Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information Processing and Management 42 (2006) 4–30
INFORMATION PROCESSING & MANAGEMENT
www.elsevier.com/locate/infoproman

A general matrix framework for modelling Information Retrieval

Thomas Rölleke *, Theodora Tsikrika, Gabriella Kazai

Department of Computer Science, Queen Mary University of London, London E1 4NS, UK

Accepted 12 November 2004
Available online 5 January 2005

Abstract

In this paper, we present a well-defined general matrix framework for modelling Information Retrieval (IR). In this framework, collections, documents and queries correspond to matrix spaces. Retrieval aspects, such as content, structure and semantics, are expressed by matrices defined in these spaces and by matrix operations applied on them. The dualities of these spaces are identified through the application of frequency-based operations on the proposed matrices and through the investigation of the meaning of their eigenvectors. This allows term weighting concepts used for content-based retrieval, such as term frequency and inverse document frequency, to translate directly to concepts for structure-based retrieval. In addition, concepts such as pagerank, authorities and hubs, determined by exploiting the structural relationships between linked documents, can be defined with respect to the semantic relationships between terms. Moreover, this mathematical framework can be used to express classical and alternative evaluation measures, involving, for instance, the structure of documents, and to further explain and relate IR models and theory. The high level of reusability and abstraction of the framework leads to a logical layer for IR that makes system design and construction significantly more efficient, and thus, better and increasingly personalised systems can be built at lower costs.

Keywords: Information Retrieval; Content; Structure; Semantics; Matrix spaces; Frequency-based operations; tf-idf; Evaluation measures; IR models; Eigenvectors

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 of building complex information systems that provide ranking

* Corresponding author.
E-mail addresses: thor@dcs.qmul.ac.uk (T. Rölleke), theodora@dcs.qmul.ac.uk (T. Tsikrika), gabs@dcs.qmul.ac.uk (G. Kazai).

0306-4573/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.ipm.2004.11.006

functionality. In this paper, we present a matrix framework in which key Information Retrieval (IR) concepts (Baeza-Yates & Ribeiro-Neto, 1999; Belew, 2000; Grossman & Frieder, 1998; van Rijsbergen, 1979) are described. This matrix framework supports the construction of efficient, flexible and robust search systems, since the matrix operations provide a high level of reusability and abstraction. For a search system engineer, this flexibility of retrieval and indexing functions is crucial, since it yields the possibility to tune the effectiveness and efficiency of a system for the particular personalised needs of end users.

The major theoretical foundations and motivations for this framework include the generalised vector-space model (Wong & Yao, 1995) and the probabilistic framework (Wong & Yao, 1995) for IR. Furthermore, research on the duality of document indexing and relevance feedback (Amati & van Rijsbergen, 1998), on term frequencies normalisation (Amati & van Rijsbergen, 2002) and on link analysis ranking algorithms for Web IR (Kleinberg, 1999; Page, Brin, Motwani, & Winograd, 1998) motivated the development of our matrix framework. While these works, however, address the formalisation of either content or structure, we propose a general matrix framework for both content and structure together with semantics. In addition, we include the modelling of evaluation measures and of retrieval models.

Throughout the paper, particular emphasis is given to a well-defined notation of matrix norms and operations. This allows for the dualities of the matrices and spaces defined within the framework to be systematically explored. For instance, widely used frequencies, such as tf-idf term weighting used for content-based retrieval, can be applied on the structure of collections or documents. On the other hand, concepts, such as pagerank, authorities and hubs, determined by the relationships between the documents of a collection (the collection structure), can be transferred to the relationships between the terms of a collection (the collection semantics).

The paper is structured as follows. Section 2 introduces the content-based, structure-based and semantic-based aspects of retrieval. We consider collection, document, and query matrix spaces and define matrices and operations on them to express these retrieval aspects. Section 3 proposes frequency-based operations for expressing basic content-based IR concepts, such as term frequency and inverse document frequency. Section 4 shows how to use the general matrix framework for modelling classical and alternative evaluation measures. In Section 5, the general matrix framework is used for the modelling of retrieval models. Finally, Section 6 examines the meaning of the eigenvectors of the symmetric matrices and shows the dualities between collection, document and query spaces.

2. Retrieval aspects expressed in the matrix spaces

The underlying framework of our general IR model consists of matrix spaces, matrices associated with each element of a space and standard linear algebra operations on matrices. We consider three matrix spaces: a collection space, a document space and a query space. Each space may contain several elements. For example, the collection space contains collections and the document space contains documents. Each space has two dimensions. For example, the collection space has document and term dimensions, each represented by a vector. For each element of a space, we introduce matrices to represent the relationships between pairs of elements of the two dimensions of the space and matrices to represent the parent–child relationships [1] between pairs of elements of a single dimension of the space.

We propose a carefully chosen notation for indicating the spaces and their associated matrices. In our notation, a space is represented by a lower case letter and its dimensions by capital case letters. Let us consider a space s and its dimensions X and Y. The vectors X_s, Y_s contain the elements of the dimensions. The

[1] The terminology *parent–child relationship* is used to describe any directed association between a source (*parent*) and a target (*child*), i.e. it is not restricted to (tree-like) hierarchical relationships.

matrix XY_s reflects the relationships between pairs of elements of the two dimensions of this space. This notation further denotes that the elements of dimension X are represented as rows of the matrix XY_s, whereas the elements of dimension Y are represented as columns of the matrix XY_s. To represent the matrix, which reflects the parent–child relationships among the elements of one dimension, we use matrices named PC for parent–child, carrying a subscript to indicate the dimension and the space. To represent, for instance, the parent–child relationships among the elements of the X dimension in space s, we use the PC_{X_s} matrix.

In this framework, collections, documents and queries correspond, respectively, to the following matrix spaces: the collection space c, the document space d, and the query space q. Retrieval aspects, such as content, structure and semantics, are expressed by matrices defined in these spaces. These matrices are discussed in the following sections.

2.1. Content

In our matrix framework, the content of a collection is represented by the document–term matrix DT_c of the collection space and the content of a document is represented by the location–term matrix LT_d of the document space. We choose the terminology *location* to cover concepts indicating document components of varying granularity, such as *section*, *paragraph*, and *position*. Within the query space, the "content" of a query is defined in terms of the relevance assessments provided for the query, and, hence, is represented by the document-assessor matrix DA_q or the location-assessor matrix LA_q. Table 1 shows the content matrices associated with the collection space and the document space, whereas Table 2 shows the content matrices associated with the query space. Next, we discuss in detail the content matrices of our spaces.

2.1.1. Collection space

In a collection space c, the two dimensions are documents D and terms T. We define the vector of documents in the collection as $D_c = [w_{d_i}]_{N\times 1}$, where $w_{d_i} \geqslant 0$ is the weight of document d_i. This weight can be

Table 1
Content of collection and document spaces

Content	
Collection space	Document space
DT_c: Documents × terms	LT_d: Locations × terms
$DD_c = DT_c \cdot DT_c^T$ Document similarity (term degree)	$LL_d = LT_d \cdot LT_d^T$ Location similarity (term degree)
$TT_c = DT_c^T \cdot DT_c$ Term similarity (document degree)	$TT_d = LT_d^T \cdot LT_d$ Term similarity (location degree)

Table 2
Content of query space

Content	
Query space	
DA_q: Documents × assessors	LA_q: Locations × assessors
$DD_q = DA_q \cdot DA_q^T$ Document similarity (assessor degree)	$LL_q = LA_q \cdot LA_q^T$ Location similarity (assessor degree)
$AA_q = DA_q^T \cdot DA_q$ Assessor similarity (document degree)	$AA_q = LA_q^T \cdot LA_q$ Assessor similarity (location degree)

used to define the importance of a document in the collection. It can be estimated by taking into account the source of the document, its size, the number of incoming and outgoing links (in the case of hyperlinked documents) or other available evidence. In the simple case, the document weight denotes just the presence (occurrence) of the document in the collection:

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

In this case, the L_1-norm (Golub & van Loan, 1996) of a document vector, defined as $\|D_c\|_1 \equiv \sum_{i=1}^{N} |w_{d_i}|$, represents the number of documents in the collection: $N_{D_c} = \|D_c\|_1$.

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

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

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

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

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

where $1 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant M$.

We can consider the product of the collection's content matrix DT_c and its transpose. [2] Using post-multiplication, we generate the $DD_c = DT_c \cdot DT_c^T$ matrix, its elements reflecting document similarity within the collection. To be more specific, the ijth element of matrix DD_c expresses the similarity of documents d_i and d_j, as this is reflected by the overlap in their term occurrences. Using pre-multiplication, on the other hand, the $TT_c = DT_c^T \cdot DT_c$ matrix is produced, its elements representing the number of common documents containing each pair of terms and thus reflecting term similarity within the collection. [3]

2.1.2. Document space

Similarly to the description of the collection space, the two dimensions of the document space d are locations L and terms T. We define the vector of locations in the document as $L_d = [w_{l_i}]_{R\times 1}$, $w_{l_i} \geqslant 0$ and the vector of terms in the document as $T_d = [w_{t_i}]_{S\times 1}$, $w_{t_i} \geqslant 0$.

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

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

where $1 \leqslant i \leqslant R$ and $1 \leqslant j \leqslant S$.

