\ d_4 & 0.5 & 0 \end{array} \right)$$

and

$$DA_{q_2} = \left(\begin{array}{c|cc} & ret & rel \\ \hline d_1 & 0.5 & 1 \\ d_2 & 0.7 & 0 \\ d_3 & 0.0 & 1 \\ d_4 & 0.5 & 0 \end{array} \right)$$

be the retrieval results of two systems, where the ranks and RSV's of d_1 and d_2 are swapped. We obtain the assessment-assessment matrices

$$AA_{q_1} = \left(\begin{array}{c|cc} & ret & rel \\ \hline ret & 0.99 & 0.7 \\ rel & 0.7 & 2 \end{array} \right)$$

and

$$AA_{q_2} = \left(\begin{array}{c|cc} & ret & rel \\ \hline ret & 0.99 & 0.5 \\ rel & 0.5 & 2 \end{array} \right)$$

The precision and recall value derived from AA_{q_1} are higher than the values derived from of AA_{q_2}. That meets our expectation since system 1 retrieves d_1, a relevant document, with a higher value (namely, $RSV(d_1, q) = AA_{q_1}(d_1, ret) = 0.7$) than system 2 does (here, $RSV(d_1, q) = AA_{q_2}(d_1, ret) = 0.5$).

We have sketched in this section the usage of our matrix framework for a retrieval quality measure. The potential of the matrix framework lies in the definition and management of more complex measures. For example, we want to consider the efficiency of query processing and the structure of documents in system evaluation. For efficiency, we can introduce an additional assessment column in DA_q where the column reflects the time at which a document is delivered by a system. For structure, we can exploit the links of the location dimension of a document (matrix PC_{Ld}). The definition of those new evaluation measures and the extension of DA_q is important for structured document retrieval and the matrix framework is a formalism in which those new evaluation measures can be established.

As a last application of our matrix framework, we investigate the modelling of the probabilistic retrieval model based on relevance feedback.

6 Relevance Feedback

There are two widely known relevance feedback models: the probabilistic model and the vector-based model. We concentrate here on the probabilistic model, since the modelling of the vector-based model follows directly from the vector representation of documents.

The common notation for the probabilistic model is:

$P(t\|R)$	probability that term t occurs in a relevant document
$P(t\|\neg R)$	probability that term t occurs in a non-relevant document
$P(\neg t\|R)$	probability that term t does not occur in a relevant document
$P(\neg t\|\neg R)$	probability that term t does not occur in a non-relevant document
$RSV(d,q)$	retrieval status value for the document-query pair (d,q)

$$\begin{aligned} c_t &:= \log \frac{P(t|R) \cdot P(\neg t|\neg R)}{P(t|\neg R) \cdot P(\neg t|R)} \\ RSV(d,q) &:= \sum_{t \in d \cap q} c_t \end{aligned}$$

For literature on the probabilistic model see, for example, [9], [5], or [10].

Now, we use our matrix framework for describing the probabilistic retrieval model. Consider the following document-term matrix:

$$DT_c = \left(\begin{array}{c|cc} & t_1 & t_2 \\ \hline d_1 & 1 & 1 \\ d_2 & 1 & 0 \\ d_3 & 0 & 1 \\ d_4 & 1 & 1 \\ d_5 & 1 & 0 \\ d_6 & 0 & 1 \end{array} \right)$$

For the document-assessment matrix, let d_1 and d_6 be retrieved and relevant documents, whereas d_2, d_3, and d_5 are retrieved but not relevant. d_4 is not retrieved. We store this information in an assessment matrix where the matrix elements are normalised such that each column sum is equal to one.

$$DA_q = \left(\begin{array}{c|cc} & ret \wedge rel & ret \wedge \neg rel \\ \hline d_1 & 1/2 & 0 \\ d_2 & 0 & 1/3 \\ d_3 & 0 & 1/3 \\ d_4 & 0 & 0 \\ d_5 & 0 & 1/3 \\ d_6 & 1/2 & 0 \end{array} \right)$$

The equation

$$TA_q = {DT_c}^T \cdot DA_q$$

yields the following term-assessment matrix:

$$TA_q = \left(\begin{array}{c|cc} & ret \wedge rel & ret \wedge \neg rel \\ \hline t_1 & 1/2 & 2/3 \\ t_2 & 2/2 & 1/3 \end{array} \right)$$

Here, term t_1 occurs in one of the two retrieved and relevant documents, and in two of the three retrieved but not relevant documents. Term t_2 occurs in all retrieved and relevant documents, and in one of the retrieved but not relevant documents. The term-assessment matrix TA_q has the following probabilistic semantics:

$$TA_q = \left(\begin{array}{c|cc} & ret \wedge rel = R & ret \wedge \neg rel = \neg R \\ \hline t_1 & P(t_1|q,R) & P(t_1|q,\neg R) \\ t_2 & P(t_2|q,R) & P(t_1|q,\neg R) \end{array} \right)$$

The equation

$$NTA_q = 1 - TA_q$$

yields the probabilities $P(\neg t|R)$ and $P(\neg t|\neg R)$.

