DOKTORI (Ph.D.) ÉRTEKEZÉS

GENERAL FORMAL FRAMEWORK FOR INFORMATION RETRIEVAL AND ITS APPLICATIONS

INFORMÁCIÓ-VISSZAKERESŐ MÓDSZEREK EGYSÉGES KERETRENDSZERE ÉS ALKALMAZÁSAI

Kiezer Tamás

Témavezető: Dr. Dominich Sándor†
(1954 - 2008)

Pannon Egyetem
Műszaki Informatikai Kar
Informatikai Tudományok Doktori Iskola

2010

GENERAL FORMAL FRAMEWORK FOR INFORMATION RETRIEVAL AND ITS APPLICATIONS

INFORMÁCIÓ-VISSZAKERESŐ MÓDSZEREK EGYSÉGES KERETRENDSZERE ÉS ALKALMAZÁSAI

Értekezés doktori (PhD) fokozat elnyerése érdekében
a Pannon Egyetem Informatikai Tudományok
Doktori Iskolájához tartozóan.

Írta: Kiezer Tamás

Témavezető: Dr. Dominich Sándor†

A Doktori Iskola megbízásából elfogadásra javaslom (igen / nem)

(aláírás)

A jelölt a doktori szigorlaton .....%-ot ért el.

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: …..…..... igen /nem

…….
(aláírás)

Bíráló neve: …..…..... igen /nem

…….
(aláírás)

A jelölt az értekezés nyilvános vitáján ….....%-ot ért el.

Veszprém, …….
a Bíráló Bizottság elnöke

A doktori (PhD) oklevél minősítése….....

……
Az EDHT elnöke

TARTALMI KIVONAT

Az Internet és a World Wide Web megjelenése mind gyakorlati, mind elméleti szempontból jelentős mértékben növelte az információ-visszakeresés fontosságát. Sokféle visszakereső módszer került kidolgozásra az elmúlt fél évszázad során, melyeket ma is folyamatosan fejlesztenek tovább.

A klasszikus módszerek egyike a vektortér módszer (Vector Space Model – VSM). Már két évtizede tudjuk, hogy a VSM nem vezethető le következetesen azon matematikai fogalmakból, melyeken alapszik, de ezidáig nem született megfelelő megoldás a problémára. Disszertációmban egy egységes, következetes, formális információ-visszakereső keretrendszert adok meg és bemutatom, hogy ennek alkalmazásával az általánosított vektortér módszer (Generalised Vector Space Model – GVSM), az LSI módszer (Latent Semantic Indexing model) és a VSM helyes matematikai formalizmust kap, amely konzisztens a gyakorlattal.

Az egységes keretrendszerben új, konzisztens visszakereső módszereket adok meg: az entrópia- és valószínűség-alapú módszert, valamint a kifejezetten Webes információ-visszakeresésre használható kombinált fontosság-alapú módszert. Utóbbit a WebCIR Webes keresőmotorban implementáltuk, mely szintén bemutatásra kerül a dolgozatban.

A megadott módszerek relevancia-hatékonyságát kísérleti úton vizsgáltam meg. Az entrópia- és valószínűség-alapú módszerek in vitro kiértékelése során 5 és 19 százalék közti javulás volt mérhető a VSM és LSI módszerekkel szemben. A WebCIR keresőmotor in vivo tesztelése során kapott eredmények alapján – a Yahoo!, Altavista, és MSN kereskedelmi keresőmotorok eredményeivel összehasonlítva – mondhatjuk, hogy a WebCIR visszakereső és rangsoroló technológiája versenyképes alternatívát jelent.

ABSTRACT

With the advent of the Internet and World Wide Web (Web), Information Retrieval (IR) gained tremendous practical impact and theoretical importance. A number of retrieval methods have been elaborated since the inception, about half a century ago, which have been continuously evolving nowadays as well.

One of the classical methods is the Vector Space Model (VSM). It has been known for two decades that the VSM does not follow logically from the mathematical concepts on which it has been claimed to rest, but no proper solution has emerged so far. In this thesis, a general, discrepancy-free formal framework for IR is given and it is shown that using the concepts of this framework the Generalised Vector Space retrieval Model (GVSM), the Latent Semantic Indexing retrieval model (LSI) and the classical vector space retrieval model gain a correct formal mathematical formulation and background that is consistent with practice.

Based on this general framework the Entropy- and Probability-based retrieval methods are formulated consistently. Suited especially for the World Wide Web, the Combined Importance-based method is also derived from this framework. A search engine called WebCIR is introduced, which implements this method.

Experimental evaluation results of the given methods are also reported. In vitro measurement of the Entropy- and Probability-based methods showed that, using these methods, improvement levels between 5 and 19 percent can be reached in comparison with the VSM and LSI methods. In vivo evaluation of the WebCIR search engine was also carried out. The results, which were compared to commercial search engines including Yahoo!, Altavista, and MSN, suggest that WebCIR is a very competitive retrieval and ranking technology.