Again, we compute the product of the document content matrix LT_d and its transpose, to generate the $LL_d = LT_d \cdot LT_d^T$ matrix using post-multiplication and the $TT_d = LT_d^T \cdot LT_d$ matrix using pre-multiplication. The elements of the LL_d matrix reflect location similarity within the document with respect to the overlap in

[2] Generally speaking, an element y_{ij} of a matrix Y, generated by post-multiplying a matrix X by its transpose X^T, i.e. $Y = X \cdot X^T$, corresponds to the dot product of the rows i, j of the X matrix. Similarly, an element z_{ij} of a matrix Z, generated by pre-multiplying a matrix X by its transpose X^T, i.e. $Z = X^T \cdot X$, corresponds to the dot product of the columns i, j of the X matrix.

[3] We apologise for the double meaning of the letter T, namely for the term dimension and for matrix transposition. We considered alternative notations for the transposition, but decided finally that it is best to keep the common notation, since transposition is always an exponent ^{T}, whereas the term dimension is always part of a matrix name.

their term occurrences, whereas the elements of the TT_d matrix reflect term similarity with respect to overlap in the locations containing them.

2.1.3. Query space

We consider two content matrices associated with the query space: the document-assessor matrix DA_q and the location-assessor matrix LA_q (Table 2). Both matrices represent relevance assessments made by assessors (elements of dimension A) over the set of documents or locations in the query space (elements of dimension D or L, respectively). We consider both relevance judgements made by human assessors and estimations of relevance produced by retrieval systems simply as relevance assessments. The DA_q matrix reflects traditional IR (e.g. document retrieval), where the unit of retrieval is the document, whereas the LA_q matrix represents assessments at location (e.g. document component) level. This view reflects structured document retrieval, where any location may serve as a retrieval unit. Note that the space of the locations here is the query, meaning that components from different documents may be included. In addition, the documents themselves may be viewed as types of locations. Therefore, the DA_q matrix can be viewed as a special case of the LA_q matrix.

We define the vector of documents in the query space as $D_q = [w_{d_i}]_{K\times 1}$, where $w_{d_i} \geqslant 0$ is the weight of document d_i. This weight may combine different query-specific or query-independent evaluation parameters or may in the simple case represent the presence (or absence) of documents in the query space:

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

For the assessor dimension, we define the vector of assessors as $A_q = [w_{a_i}]_{L\times 1}$, where $w_{a_i} \geqslant 0$ is the weight associated with an assessor a_i. The weights associated with an assessor may reflect its quality or trust value, or may in the simple case be:

$$w_{a_i} := \begin{cases} 1 & \text{if } a_i \text{ has provided assessments for the query} \\ 0 & \text{if } a_i \text{ has not provided assessments for the query} \end{cases}$$

Let $DA_q = [da_{ij}]_{K\times L}$, be the matrix of document-assessor information associated with a query, where rows correspond to documents and columns to assessors. We define each matrix element as:

$$da_{ij} := \begin{cases} 1 & \text{if } d_i \text{ is judged relevant by assessor } a_j \\ 0 & \text{if } d_i \text{ is judged not relevant by assessor } a_j \end{cases}$$

where $1 \leqslant i \leqslant K$ and $1 \leqslant j \leqslant L$. In the general case, the weight of an element in the DA_q matrix can reflect the relevance degree or relevance status value of a document to the query as judged by an assessor (human judge or retrieval system).

Similarly to the above definitions, the vector of locations in the query space is defined as $L_q = [w_{l_i}]_{V\times 1}$, $w_{l_i} \geqslant 0$, the assessors dimension as $A_q = [w_{a_i}]_{U\times 1}$, $w_{a_i} \geqslant 0$, and the $LA_q = [la_{ij}]_{V\times U}$ matrix is defined in the simple case as:

$$la_{ij} := \begin{cases} 1 & \text{if } l_i \text{ is judged relevant by assessor } a_j \\ 0 & \text{if } l_i \text{ is judged not relevant by assessor } a_j \end{cases}$$

where $1 \leqslant i \leqslant V$ and $1 \leqslant j \leqslant U$.

The product of these matrices and of their transpose provides, on the one hand a representation for document and location similarity expressing assessor agreement (DD_q and LL_q), while the AA_q matrices reflect assessor similarity and provide a framework for the calculation of precision/recall measures (see Section 4).

Next we define the matrices PC_{X_s} reflecting the parent–child relationships between the elements of a single dimension X of a matrix space x. We define the elements of these matrices as $pc_{ij} = 1$ if element x_i is

parent of element x_j and 0 otherwise. For the scope of this paper, we restrict ourselves to the relationships of the elements along the dimensions of the collection space and the document space. First, we consider the document and location dimensions in the collection and document space, respectively (Section 2.2), and then the term dimensions in these two spaces (Section 2.3).

2.2. Structure

The parent–child relationships among the documents in the collection space and the locations in the document space constitute the structure of a collection and a document, respectively. Table 3 shows the modelling of these relationships. The matrix PC_{D_c} in a collection space could represent the link-structure of a Web document collection (collection structure), while the PC_{L_d} matrix in the document space could represent the relationships among document parts (document structure).

By multiplying the PC_{D_c} matrix with its transpose, the matrices $PP_{D_c} = PC_{D_c} \cdot PC_{D_c}^T$ and $CC_{D_c} = PC_{D_c}^T \cdot PC_{D_c}$ are generated, whose elements reflect document parent and document child similarity, respectively. An element of PP_{D_c} is also referred to as *bibliographic coupling degree*, i.e. it reflects the degree to which two documents cite the same children. An element of CC_{D_c} is referred to as *co-citation degree*, i.e. it reflects the degree to which two documents are cited by the same parents. These are measures of the similarity of two pages and whereas the terminology was initially introduced in the field of bibliometric studies, it has been adopted in the field of link analysis algorithms in Web IR, by considering that the links between documents act as citations. One of the most prominent link analysis algorithms is HITS (Kleinberg, 1999), which considers that there are two types of quality Web pages: *authorities*, which contain definitive, high-quality information and *hubs*, which are comprehensive lists of links to authorities. Every page is viewed as being to some extent both a hub and an authority. These hub and authority values correspond to the principal eigenvectors of the PP_{D_c} and CC_{D_c} matrices, respectively (see Section 6 for discussion on the meaning of the eigenvectors of the matrices defined in the framework).

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

2.3. Semantics

Table 4 shows the modelling of the parent–child relationships among the terms of the collection space and the terms of the document space, respectively. In this case, the relationships among the terms constitute the semantics in a collection and a document, respectively. The semantics in the collection space is reflected in PC_{T_c}, from which we derive $PP_{T_c} = PC_{T_c} \cdot PC_{T_c}^T$ and $CC_{T_c} = PC_{T_c}^T \cdot PC_{T_c}$, representing the term parent and term child similarities. Similarly, PC_{T_d} reflects the semantics in the document space, from which we derive PP_{T_d} and CC_{T_d} to denote respectively the term parent and term child similarities in a document.

Table 3
Structure of collection space and document space

Structure	
Collection space	Document space
PC_{D_c}: Parents × children	PC_{L_d}: Parents × children
$PP_{D_c} = PC_{D_c} \cdot PC_{D_c}^T$ Out-degree of documents	$PP_{L_d} = PC_{L_d} \cdot PC_{L_d}^T$ Out-degree of locations
$CC_{D_c} = PC_{D_c}^T \cdot PC_{D_c}$ In-degree of documents	$CC_{L_d} = PC_{L_d}^T \cdot PC_{L_d}$ In-degree of locations

Table 4
Semantics of collection space and document space

Semantics	
Collection space	Document space
PC_{T_c}: Parents × children	PC_{T_d}: Parents × children
$PP_{T_c} = PC_{T_c} \cdot PC_{T_c}^T$ Generality of terms	$PP_{T_d} = PC_{T_d} \cdot PC_{T_d}^T$ Generality of terms
$CC_{T_c} = PC_{T_c}^T \cdot PC_{T_c}$ Specificity of terms	$CC_{T_d} = PC_{T_d}^T \cdot PC_{T_d}$ Specificity of terms

From the PP_{T_c} and the PP_{T_d} matrices, we can estimate the "authority" and "hub" value of a term. An "authority" term is a term with a high number of incoming links, i.e. it 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 specialisations). 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. This hub and authority view on terms could be combined with the term similarity matrix TT_c (Section 2.1.1) to add a semantic aspect to an otherwise purely occurrence-based similarity measure.

So far, we have described how the retrieval aspects are expressed in the spaces of our matrix framework. The notation we have introduced allows for a general IR model with high abstraction. In the next section, we present the frequency-based operations on the matrices of our framework, which allows for the modelling of classical term weighting schemes, such as term frequency and inverse document frequency, using the content matrices of the collection and the document space.

3. Frequency-based operations on matrices

Our aim is to describe key IR concepts using our matrix framework. In this section, we propose frequency-based operations on matrices. We focus on the exploitation and definition of widely used frequencies, such as the classical term weighting schemes used for content-based retrieval. This section investigates how the inverse document frequency of terms can be described using a collection's content matrix (Section 3.1) and how the location frequency of terms (commonly referred to as term frequency in IR) can be described using a document's content matrix (Section 3.2).

3.1. Collection space

The common IR 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 the collection in which term t occurs.

Following these definitions, we provide the definition of the inverse document frequency in our matrix framework. We describe the inverse document frequency *idf* of a term in a collection, by defining a vector of the terms in the collection, where each element is the number of documents that contain the term:

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

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:

$$D_c = \left(\begin{array}{ccc} d_1 & d_2 & d_3 \\ \hline 1 & 1 & 1 \end{array}\right)^T \qquad T_c = \left(\begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \\ \hline 1 & 1 & 1 & 1 \end{array}\right)^T$$

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:

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

For our example, we then obtain:

$$nd_{T_c} = D_c^T \cdot DT_c = \left(\begin{array}{cccc} t_1 & t_2 & t_3 & t_4 \\ \hline 2 & 3 & 2 & 1 \end{array}\right)$$

By normalising each element of nd_{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:

$$df_{T_c} = \frac{1}{N_{D_c}} \cdot nd_{T_c}$$

For our example, this yields:

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

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

$$idf_{T_c} = \text{apply}(-\log, df_{T_c})$$

For our example, we obtain:

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

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*, as traditionally considered in IR. The latter one is dealt with in Section 3.2.