By rewriting c_t as follows

$$\begin{aligned} c_t &= \log \frac{P(t|R)P(\neg t|\neg R)}{P(t|\neg R)P(\neg t|R)} \\ &= \log P(t|R) + \log P(\neg t|\neg R) - \\ &\quad \log P(t|\neg R) - \log P(\neg t|R) \end{aligned}$$

we can not use the TA_q and NTA_q matrices for computing c_t. Let $\log \pi[1](TA_q)$ apply the logarithm to each element of the first column of TA_q, i. e. π projects on the first column of TA_q. We obtain:

$$\begin{aligned} C = [c_t] &= \log(\pi[1](TA_q)) + \\ &\quad \log(\pi[2](NTA_q) - \\ &\quad \log(\pi[2](TA_q)) - \\ &\quad \log(\pi[1](NTA_q)) \end{aligned}$$

With $\vec{dq}$ being a vector of document-query terms (i. e. $t \in d \cap q$), we obtain the retrieval status values according to the probabilistic relevance feedback model.

$$RSV(d,q) = C_T \cdot \vec{dq}$$

We have shown the description of the probabilistic relevance feedback model based on our matrix framework. The next research challenge is to investigate the duality of the probabilistic relevance feedback model, in which the logarithm on a probability is applied and the relationship of disjoint matrix columns is exploited (*retrieved* $\wedge$ *relevant* and *retrieved* $\wedge\neg$ *relevant* are disjoint), to the other spaces and matrices introduced in this paper.

7 Summary

We have defined a general matrix framework for describing key concepts of information retrieval. We

considered three spaces: a collection space, a document space, and a query result space. Each space is associated with two dimensions, for each dimension we consider an adjacent (parent-child) matrix.

The benefit of our approach is that we achieve a high level of reusability and abstraction in modelling information retrieval and building retrieval systems. The dualities we presented include that the similarity measures as known for the document-term matrix of a collection correspond to precision-recall measures in the query result space, and that the link-based retrieval techniques correspond to Eigenvectors on the matrices of the document dimension in the collection space.

Matrix operations have a close link to relational algebra. The framework presented here paves the way for modelling IR on the layer of relational algebra, and thus we achieve a strong integration between retrieval and database technology. With this framework, we make the construction of IR systems more efficient, and in the end we can build more effective and personalised retrieval systems since the costs for building IR systems are reduced when having a logical model such as this general matrix framework available.

Acknowledgment: TO BE INCLUDED IN PUBLICATION.

References

[1] G. Amati and C. J. Rijsbergen. Term frequency normalization via Pareto distributions. In F. Crestani, M. Girolami, and C. J. Rijsbergen, editors, *24th BCS-IRSG European Colloquium on IR Research, Glasgow, Scotland*, 2002.

[2] Gianni Amati and C. J. van Rijsbergen. Semantic Information Retrieval. In F. Crestani, M. Lalmas, and C. J. van Rijsbergen, editors, *Information Retrieval: Uncertainty and Logics - Advanced models for the representation and retrieval of information*, pages 189–219. Kluwer Academic Publishers, 1998.

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[4] Richard K. Belew. *Finding out about*. Cambridge University Press, 2000.

[5] N. Fuhr and C. Buckley. A probabilistic learning approach for document indexing. *ACM Transactions on Information Systems*, 9(3):223–248, 1991.

[6] David A. Grossman and Ophir Frieder. *Information Retrieval: Algorithms and Heuristics*. Kluwer, Massachusetts, 1998.

[7] J. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46, 1999.

[8] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[9] S.E. Robertson and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[10] I. Ruthven. *Abduction, explanation and relevance feedback*. PhD thesis, University of Glasgow, 2001.

[11] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[12] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, 2. edition, 1979. http://www.dcs.glasgow.ac.uk/Keith/Preface.html.

[13] S.K.M. Wong and Y.Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.

[14] S.K.M. Wong, W. Ziarko, and P.C.N. Wong. Generalized vector space model in information retrieval. In *Proceedings of the 8th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 18–25, New York, 1985. ACM.