AUSZUG

Durch die Erscheinung vom Internet und World Wide Web wurde die Bedeutung vom Information Retrieval (IR) sowohl aus praktischer als auch aus theoretischer Hinsicht deutlich erhöht. Während des vorigen halben Jahrhunderts wurden vielerlei Retrievalmethoden konzipiert, die auch heute kontinuierlich weiterentwickelt werden.

Eine von den klassischen Methoden ist die Vektorraum Methode (Vector Space Model – VSM). Laut unseres Wissens konnte diese zwei Jahrzehnte lang nicht von den mathematischen Begriffen konsequent abgeleitet werden, worauf diese Methode aufbaut. Bislang wurde keine entsprechende Lösung gefunden. In meiner Abhandlung gebe ich ein einheitliches, konsequentes, formales Framework für Information Retrieval an und lege eine mit der Praxis konsistente Anwendung vor, wobei die verallgemeinerte Vektorraum Methode (Generalised Vector Space Model – GVSM), die LSI Methode (Latent Semantic Indexing Modell) und die VSM die richtige mathematische Abbildung bekommen.

In dem einheitlichen Framework gebe ich neue, konsistente Retrieval Methoden an. Sowohl die Entropie- und Wahrscheinlichkeitsbasierte Methode als auch die auf kombinierte Wichtigkeit basierende Methode, die besonders zum Web Information Retrieval geeignet ist, werden erläutert. Letztere wurde bei der WebCIR Suchmaschine eingesetzt, die in der Abhandlung auch vorgestellt wird.

Die Relevanzwirksamkeit der angegebenen Methoden untersuchte ich durch Experimente. Bei der in vitro Bewertung der Entropie- und Wahrscheinlichkeitsbasierten Methoden konnte 5 bis 19 Prozent Verbesserung gemessen werden gegenüber der VSM und LSI Methoden. Laut meiner Ergebnisse von WebCIR Suchmaschine bei der in vivo Testverfahren und im Vergleich zur Yahoo!, Altavista und MSN Suchmaschinen, kann man behaupten, dass die Retrieval und Ranking Technologie von WebCIR eine wettbewerbsfähige Alternative darstellt.

ACKNOWLEDGEMENTS

First of all, I want to express my sincere gratitude to my supervisor, Sándor Dominich for the continuous guidance, support and for the creative ideas during my research. The results reported in this thesis have been achieved under Sandor's guidance, however, after his sudden and unexpected death in 2008, I had to complete this thesis considering the recommendations I had received from him earlier. I am thankful for everything that I learned from Sándor both professionally and as a person.

Thanks goes to Professor Ferenc Friedler for providing the highly supportive environment at the Department of Computer Science and Systems Technology, University of Pannonia.

Many thanks to my colleagues, Júlia Góth, Adrienn Skrop, Zoltán Szlávik and Miklós Erdélyi for the helpful comments in the period of writing the thesis. Furthermore thanks to all the people at Department of Computer Science and Systems Technology for the supportive environment.

The greatest acknowledgement I reserve for my mother who, unfortunately, could not live long enough to see me submitting this thesis, something she would be deeply proud of as a mother. I received the greatest support and encouragement from her throughout my life and studies. This thesis is dedicated to her.

TABLE OF CONTENTS

1 Introduction 1

1.1 The Vector Space Model of Information Retrieval 1

1.2 Motivation: Is the dot product – dot product? 3

1.3 The basis of the space: point of view 5

1.4 Kernel based methods 6

1.5 Information retrieval and measure theory 7

1.6 Organisation of the thesis 7

2 Methods applied and collections used in the experiments 9

2.1 Standard test collections 9

2.1.1 ADI collection 9

2.1.2 MED collection 9

2.1.3 TIME collection 9

2.1.4 CRAN collection 10

2.2 The „vein.hu” collection 10

2.3 Evaluation methods and measures 11

2.3.1 Main concepts 11

2.3.2 Precision-recall method 12

2.3.3 MLS method 14

2.3.4 DCG method 15

2.3.5 RC method 15

2.3.6 RP method 16

3 A measure theoretic approach to information retrieval 17

3.1 General formal framework for information retrieval 17

3.2 Measure theoretic definition of information retrieval [Thesis 1.a] 20

4 Measure theoretic aspect of “classical” retrieval methods 23

4.1 Information retrieval in linear space with general basis 24

4.1.1 Mathematical concepts 24

4.1.2 Generalised Vector Space Model 25

4.1.3 General basis-based retrieval method (GB method) [Thesis 1.b] 25
4.2 Latent Semantic Indexing retrieval method [Thesis 1.b] 27
4.3 Information retrieval in linear space with inner product 28
4.4 Information retrieval in orthonormal real linear space [Thesis 1.b] 30
4.5 Principle of Object Invariance 32