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

$$tf(d,c) := \frac{n_T(d,c)}{N_T(c)} \tag{3}$$

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

where $N_T(c)$ is the number of terms in the collection and $n_T(d,c)$ is the number of terms occurring in document d. Note the correspondence between the definition of document frequency (Eq. (1)) and term frequency (Eq. (3)) and between the definition of inverse document frequency (Eq. (2)) and inverse term frequency (Eq. (4)).

Therefore, we can analogously define the following three vectors: a vector of the documents in the collection, where each element is the number of terms occurring in the document: $nt_{D_c} = DT_c \cdot T_c = [nt_{d_i}]_{N\times 1}$, a vector of the documents in the collection, where each element is the term frequency of the document: $tf_{D_c} = \frac{1}{N_{T_c}} \cdot nt_{D_c}$ and a vector of the documents in the collection where each element is the inverse term frequency of the document: $itf_{D_c} = \text{apply}(-\log, tf_{D_c})$. For our example, we obtain:

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

The inverse document frequency reflects the so-called *discriminative power* (occurrence) of a term, the inverse term frequency reflects the *specificity* (length) of a document. Note the perfect mathematical analogy between document and term frequency. However, there is a terminological misfit with the common IR definition of term frequency (where term frequency is used for a term in a document) and the term frequency of a document used here. 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 the next section.

3.2. Document space

Let a document with content such as "sailing boats greece sailing" be given. The location (where location in this case corresponds to a term position in the document) and the term vectors of this document are then defined as:

$$L_d = \left(\begin{array}{cccc} l_1 & l_2 & l_3 & l_4 \\ \hline 1 & 1 & 1 & 1 \end{array}\right)^T \qquad T_d = \left(\begin{array}{ccc} t_1 & t_2 & t_3 \\ \hline 1 & 1 & 1 \end{array}\right)^T$$

with t_1 = "sailing", t_2 = "greece" and t_3 = "boats". 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.

Let $n_L(t,d)$ be the number of locations at which t occurs in d. Then, the common IR definition of the term frequency is as follows:

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

where $n_L(t_{\max},d)$ is the maximal occurrence, i.e. $\forall t$: $n_L(t,d) \leqslant n_L(t_{\max},d)$.

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

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

For our example, we obtain:

$$nl_{T_d} = \left(\begin{array}{ccc} t_1 & t_2 & t_3 \\ 2 & 1 & 1 \end{array}\right)$$

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: $lf_{T_d} = \frac{nl_{T_d}}{\|nl_{T_d}\|_\infty}$ (where $\|\cdot\|_\infty$: $L_{\infty-\text{norm}}\|\vec{x}\|_\infty \equiv \max_i|x_i|$ (Golub & van Loan, 1996)). 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:

$$lf_{T_d} = \left(\begin{array}{ccc} t_1 & t_2 & t_3 \\ \frac{2}{2} & \frac{1}{2} & \frac{1}{2} \end{array}\right)$$

However, there are several alternative approaches for estimating the classical IR notion of term frequency. These approaches are based on the idea of lifting the probabilities of rare terms. Experiments (see Robertson, Walker, & Hancock-Beaulieu, 1995; Salton & Buckley, 1988 and related publications) prove that such approaches improve retrieval quality. We can distinguish between two main approaches: (1) linear lifting and (2) non-linear lifting using a Poisson approximation.

In the first approach, the lifting to the interval $a \leqslant tf(t,d) \leqslant 1$ is described by:

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

On the other hand, the estimate based on a Poisson approximation leads to a non-linear increase of *tf*-values. In addition, a document length normalisation can be considered, while *tf*-values still remain in the interval $[0,1]$, which is a welcome property in a probabilistic framework. This is described by:

$$tf(t,d) := \frac{n_L(t,d)}{K + n_L(t,d)}$$

The definition of K includes parameters such as k_1 and b for controlling the influence of K itself and of the document length normalisation:

$$K := k_1 \cdot \left((1-b) + b\cdot\frac{N_L(d)}{\text{avg}N_L(d)}\right)$$

The parameter K is small for small documents and large for large documents. Small k_1 values lead already for relatively small $n_L(t,d)$ values to large *tf*-values, while a small b reduces the impact of the length normalisation.

Whereas linear lifting requires just a multiplication of the involved matrices with a scalar, in order to be expressed in the matrix framework, the Poisson-based non-linear lifting requires more complex operations. The inclusion of non-linear lifting is a topic of future research in extending the matrix framework.

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 nt_{L_d}, where each element is the number of terms occurring in the

location: $nt_{L_d} = LT_d \cdot T_d = [nt_{l_i}]_{R\times 1}$ and the vector of the locations in the document tf_{L_d}, where each element is the term frequency of the location:

$$tf_{L_d} = \frac{nt_{L_d}}{\|nt_{L_d}\|_\infty}$$

We have discussed term weighting in the collection space and in the document space by applying frequency-based operations on the content matrices of these spaces. Note that similar frequency-based operations could be applied to the structure and semantics matrices of the collection and document spaces. Therefore, *tf-idf* like measures may be applied to gauge the level of structural or semantic dependency. Next, we discuss evaluation measures in the query space.

4. Evaluation

We show in this section how to express evaluation measures in our general matrix framework. This integration of evaluation concepts within our framework allows to fully realise and exploit the duality of the meanings of the applied matrix operations within the different spaces. For example, we can build a similarity matrix for assessors just as we did for terms or documents, or we can apply the notion of precision and recall for terms in a collection space DT_c^T (representing the ratio of the number of co-occurring terms in two documents to the length of the individual documents, respectively).

First, we discuss the standard precision and recall measures, then we extend the discussion and demonstrate how to use the collection and document structure matrices (PC_{D_c} and PC_{L_d}) for expressing concepts such as aggregated relevance and novelty-based evaluation.

4.1. Precision and recall

Precision and recall (Baeza-Yates & Ribeiro-Neto, 1999; van Rijsbergen, 1979) are the most common quality evaluation measures in IR, and are defined as follows:

$$\text{precision} := \frac{\text{retrieved} \cap \text{relevant}}{\text{retrieved}} \qquad \text{recall} := \frac{\text{retrieved} \cap \text{relevant}}{\text{relevant}}$$

The description of precision/recall in our matrix framework is based on the document-assessor DA_q and location-assessor LA_q matrices of a query as introduced in Section 2.1.3.

First we consider the DA_q matrix and define precision and recall when documents represent the atomic unit of retrieval. In our definition, we make use of the notation $DA_q(:, a_i)$ (Golub & van Loan, 1996) denoting the ith assessor column of the DA_q matrix. Each such column of DA_q represents the assessments of an assessor a_i (i.e. retrieval system or human judge) over the documents of the query space. The L_1-norm of a column vector then gives the number of documents that a given assessor judged relevant. [4] The number of retrieved and relevant documents is the number of documents that have been assessed relevant by both assessors (i.e. by the system under investigation and by the human judge, whose assessment is considered as the ground truth that the system is evaluated against). This can be calculated simply as the dot product of the two column-vectors. Based on these, precision and recall can be obtained as:

$$\text{precision}(a_i, a_j) := \frac{DA_q(:, a_i)^T \cdot DA_q(:, a_j)}{\|DA_q(:, a_i)\|_1}$$

[4] We assume binary relevance assessments here. For graded assessments (or RSVs) the L_1-norm gives the sum of the relevance scores, which can then be used to calculate generalised precision and recall (Kekalainen & Jarvelin, 2002).

$$\text{recall}(a_i, a_j) := \frac{DA_q(:, a_i)^T \cdot DA_q(:, a_j)}{\|DA_q(:, a_j)\|_1}$$

The dot product of any two assessment vectors can also be obtained directly from an assessor–assessor matrix AA_q, derived from DA_q using pre-multiplication (e.g. as we have done when obtaining a term–term matrix from the document–term matrix):

$$AA_q = DA_q^T \cdot DA_q$$

Given AA_q, the precision/recall values for an assessor a_i evaluated against the ground truth of the assessment of a_j can be calculated as:

$$\text{precision}(a_i, a_j) := \frac{AA_q(a_i, a_j)}{\|DA_q(:, a_i)\|_1} \qquad \text{recall}(a_i, a_j) := \frac{AA_q(a_i, a_j)}{\|DA_q(:, a_j)\|_1}$$

In the special case of binary assessment values, $\|DA_q(:, a_i)\|_1$ and $\|DA_q(:, a_j)\|_1$ can also be directly obtained from the AA_q matrix as $AA_q(a_i, a_i)$ and $AA_q(a_j, a_j)$, respectively.

The generalisation of the AA_q matrix is that it provides a complete summary of the evaluation of any system or user assessment against any other system or user assessment, giving the relative performance of one assessor against another. In addition, any element within the matrix reflects the relative similarity between assessments and/or retrieval strategies.

The above definitions reflect the set view on retrieval results. However, a retrieval system returns a ranked list of documents rather than a set of documents. We can capture the ranking information and calculate precision/recall at a given rank by taking the sub-matrix of the DA_q assessment matrix containing documents retrieved up to that rank.

Similarly to the calculations based on the DA_q matrix, we can obtain the precision and recall measures for structured document retrieval systems whose assessments, consisting of varying granularity document components, are described within a LA_q matrix. The assessor–assessor matrix in this case reflects assessment similarity at a location level (i.e. location degree).

So far in this section, we have sketched 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 may want to consider the efficiency of query processing. We may then introduce an additional assessment column in DA_q or LA_q, where the column reflects, for instance, the time at which a document is delivered by a system. Numerous other factors may be considered in system evaluation, including the structure of documents, which we investigate next.

4.2. Evaluation measures for linked documents

In this section, we look at the extension of our assessment matrices for situations where dependencies (e.g. links) among the documents of a collection or among document parts within a document exist. By exploiting the dependency information, we can extend traditional evaluation measures to consider the retrieval of indirectly relevant documents as partial successes or to penalise the retrieval of multiple related, and hence redundant, documents.

4.2.1. Near-misses

We use the term "near-miss" to refer to a document (or location), which is not itself relevant to a given query, but which is linked to one or more relevant documents (or locations). The rationale behind incorporating a mechanism within the evaluation to score the retrieval of near-misses is that such documents may still be considered useful for the user (especially if the relevant document itself is not found by the search engine) (Hawking, Voorhees, Craswell, & Bailey, 1999). With this aim, we describe the process of

relevance propagation, which supports an evaluation framework, where additional (partial) scores may be rewarded for near-misses.

Given a link structure PC_{D_c} of documents within a collection and the document-assessor matrix DA_q for the same set of documents, we can propagate relevance along the links, in a given direction, reflecting the notion that if a document is relevant, then a document linked to or from it may also be considered relevant (to some degree). Previous research has investigated a number of relevance propagation strategies, including pessimistic and optimistic approaches (Roelleke, Lalmas, Kazai, Ruthven, & Quicker, 2002). A pessimistic strategy only considers a linked document relevant if all documents linking to or from it are relevant. For the optimistic propagation it is sufficient if only one of the linked documents is relevant.

For each step of a propagation approach, we first derive a matrix whose elements reflect the number of relevant linked parent (nPA_q) or child (nCA_q) documents (assuming binary assessment values in DA_q), where relevance is propagated to children or parent documents, respectively:

$$nPA_q = PC_{D_c}^T \cdot DA_q \qquad nCA_q = PC_{D_c} \cdot DA_q$$

From these, the propagated child CA_q and parent PA_q assessment matrices can be derived by assigning assessment weights according to the selected propagation strategy. For example, for the optimistic strategy, we can obtain each element of the propagated child assessment matrix as:

$$ca_{ij} := \begin{cases} 1 & \text{if } nPA_q(d_i, a_j) > 0 \\ 0 & \text{otherwise} \end{cases}$$

For the propagated parent assessment matrix, we assign $pa_{ij} := 1$ if $nCA_q(d_i, a_j) > 0$ and 0 otherwise. With respect to the pessimistic strategy, the elements of the propagated child assessment vector are calculated as:

$$ca_{ij} := \begin{cases} 1 & \text{if } nPA_q(d_i, a_j) = \|PC_{D_c}(:, d_i)\|_1 \\ 0 & \text{otherwise} \end{cases}$$

Similarly, for the propagated parent assessment matrix, we assign $pa_{ij} := 1$ if $nCA_q(d_i, a_j) = \|PC_{D_c}^T(:, d_i)\|_1$ and 0 otherwise. According to this, the propagated assessment value of 1 is assigned to a child (or parent), if the number of linked relevant documents equals the number of parent (or child) documents.

Alternative propagation methods may consider other threshold values, and various parameters. One such parameter is the number of linked relevant documents, which may be regarded as a measure of document "relevancy-authority" or "relevancy-hub" value. In addition, as the propagation process is repeated iteratively, the propagated values may be normalised to reflect the distance from the original relevant document. Such a strategy reflects the increasing user effort required in locating relevant documents from returned near-misses. When non-binary relevance scores are propagated, various adaptations of the optimistic and pessimistic strategies can be employed, such as assigning a parent or child document the average or maximum of the linked documents' relevance scores.

The derived propagated assessments matrices, combined with the original DA_q matrix, allow a possible evaluation metric to score near-misses. The definition of such a metric is, however, a non-trivial issue [5] and is outside the scope of this paper.

The exact same procedures can be applied to linked document parts and hence propagate relevance to related components within a document. The structure matrix employed here would be a matrix, which combines all the PC_{L_d} matrices of documents contained in the query space. This can be derived by spanning the PC_{L_d} matrices within the diagonal of the main structure matrix. The resulting framework allows to

[5] Within the standard precision/recall framework, the simple addition of near-misses to the recall-base can lead to skewed effectiveness results, where 100% recall can only be reached if systems return all relevant and near-miss documents (Hawking et al., 1999; Kazai, Lalmas, & de Vries, 2004).

calculate precision/recall for assessments containing varying granularity document components while also allowing to reward near-misses.

4.2.2. Novelty

While, on the one hand, we may want to reward systems for retrieving near-misses, on the other hand, we may also want to discourage systems from returning redundant results, i.e. multiple related documents (or components). In this case, the evaluation should consider the dependency among documents/components in order to score systems based on the novelty value [6] of the returned results.

A simple (heuristic) measure of novelty may be given as the inverse of dependency:

$$\text{novelty} := \frac{1}{\text{dependency} + 1}$$

where dependency may be calculated as:

$$\text{dependency} := \sum_{k=1}^{N_{D_{a_i}}} \|PC_{D_{a_i}}(:, d_k)\|_1 = \sum_{k=1}^{N_{D_{a_i}}} \sum_{j=1}^{N_{D_{a_i}}} PC_{D_{a_i}}(d_j, d_k)$$

The $PC_{D_{a_i}}$ matrix here is the structure matrix derived for documents assessed relevant by a_i, where $N_{D_{a_i}}$ is the number of documents. For locations, we sum over the $PC_{L_{a_i}}$ structure matrix reflecting dependencies among locations assessed relevant by a_i.

The obtained novelty value of 1 reflects independence and high novelty, while values close to 0 represent high dependency and low novelty among assessments.

An exact definition of novelty and its adoption within the recall/precision measures will depend on the objective of the evaluation, which is again outside the scope of this paper. Here our aim is only to highlight that our matrix framework provides a way to extend the classical evaluation approaches to consider the dependency among documents (components) in order to allow rewarding near-misses and novelty. The definition of new evaluation measures based on the above extensions of DA_q and LA_q are especially important for Web and structured document retrieval, where assessments may contain high ratios of related components. The matrix framework provides a formalism in which those new evaluation measures can be established.

5. Retrieval models

In this section, we use our matrix framework for expressing the vector-space model (Section 5.1), the logical approach (Section 5.2), the probabilistic inference network model (Section 5.3), and the probability of relevance models (Section 5.4), where in the latter section we consider the binary independent retrieval model (Section 5.4.1) and language modelling (Section 5.4.2). The main outcome of viewing all models in the matrix framework is to highlight the parallels and dualities of the models.

5.1. Vector-space model

The vector-space model is by its nature straight-forward to formalise in the matrix framework. We start with a binary document–term matrix, consider then tf-idf, and extend the discussion with the generalised vector-space model.

[6] Note that novelty is considered here only with regards to redundancy among returned documents/components and not with respect to the information contained within the documents.

Consider the product $DT_c \cdot DT_c^T$ of the document–term matrix DT_c. The equation

$$DD_c = DT_c \cdot DT_c^T$$

yields in DD_c a similarity measure for each pair of documents. This similarity measure is also referred to as retrieval status value (RSV). Rows of the DT_c matrix constitute document or query vectors, respectively. The notation $DT_c(d_i,:)$ selects document and query vectors. Considering row d_k as a query, we write:

$$\vec{q} = DT_c(d_k,:)^T$$

The equation

$$\begin{pmatrix} \mathrm{RSV}_{\mathrm{VSM}}(d_1,q) \\ \vdots \\ \mathrm{RSV}_{\mathrm{VSM}}(d_n,q) \end{pmatrix} = DT_c \cdot \vec{q} \tag{6}$$

yields a vector of RSV's for query q. Assuming a binary matrix DT_c, this formulation of the VSM in the matrix framework corresponds to the so-called coordination level match, i.e. the RSV corresponds to the number of terms shared by the document and query.

The coordination level match based on a binary matrix DT_c is outperformed by the tf-idf approach. The tf-idf approach is described in the matrix framework by using a DT_c matrix in which the components dt_{ij} correspond to the within-document–term frequency (actually, location frequency, see Section 3.2 for the computation of the location frequency) of term t_j in document d_i. Further, we use the vector idf_{T_c} which contains the idf-values of the terms in collection c (see Section 3.1 for the computation of idf_{T_c}). The tf-idf approach without normalisation is then described as the product of a document–term matrix with location frequencies and the diagonal matrix $\mathrm{diag}(idf_{T_c})$ of *idf*-values.

$$\begin{pmatrix} lf_{T_{d_1}} \\ \vdots \\ lf_{T_{d_n}} \end{pmatrix} \cdot \mathrm{diag}(idf_{T_c}) = \begin{pmatrix} lf_{T_{d_1}} \\ \vdots \\ lf_{T_{d_n}} \end{pmatrix} \cdot \begin{pmatrix} idf(t_1,c) & 0 & \dots & 0 \\ 0 & idf(t_2,c) & & 0 \\ 0 & \dots & 0 & idf(t_n,c) \end{pmatrix}$$

The multiplication of the location frequency matrix and the diagonal matrix of the vector idf_{T_c} yields a $D \times T$ matrix in which dt_{ij} is the tf-idf value of a document–term pair. The L_2-norm (Euclidean norm) applied to each row yields a classical document length normalisation.

One motivation to view IR in the general matrix framework of this paper comes from the work (Wong, Ziarko, & Wong, 1985) on the generalised VSM, in which a matrix G is introduced as follows:

$$\mathrm{RSV}_{\mathrm{GVSM}}(d,q) = \vec{d}^T \cdot G \cdot \vec{q} \tag{7}$$

G is a term $\times$ term matrix that reflects semantic relationships between terms. By setting a matrix element such as g_{12} to 1, we obtain, for example, a revised document vector:

$$\vec{d}^T \cdot G = \begin{pmatrix} dt_1 \\ dt_2 \\ \vdots \\ dt_n \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ & & \ddots & & \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} = \begin{pmatrix} dt_1 \\ dt_1 + dt_2 \\ \vdots \\ dt_n \end{pmatrix}^T$$

Because $g_{12} = 1$, the weight dt_1 [7] of term $\vec{t_1}$ is added to the weight dt_2 of $\vec{t_2}$, and the sum $dt_1 + dt_2$ is the new weight of term $\vec{t_2}$. For a query, the weight qt_2 is added to qt_1, and the sum $qt_1 + qt_2$ becomes the weight of $\vec{t_1}$. In the scalar product of document and query, the factor $g_{12} \cdot dt_1 \cdot qt_2$ is added to the basic scalar product. This generalisation of the scalar product is useful for addressing word-mismatch problems that are beyond stemming. For example, a query for "classification" shall retrieve documents that are indexed with "categorisation", and this can be achieved by setting the corresponding element in G that relates the two terms. With respect to the matrices introduced in this paper, the TT_c matrix for term similarity or the PC_{T_c} matrix of term relationships can be viewed as settings for the generalisation matrix G.

The general vector-space model has an interesting relationship with the logical approach to IR, which is highlighted in the next section.

5.2. Logical approach $P(d \rightarrow q)$

In van Rijsbergen (1986), a logical approach for IR is proposed. The idea is to define a logic such that the probability $P(d \rightarrow q)$ is a good estimate of the probability of relevance. We show in the context of the matrix framework, how an interpretation of $P(d \rightarrow q)$ as conditional probability $P(q|d)$ relates to the generalised vector-space model. Based on a set of disjoint terms (see Wong & Yao, 1995), the conditional probability is expressed as the sum over the product of query and term probabilities.

$$P(d \rightarrow q) := P(q|d) = \sum_t P(q|t) \cdot P(t|d)$$

Here, we assume that $P(q|t) = P(q|d,t)$, i.e. given term t, the query does not depend on the document. Using Bayes for $P(t|d)$, we rewrite the equation and obtain:

$$P(q|d) = \frac{1}{P(d)} \cdot \sum_t P(q|t) \cdot P(d|t) \cdot P(t)$$

Now, consider vectors $\vec{q} = (P(q|t_1), \ldots, P(q|t_n))^T$ and $\vec{d} = (P(d|t_1), \ldots, P(d|t_n))^T$ for query and document. Then, we can write $P(q|d)$ in a form similar to the generalised vector-space model:

$$P(q|d) = \frac{1}{P(d)} \cdot \vec{d}^T \cdot \mathrm{diag}(P(t_1), \ldots, P(t_n)) \cdot \vec{q} \tag{8}$$

The diagonal matrix of term probabilities connects query and document vector. Query and document vector have the same semantics, i.e. both contain probabilities depending on a term.

We have shown how to express $P(q|d)$ in our matrix framework, and that the modelling is related to the generalised vector-space model. Next, we look at the computation of $P(q|d)$ based on probabilistic inference networks.

5.3. Probabilistic inference network (PIN) model

Fig. 1 shows a PIN. The general computation of $P(q)$ in the PIN is based on the link matrix L and the vector of incoming probabilities, where each combination of incoming events has to be considered.

$$\begin{pmatrix} P(q|d) \\ P(\bar{q}|d) \end{pmatrix} = L \cdot \begin{pmatrix} P(t_1, t_2|d) \\ P(t_1, \bar{t_2}|d) \\ P(\bar{t_1}, t_2|d) \\ P(\bar{t_1}, \bar{t_2}|d) \end{pmatrix}$$

[7] dt_i is here the scalar in row t_i of vector $\vec{d}$.

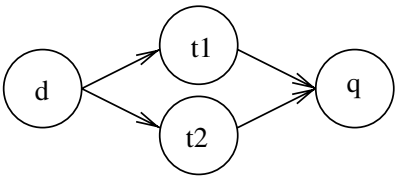

Fig. 1. A probabilistic inference network.

The link matrix L contains conditional probabilities of the form

$$L = \begin{pmatrix} P(q|t_1,t_2) & P(q|t_1,\bar{t_2}) & P(q|\bar{t_1},t_2) & P(q|\bar{t_1},\bar{t_2}) \\ P(\bar{q}|t_1,t_2) & P(\bar{q}|t_1,\bar{t_2}) & P(\bar{q}|\bar{t_1},t_2) & P(\bar{q}|\bar{t_1},\bar{t_2}) \end{pmatrix}$$

In Turtle and Croft (1991) and related publications a special setting of the link matrix is proposed. Let w_i be query term weights, and let w_s be the sum of query term weights ($w_s = \sum_i w_i$). The link matrix is defined as follows:

$$L_{\text{PIN}} := \begin{pmatrix} \frac{w_1+w_2}{w_s} & \frac{w_1}{w_s} & \frac{w_2}{w_s} & 0 \\ 0 & \frac{w_2}{w_s} & \frac{w_1}{w_s} & \frac{w_1+w_2}{w_s} \end{pmatrix}$$

This definition leads to a closed form for $P(q|d)$:

$$P(q|d) = \frac{1}{w_s} \cdot \sum_i w_i \cdot P(t_i|d) = \frac{1}{w_s} \cdot \sum_i P(q|t_i) \cdot P(t_i|d)$$

The same closed form can be obtained by representing the network in a matrix and computing the eigenvector of the matrix. Consider the matrix PIN representing the network:

$$\text{PIN} = \left(\begin{array}{c|cccc} & d & t_1 & t_2 & q \\ \hline d & 0 & 0 & 0 & 0 \\ t_1 & P(t_1|d) & 0 & 0 & 0 \\ t_2 & P(t_2|d) & 0 & 0 & 0 \\ q & 0 & P(q|t_1) & P(q|t_2) & 0 \end{array}\right)$$

With $\vec{x} = (P(d), P(t_1|d), P(t_2|d), P(q|d))^T$, we need to solve the equation system

$$0 = \vec{b} + (\text{PIN} - I) \cdot \vec{x}$$

for computing $P(q|d)$, which is the fourth component of vector $\vec{x}$. For demonstrating the solution of this equation system, we rewrite it first:

$$\left(\begin{array}{c|cccc} b_1 & -1 & 0 & 0 & 0 \\ b_2 & P(t_1|d) & -1 & 0 & 0 \\ b_3 & P(t_2|d) & 0 & -1 & 0 \\ b_4 & 0 & P(q|t_1) & P(q|t_2) & -1 \end{array}\right)$$

From the rewritten form, $x_1 = b_1$ follows directly, then x_2 and x_3 follow. We use x_2 and x_3 and obtain x_4.

$$\begin{aligned} x_1 &= b_1 \\ x_2 &= b_1 \cdot P(t_1|d) + b_2 \\ x_3 &= b_1 \cdot P(t_2|d) + b_3 \\ x_4 &= (b_1 \cdot P(t_1|d) + b_2) \cdot P(q|t_1) + (b_1 \cdot P(t_2|d) + b_3) \cdot P(q|t_2) + b_4 \end{aligned}$$

By setting the starting vector $\vec{b} := (0, P(t_1|d)/w_s, P(t_2|d)/w_s, 0)$, we obtain the ranking formula of the PIN approach as the solution for x_4:

$$x_4 = P(q|d) = P(t_1|d)/w_s \cdot P(q|t_1) + P(t_2|d)/w_s \cdot P(q|t_2) = \frac{1}{w_s} \cdot \sum_i P(q|t_i) \cdot P(t_i|d)$$

This result connects the PIN approach with Eigenvector computation. The impact of this result is a topic of further research.

Finally, we express the PIN approach in a form based on vectors $\vec{d} = (P(t_1|d), \ldots, P(t_n|d))^T$ and $\vec{q} = (P(q|t_1), \ldots, P(q|t_n))^T$. We obtain:

$$\mathrm{RSV}_{\mathrm{PIN}}(d, q) := \frac{1}{\|\vec{q}\|_1} \cdot \vec{d}^T \cdot \vec{q} \tag{9}$$

Here, $\|\vec{q}\|_1 = w_s$ is the sum of query term weights.

Next, we recall models based on the probability of relevance and show how to express the models in the general matrix framework.

5.4. Probability of relevance models

The probability $P(r|d,q)$ of relevance r given a document–query pair d, q is the optimal measure for ranking retrieved documents (Robertson, 1977; Robertson & Sparck Jones, 1976). Using the theorem of Bayes, we obtain:

$$P(r|d,q) = \frac{P(d,q,r)}{P(d,q)}$$

Depending on whether we let the document to be conditioned by the query, or the query conditioned by the document, we can write the numerator as follows:

$$\begin{aligned} P(d,q,r) &= P(d|q,r) \cdot P(r|q) \cdot P(q) \\ &= P(q|d,r) \cdot P(r|d) \cdot P(d) \end{aligned}$$

With an odds formulation on the relevance event, i.e. using $\mathrm{RSV}(d,q) = P(r|d,q)/P(\bar{r}|d,q)$ as the retrieval status value (RSV), probabilities $P(d,q)$, $P(d)$ and $P(q)$ drop out. We obtain:

$$\mathrm{RSV}(d,q) = \frac{P(d|q,r)}{P(d|q,\bar{r})} \cdot \frac{P(r|q)}{P(\bar{r}|q)} \tag{10}$$

$$= \frac{P(q|d,r)}{P(q|d,\bar{r})} \cdot \frac{P(r|d)}{P(\bar{r}|d)} \tag{11}$$

The approach with d depending on q (Eq. (10)) is the foundation of the binary independent retrieval (BIR) model, and the approach with q depending on d (Eq. (11)) is the foundation of the language modelling approaches (Lafferty & Zhai, 2002).

5.4.1. Binary independent retrieval (BIR) model

For the BIR model, the probabilities $P(r|q)$ and $P(\bar{r}|q)$ do not affect the ranking of documents with respect to one query. Therefore, we do not consider this factor further.

One of the main assumptions in the BIR model is to represent a document d by a vector $\vec{x}$ of independent features x_i.

$$P(d|q,r) = P(\vec{x}|q,r) = \prod_i P(x_i|q,r)$$

Assuming the features to be terms, and assuming that non-query terms are distributed in relevant as they are distributed in non-relevant documents (i.e. $P(x_i|q,r) = P(x_i|q,\bar{r})$), we obtain:

$$\frac{P(\vec{x}|q,r)}{P(\vec{x}|q,\bar{r})} = \prod_{t_i \in q} \frac{P(x_i|q,r)}{P(x_i|q,\bar{r})}$$

Assuming further that features are binary, i.e. a vector component x_i is 1 if term t_i occurs in the document, otherwise the component is 0, we can split the product into $x_i = 1$ and $x_i = 0$, and obtain:

$$\frac{P(\vec{x}|q,r)}{P(\vec{x}|q,\bar{r})} = \prod_{t_i \in d \cap q} \frac{P(x_i = 1|q,r)}{P(x_i = 1|q,\bar{r})} \cdot \prod_{t_i \in q \setminus d} \frac{P(x_i = 0|q,r)}{P(x_i = 0|q,\bar{r})}$$

Multiplying the equation with 1.0 as expressed in the following equation

$$1.0 = \prod_{t_i \in d \cap q} \frac{P(x_i = 0|q,\bar{r})}{P(x_i = 0|q,r)} \cdot \frac{P(x_i = 0|q,r)}{P(x_i = 0|q,\bar{r})}$$

yields

$$\frac{P(\vec{x}|q,r)}{P(\vec{x}|q,\bar{r})} = \prod_{t_i \in d \cap q} \frac{P(x_i = 1|q,r)}{P(x_i = 1|q,\bar{r})} \cdot \frac{P(x_i = 0|q,\bar{r})}{P(x_i = 0|q,r)} \cdot \prod_{t_i \in q} \frac{P(x_i = 0|q,r)}{P(x_i = 0|q,\bar{r})}$$

The product for $t_i \in q$ does not influence the ranking for one query, and therefore can be dropped. This leads to the following parameters of the BIR model:

$P(t_i|r) := P(x_i = 1|q,r)$ probability that t_i occurs in relevant documents
$P(t_i|\bar{r}) := P(x_i = 1|q,\bar{r})$ probability that t_i occurs in non-relevant documents
$P(\bar{t}_i|r) := P(x_i = 0|q,r)$ probability that t_i does not occur in relevant documents
$P(\bar{t}_i|\bar{r}) := P(x_i = 0|q,\bar{r})$ probability that t_i does not occur in non-relevant documents

Using the abbreviation

$$c_i := \log \frac{P(t_i|r) \cdot P(\bar{t}_i|\bar{r})}{P(t_i|\bar{r}) \cdot P(\bar{t}_i|r)}$$

in which the logarithm is a monotonous transformation, we obtain the RSV for the BIR model:

$$\mathrm{RSV}_{\mathrm{BIR}}(d,q) := \sum_{t_i \in d \cap q} c_i$$

Now, we use our matrix framework for describing the BIR model. Consider the following document–term (content) matrix DT_c, in which term t_1 occurs in documents d_1, d_2, d_4, d_5, t_2 occurs in d_1, d_3, d_4, d_6, and t_3 occurs in d_4.

$$DT_c = \left(\begin{array}{c|cccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline t_1 & 1 & 1 & 0 & 1 & 1 & 0 \\ t_2 & 1 & 0 & 1 & 1 & 0 & 1 \\ t_3 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right)^T$$

For the document-assessor matrix DA_q of query q, let the five documents d_1, d_2, d_3, d_5, d_6 be retrieved, and let d_4 be not retrieved. Let d_1 and d_6 be relevant, and let d_2, d_3, and d_5 be not relevant. Since d_4 is not

retrieved, we do not have relevance information on d_4, but we work here with a closed world assumption and assume that all not retrieved documents are not relevant. The information about retrieved and relevant is represented in the document-assessor matrix DA_q:

$$DA_q = \left(\begin{array}{c|cc} & \textit{retrieved} \text{ (by system)} & \textit{relevant} \text{ (judged by assessor)} \\ \hline d_1 & 1 & 1 \\ d_2 & 1 & 0 \\ d_3 & 1 & 0 \\ d_4 & 0 & 0 \\ d_5 & 1 & 0 \\ d_6 & 1 & 1 \end{array}\right)$$

We create a $N_{D_c} \times N_{D_c}$ square matrix with the system assessment (column system retrieved in matrix DA_q) on the main diagonal ($N_{D_c} = 6$ is the number of documents).

$$diag(DA_q(:, retrieved)) = \left(\begin{array}{c|cccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\ \hline d_1 & 1 & & & & & \\ d_2 & & 1 & & & & \\ d_3 & & & 1 & & & \\ d_4 & & & & 0 & & \\ d_5 & & & & & 1 & \\ d_6 & & & & & & 1 \end{array}\right)$$

The equation

$$\text{retrieved} \cap \text{relevant} = \text{diag}(DA_q(:, \text{retrieved})) \cdot DA_q(:, \text{relevant})$$

yields a vector retrieved $\cap$ relevant in which the document components reflect retrieved and relevant documents. Similar, we obtain the retrieved but not relevant documents:

$$\text{retrieved} \cap \overline{\text{relevant}} = \text{diag}(\text{DA}_\text{q}(:, \text{retrieved})) \cdot (\mathbf{1} - \text{DA}_\text{q}(:, \text{relevant}))$$

Here, **1** is a matrix with 1 in each component. Now we can compose the matrix DA'_q which contains normalised vectors of retrieved and relevant, and retrieved but not relevant.

$$DA'_q = [1/\|\text{retrieved} \cap \text{relevant}\|_1 \cdot \text{retrieved} \cap \text{relevant},\\ 1/\|\text{retrieved} \cap \overline{\text{relevant}}\|_1 \cdot \text{retrieved} \cap \overline{\text{relevant}}]$$

$$DA'_q = \left(\begin{array}{c|cc} & \textit{retrieved} \cap \textit{relevant} & \textit{retrieved} \cap \overline{\textit{relevant}} \\ \hline d_1 & 1/2 & 0 \\ d_2 & 0 & 1/3 \\ d_3 & 0 & 1/3 \\ d_4 & 0 & 0 \\ d_5 & 0 & 1/3 \\ d_6 & 1/2 & 0 \end{array}\right)$$

The equation

$$TA_q = DT_c^T \cdot DA'_q$$

yields in TA_q for each term the probability that the term occurs in relevant or non-relevant of the retrieved documents.

$$TA_q = \begin{pmatrix} & retrieved \cap relevant & retrieved \cap \overline{relevant} \\ t_1 & 1/2 & 2/3 \\ t_2 & 2/2 & 1/3 \\ t_3 & 0/2 & 0/3 \end{pmatrix}$$

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. Term t_3 occurs only in documents that are not retrieved. The term-assessor matrix TA_q has the following probabilistic semantics:

$$TA_q = \begin{pmatrix} & retrieved \cap relevant & retrieved \cap \overline{relevant} \\ t_1 & P(t_1|q,r) & P(t_1|q,\bar{r}) \\ t_2 & P(t_2|q,r) & P(t_2|q,\bar{r}) \\ t_3 & P(t_3|q,r) & P(t_3|q,\bar{r}) \end{pmatrix}$$

Then, the equation

$$NTA_q = \mathbf{1} - TA_q$$

yields the probabilities $P(\bar{t}|r)$ and $P(\bar{t}|\bar{r})$.

Next, we rewrite the term weight c_i as a sum of logarithms on the involved probabilities:

$$\begin{aligned} c_i &= \log \frac{P(t_i|r) \cdot P(\bar{t}_i|\bar{r})}{P(t_i|\bar{r}) \cdot P(\bar{t}_i|r)} \\ &= \log P(t_i|r) + \log P(\bar{t}_i|\bar{r}) - \log P(t_i|\bar{r}) - \log P(\bar{t}_i|r) \end{aligned}$$

Now, we can use the TA_q and NTA_q matrices for computing c_t.

$$\begin{aligned} C_T = [c_i] = {} & \log(TA_q(:, \text{retrieved} \cap \text{relevant})) + \log(NTA_q(:, \overline{\text{retrieved}} \cap \overline{\text{relevant}})) \\ & - \log(TA_q(:, \text{retrieved} \cap \overline{\text{relevant}})) - \log(NTA_q(:, \overline{\text{retrieved}} \cap \text{relevant})) \end{aligned}$$

Finally, we need to multiply the term weight vector C_T with a vector that represents the intersection of terms, so to obtain the sum of c_t weights. The product $\vec{d}^T \cdot \text{diag}(\vec{q})$ yields the vector representing the intersection of document and query terms. For the RSV of the BIR model, we obtain:

$$\text{RSV}_{\text{BIR}}(d,q) = \vec{d}^T \cdot \text{diag}(\vec{q}) \cdot C_T \tag{12}$$

We have expressed the BIR model in our general matrix framework. Next, we address the language modelling approach.

5.4.2. Language modelling (LM)

In LM, the task is to estimate the following parameters:

$P(q|d,r)$: probability of $q$given $d$and relevant documents

$P(q|d,\bar{r})$: probability of $q$given $d$and non-relevant documents

$P(r|d)$: probability of relevance given d

$P(\bar{r}|d)$: probability of non-relevance given d

Similar to the BIR model, there are several assumptions involved in the LM approach for achieving an appropriate ranking formula, but the assumptions differ from the BIR model. The first assumption is that we consider a query as a conjunction of term events, and those term events are independent:

$$P(q|d,r) = \prod_{t \in q} P(t|d,r)$$

The next assumption concerns the probability $P(q|d,\bar{r})$. The LM approach assumes that the probability of a query shall not depend on d given non-relevant documents:

$$P(q|d,\bar{r}) = P(q|\bar{r})$$

This assumption is welcome since $P(q|\bar{r})$ is independent from d and thus $P(q|\bar{r})$ is a constant factor that does not influence the ranking. If the assumption has a reasonable rationale shall be not the topic of this paper (see Croft & Lafferty, 2003 for discussions). Further assumptions could be made for $P(r|d)$ and $P(\bar{r}|d)$. We could argue that relevance (respectively non-relevance) does not depend on a particular d, or that the probability of relevance given d is equal to the probability of non-relevance given d. For both assumptions, the factor $P(r|d)/P(\bar{r}|d)$ would not affect the ranking.

However, rather than finding assumptions to drop $P(r|d)/P(\bar{r}|d)$, it makes sense to use the factor as a query-independent measure of document relevance: for example, number of incoming links, distribution of important terms, or other parameters.

After these considerations, we obtain the following RSV for language modelling, in which we use as before for the BIR model a formulation based on the logarithm of probabilities:

$$\mathrm{RSV}_{\mathrm{LM}}(d,q) := \sum_{t_i \in q} (\log P(t_i|d,r)) + \log \frac{P(r|d)}{P(\bar{r}|d)}$$

Before we address the main problem in LM, namely the estimation of $P(t_i|d,r)$, we formulate the RSV of LM in our matrix framework. We use a vector of term probabilities $P(t_i|d,r)$, a query vector $\vec{q}$ with $q_i = 1$ for query terms and $q_i = 0$ for non-query terms, and the constant $\vec{b} = \log \frac{P(r|d)}{P(\bar{r}|d)}$. Then, we obtain:

$$\mathrm{RSV}_{\mathrm{LM}}(d,q) := (\log P(t_1|d,r) \ldots \log P(t_n|d,r)) \cdot \vec{q} + \vec{b} \tag{13}$$

Considering a matrix DT_c with components $dt_{ij} = P(t_j|d_i,r)$, we obtain $DT_c \cdot \vec{q} + \vec{b}$ and find again, as before for the PIN, a strong parallel with the basic system analysis equation $\vec{y} = A \cdot \vec{x} + \vec{b}$, where $A = DT_c$ and $\vec{x} = \vec{q}$. Exploring this parallel is a topic of future research.

In the context of this paper, we look now at the estimation of $P(t|d,r)$. The LM approach views the probability as a mixture (linear combination) of a term probability $P'(t)$ that is independent of (d,r) and a term probability $P'(t|d,r)$ that depends on (d,r).

$$P(t|d,r) = (1-\lambda) \cdot P'(t) + \lambda \cdot P'(t|d)$$

A common estimate for $P'(t)$ is the frequency of the term in the collection, whereas $P'(t|d)$ is estimated based on the within-document–term frequency.

For defining the estimates, we build on the notation of the previous sections. Let $n_L(t,c)$ be the number of locations in collection c at which term t occurs, and let $N_L(c)$ be the number of locations in collection c. The definition of $n_L(t,d)$ and $N_L(d)$ is analogous. We define the estimates:

$$P'(t) := P'(t|c) := \frac{n_L(t,c)}{N_L(c)}$$

$$P'(t|d) := \frac{n_L(t,d)}{N_L(d)}$$

The estimate for $P'(t)$, the so-called collection term frequency, can be expressed in the general matrix framework based on the matrix location–term matrix LT_c of the collection. LT_c is the concatenation of the LT_d matrices of the documents that are part of the collection.

The estimate for $P'(t|d)$, the so-called within-document term frequency, is based on the location–term matrix LT_d of the document (as shown in Section 3).

Using a vector-based and matrix-based notation, we obtain a vector $P(\vec{t}|d,r)$ of term probabilities:

$$\begin{pmatrix} P(t_1|d,r) \\ \vdots \\ P(t_n|d,r) \end{pmatrix} = (1-\lambda) \cdot \begin{pmatrix} P'(t_1) \\ \vdots \\ P'(t_n) \end{pmatrix} + \lambda \cdot \begin{pmatrix} P'(t_1|d) \\ \vdots \\ P'(t_n|d) \end{pmatrix}$$

Thus, we also have expressed in the matrix framework the mixture of term probabilities. In the following section, we summarise the definitions of the retrieval status values of the considered retrieval models.

5.5. Summary

Consider the definitions of the RSV of the models in one overview:

Vector-space model: $\mathrm{RSV}_{\mathrm{VSM}}(d,q) := \vec{d}^T \cdot \vec{q}$
Generalised VSM: $\mathrm{RSV}_{\mathrm{GVSM}}(d,q) := \vec{d}^T \cdot G \cdot \vec{q}$
Logical approach: $\mathrm{RSV}_{\mathrm{logic}}(d,q) := \frac{1}{P(d)} \cdot \vec{d}^T \cdot \mathrm{diag}(P(t_1),\ldots,P(t_n)) \cdot \vec{q}$
PIN: $\mathrm{RSV}_{\mathrm{PIN}}(d,q) := \frac{1}{\|\vec{q}\|_1} \cdot \vec{d}^T \cdot \vec{q}$
BIR model: $\mathrm{RSV}_{\mathrm{BIR}}(d,q) := \vec{d}^T \cdot \mathrm{diag}(\vec{q}) \cdot C_T$
Language modelling: $\mathrm{RSV}_{\mathrm{LM}}(d,q) := (\log P(t_1|d,r),\ldots,\log P(t_n|d,r)) \cdot \vec{q} + \log \frac{P(r|d)}{P(\bar{r}|d)}$

The logical approach and the PIN approach show strong parallels in their probabilistic and vector-based definitions. The main difference is that in the logical approach, terms are considered as disjoint events, whereas in the PIN approach, terms are independent events. The special setting of the link matrix in the PIN approach leads to a normalisation (L_1 norm) with respect to the query. Both, the logical approach and the PIN approach can be expressed in a VSM-like definition and in a GVSM-like definition with a diagonal matrix of terms. We show above the GVSM-like definition for the logical approach, and the VSM-like definition for the PIN approach. The GVSM-like definition has the advantage that the probabilities in vectors $\vec{d}$ and $\vec{q}$ have the same semantics, namely $P(d|t)$ and $P(q|t)$, respectively.

The BIR model shows a parallel to the generalised vector-space model. Here, the term vector C_T plays the role of the query in the GVSM, and the diagonal matrix of query term weights connect the vector C_T (the result of the relevance feedback) with the document vector. The language modelling approach shows the closest relationship to the basic equation of system analysis, namely $\vec{y} = A \cdot \vec{x} + \vec{b}$.

The overall result of this section is that the general matrix framework allows to explore the sometimes surprisingly close relationships of the models. This result might lead to new possibilities of how to compare models on a theoretical level and how to estimate parameters for language modelling, since the system analysis approach is one of the main foundations for parameter learning.

6. Eigenvectors

In this section, we examine the meaning of the eigenvectors of the square matrices of the spaces in our framework. Some of these matrices, together with a summary of the description of their elements and their eigenvectors, are listed in Tables 5 and 6. We start with the symmetric matrices derived from the content matrices of the spaces (Section 6.1) and we continue with the parent–child matrices, focusing on the ones related to the collection structure (Section 6.2).

Table 5
Matrices related to the content of spaces and their eigenvectors

Content (Section 2.1)			
Space	Matrix	Matrix elements	Eigenvector meaning
Collection c (Section 2.1.1)	DD_c	Number of common terms Document similarity	A term that reflects document co-containment
	TT_c	Number of common documents Term similarity	A document that reflects term co-occurrence
Document d (Section 2.1.2)	LL_d	Number of common terms Location similarity	A term that reflects location co-containment
	TT_d	Number of common locations Term similarity	A location that reflects term co-occurrence
Query q (Section 2.1.3)	DD_q	Number of common assessors Document similarity	An assessor that reflects document co-attraction
	LL_q	Number of common assessors Location similarity	An assessor that reflects location co-attraction
	AA_q	Number of common documents Assessor similarity	A document that reflects assessor co-selection

Table 6
Matrices related to the collection structure and their eigenvectors

Structure (Section 2.2)			
Space	Matrix	Matrix elements	Eigenvector meaning
Collection c (dimension D)	PC_{D_c}	$PC_{D_c} = [pc_{ij}]$, $pc_{ij} = 1$ if d_i parent of d_j	Pagerank based on outlinks hub-oriented
	$PC^T_{D_c}$	$PC^T_{D_c} = [cp_{ij}]$, $cpij = 1$ if d_i child of d_j	Pagerank based on inlinks authority-oriented
	PP_{D_c}	Number of common child documents—parent similarity	Hub
	CC_{D_c}	Number of common parent documents—child similarity	Authority

6.1. Eigenvectors of the matrices related to the content

Consider the TT_c matrix, where each element tt_{ij} represents the number of documents containing both terms t_i and t_j, and thus reflecting term similarity (term co-occurrence) within the collection. The eigenvectors $\vec{x}$ of TT_c are obtained from:

$$\lambda \cdot \vec{x} = TT_c \cdot \vec{x} \tag{14}$$

where λ is a scalar and $\vec{x}$ is a vector, its elements corresponding to the terms in the collection. Therefore, $\vec{x}$ could be either a document or a query (see discussion on the vector-space models in Section 5).

For Eq. (14) to hold, the vectors $\vec{x}$ of TT_c are the documents that reflect the information in TT_c. This means that if a term occurs in the document, then the similar terms also do occur in the document. The eigenvectors of TT_c are documents that reflect term co-occurrence. Similarly, the eigenvectors of DD_c are terms that reflect document co-containment.

In the document space, the interpretation of the eigenvectors of the LL_d and TT_d matrices works analogously. This means that the eigenvectors of TT_d are locations that reflect term co-occurrence and the eigenvectors of LL_d are terms that reflect location co-containment. In the query space, we consider documents that "attract" assessors, and assessors that "select" or "judge" documents. Using this terminology, the eigenvectors of DD_q and LL_q are assessors that reflect co-attraction, and the eigenvectors of AA_q are documents that reflect co-selection.

6.2. Eigenvectors of the matrices related to the structure

In this section, we examine the eigenvectors of the structure-related matrices of the collection space (Table 6). We consider these matrices within the context of a Web collection, where link analysis ranking algorithms can be used to compute their eigenvectors, in order to derive measures of "quality" of Web pages. We focus the discussion on the most prominent of these algorithms: PageRank (Page et al., 1998) and HITS (Kleinberg, 1999).

PageRank computes a query-independent measure of the quality of each Web page, which is recursively defined and depends on the quality of the pages pointing to it. In this algorithm, the collection c corresponds to whole of the Web and the derived $PC^T_{D_c}$ matrix has all its column-sums normalised to 1 ($\sum_{j=1}^{N_{D_c}} cp_{ij} = 1$). For the eigenvectors $\vec{y}$ of $PC^T_{D_c}$, the equation

$$\lambda \cdot \vec{y} = PC^T_{D_c} \cdot \vec{y}$$

holds and the elements of $\vec{y}$ correspond to the documents in the Web collection. The principal eigenvector of $PC^T_{D_c}$ is a vector of values measuring the "quality" of the Web documents based on their parent documents (inlinks). This measure is called pagerank and the relationship among the pagerank values is such that the child-parent structure of the collection is reflected. [8] Similarly, the principal eigenvector of PC_{D_c} is a vector of values measuring the "quality" of the Web documents based on their child documents (outlinks), where the relationship of these values is such that the parent–child structure of the collection is reflected.

HITS, on the other hand, computes two query-dependent measures of the quality of each Web page: its *authority* and its *hub*. Authorities are pages that contain definitive, high quality information on the query topic and hubs are comprehensive lists of links to quality pages on the query topic. The measure of being a good hub depends on how good neighbouring pages are as authorities and vice versa. In this case, the collection c corresponds to a query-biased subset of the Web, consisting of the top k pages retrieved in response to a query, together with their parent and child documents. Each element pp_{ij} of the PP_{D_c} represents the number of child documents pointed to by both parent documents d_i and d_j, and thus reflecting parent similarity among the documents in this collection. An eigenvector of PP_{D_c} is a vector of the documents in the collection that reflects this parent similarity. Similarly, each element cc_{ij} of the CC_{D_c} represents the number of parent documents pointing to both child documents c_i and c_j, and thus reflecting child similarity among the documents in the collection. An eigenvector of CC_{D_c} is a vector of the documents in the collection that reflects this child similarity. The principal eigenvectors of the PP_{D_c} and CC_{D_c} matrices correspond to the hub and authority values of the documents in this collection.

7. Summary

We have described key concepts of Information Retrieval in a well-defined and general matrix framework. These concepts are expressed with a set of standard linear algebra operations on matrices corresponding to elements of appropriately defined matrix spaces. We considered three spaces: a collection space with document and term dimensions, a document space with location and term dimensions, and a query space with document and assessor dimensions. In addition, we considered parent–child matrices representing the relationships between documents or locations or terms.

[8] This is a simplified view of PageRank which assumes that the Web graph is strongly connected and aperiodic. See Page et al. (1998) for discussion on these assumptions and on how to overcome them.

In the matrix framework, we described content-based retrieval, structure-based retrieval, semantic-based retrieval, evaluation, and classical IR models. The dualities we presented include that the similarity measures as known for the document–term matrix of a collection can be defined for the document-assessor matrix of a query, where the similarity measures lead to the classical measures of precision and recall. The matrix-based approach supports the well-defined modelling of more complex evaluation measures that take into account the collection and document structure. Also, the matrix-based approach proved suitable for expressing the main retrieval models and viewing the models in the matrix framework highlights the parallels and dualities of the models.

Further to the dualities regarding similarity measures, we discussed the interpretation of eigenvectors of matrices derived from the content matrices of the spaces and matrices related to the collection structure. The result of this paper is a matrix framework that is general enough to serve as a logical layer for the design and construction of IR systems.

Matrix operations have a close link to relational algebra and multi-dimensional database systems. The presented framework paves the way for modelling IR on the layer of relational and multi-dimensional database technology. Thus, IR applications can share the expressive languages and knowledge representations of data models as provided by database technology. Then, the construction of IR systems becomes efficient and flexible, and we can build effective and increasingly personalised retrieval systems.

Acknowledgments

We would like to thank the anonymous reviewers for their useful comments and Sandor Dominich for his suggestions when reviewing an earlier version of this paper. Also, we would like to thank Hugo Zaragoza for his explanations on parameter estimation in the language modelling approach.

References

Amati, G., & van Rijsbergen, C. J. (1998). Semantic information retrieval. In *Information retrieval: Uncertainty and logics—Advanced models for the representation and retrieval of information* (pp. 189–219). Kluwer Academic Publishers.

Amati, G., & van Rijsbergen, C. J. (2002). Term frequency normalization via Pareto distributions. In *Proceedings of the 24th BCS-IRSG European colloquium on IR research, Glasgow, Scotland* (pp. 183–192).

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Addison Wesley.

Belew, R. K. (2000). *Finding out about*. Cambridge University Press.

Croft, B., & Lafferty, J. (Eds.). (2003). *Language modeling for information retrieval*. Kluwer Academic publishers.

Golub, G. H., & van Loan, C. E. (1996). *Matrix computations* (3rd ed.). Baltimore, MD, USA: The Johns Hopkins University Press.

Grossman, D. A., & Frieder, O. (1998). *Information retrieval: Algorithms and heuristics*. Kluwer Academic Publishers.

Hawking, D., Voorhees, E., Craswell, N., & Bailey, P. (1999). Overview of the TREC-8 Web track. In E. M. Voorhees & D. Harman (Eds.), *Proceedings of the 8th Text REtrieval Conference (TREC-8)* (pp. 131–150). Gaithersburg, MD: NIST.

Kazai, G., Lalmas, M., & de Vries, A. (2004). The overlap problem in content-oriented XML retrieval evaluation. In *Proceedings of the 27th annual international conference on research and development in information retrieval* (pp. 72–79). Sheffield, UK: ACM.

Kekalainen, J., & Jarvelin, K. (2002). Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology, 53*(13), 1120–1129.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of ACM, 46*(5), 604–632.

Lafferty, J., & Zhai, C. (2002). Probabilistic relevance models based on document and query generation, Chapter 1. In Croft & Lafferty (2003).

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Tech. rep., Stanford Digital Library Technologies Project.

Robertson, S. (1977). The probability ranking principle in IR. *Journal of Documentation, 33*, 294–304.

Robertson, S., & Sparck Jones, K. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*, 129–146.

Robertson, S. E., Walker, S., & Hancock-Beaulieu, M. (1995). Large test collection experiments on an operational interactive system: Okapi at TREC. *Information Processing and Management, 31*, 345–360.

Roelleke, T., Lalmas, M., Kazai, G., Ruthven, I., & Quicker, S. (2002). The accessibility dimension for structured document retrieval. In *Proceedings of the 24th BCS-IRSG European colloquium on IR research*, Glasgow, Scotland (pp. 284–302).

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

Turtle, H., & Croft, W. (1991). Efficient probabilistic inference for text retrieval. In *Proceedings of RIAO 91* (pp. 644–661).

van Rijsbergen, C. J. (1979). *Information retrieval* (2nd ed.). London: Butterworths.

van Rijsbergen, C. J. (1986). A non-classical logic for information retrieval. *The Computer Journal, 29*(6), 481–485.

Wong, S., & Yao, Y. (1995). On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems, 13*(1), 38–68.

Wong, S., Ziarko, W., & Wong, P. (1985). Generalized vector space model in information retrieval. In *Proceedings of the 8th international conference on research and development in information retrieval* (pp. 18–25). New York: ACM.