5 Entropy- and probability-based retrieval 35
5.1 Entropy-based information retrieval 35
5.1.1 Entropy-based retrieval method 36
5.2 Probability-based information retrieval 37
5.2.1 Probability-based retrieval method 38
5.3 Experimental results for Entropy- and Probability-based retrieval methods 39

6 Combined importance-based information retrieval 43
6.1 Content importance 43
6.2 Similarity measure 45
6.3 Link importance 45
6.4 Combined Importance-based Web retrieval and ranking method 46
6.4.1 Web Retrieval and Ranking method 47

7 WebCIR – a search engine using the combined importance-based method 49
7.1 Web Search Engine architecture 49
7.2 WebCIR's architecture 51
7.2.1 CrawlDB 53
7.2.2 Crawler Module 53
7.2.3 LinkDB 53
7.2.4 Indexer Module 54
7.2.5 Preprocessing Module 55
7.2.6 Document Info Index 55
7.2.7 Term Info Index 56
7.2.8 Query Module 56
7.3 Querying 56
7.3.1 Query syntax 57
7.3.2 Query expansion 57

7.4 Searching ... 59
7.5 Ranking ... 60
7.6 User interface ... 61
7.7 WebCIR's evaluation ... 63
7.7.1 Evaluation methodology ... 63
7.7.2 Results of the evaluation ... 64
7.7.3 Discussion ... 66
7.7.4 Future work ... 67

8 Conclusions ... 70
8.1 Theses ... 70
8.2 Tézisek magyar nyelven ... 71
8.3 Publications ... 72
8.3.1 Publications directly related to the thesis ... 72
8.3.2 Other publications relevant to the thesis ... 72

Bibliography ... 74

Appendix A. ... 80
A.1. ADI collection ... 80
A.2. MED collection ... 81
A.3. TIME collection ... 82
A.4. CRAN collection ... 84

Appendix B. ... 85

Appendix C. ... 91

Appendix D. ... 93

Appendix E. ... 94

Appendix F. ... 95

Appendix G. ... 96

Appendix H. ... 97

LIST OF FIGURES

Figure 1.1 Document and query weight vectors. 3
Figure 1.2 Document and query weight vectors. 4
Figure 2.1 Visual representation of quantities precision, recall, fallout. 13
Figure 2.2 Typical precision-recall graph (for the test collection ADI) 14
Figure 5.1 Orthonormal (e_{1108}, e_{5637}), and general (g_{1108}, e_{5637}) basis vectors 40
Figure 7.1 General search engine architecture [5]. 50
Figure 7.2 System architecture of WebCIR 52
Figure 7.3 Starting page of WebCIR. 61
Figure 7.4 Sample search results for query: ”tanulmányi tájékoztató”. 62
Figure 7.5 Screenshot of the measurement software. 64
Figure 7.6 The distribution of fuzzy probabilities. 67

Figure A.1 Structure of the 50th document in the ADI test collection. 80
Figure A.2 Excerption from the adi.que file in the ADI test collection. 81
Figure A.3 Excerption of the relevance assessments file. 81
Figure A.4 Structure of the 8th document in the MED test collection. 82
Figure A.5 Structure of an article in the TIME test collection. 82
Figure A.6 Excerption of Time.que file.. 83
Figure A.7 Excerption of Time.rel file. 83
Figure A.8 Excerption of the TIME stoplist 83
Figure A.9 Structure of the 36th document in the CRAN test collection. 84

LIST OF TABLES

Table 3.1 Formal mathematical framework for IR 19
Table 3.2 Formal mathematical framework for automatic computerised IR 20
Table 5.1 Statistics of the test collections used in experiments. 39
Table 5.2 Mean average precision obtained on standard test collections. 41
Table 5.3 Mean average precision obtained on standard test collections. 42
Table 7.1 Term weighting scheme. 61
Table 7.2 Example weights for the segmented fuzzy probabilities. 68

1 Introduction

Information Retrieval (*IR*) is concerned with finding and returning information items stored in computers, which are relevant to a user's information need (materialised in a request or query). With the advent of the Internet and World Wide Web (Web for short), *IR* has a tremendous practical impact and theoretical importance. Many retrieval methods have been elaborated since the inception, about half a century ago, which are continuously evolving nowadays as well [62][23][58][76].

One of the classical methods is the so called Vector Space Model (VSM); which was inspired by the following ideas:

If it is assumed, naturally enough, that the most obvious place where appropriate content identifiers might be found is the document itself, then the number of occurrences of terms can give meaningful indication of its content [43]. Given m documents and n terms, each document can be assigned a sequence (of length n) of weights which represent the degrees to which terms pertain to (characterise) that document. If all these sequences are put together, an $n \times m$ matrix, called term-document matrix, of weights is obtained, where the columns correspond to documents, while the rows to terms.

Let us consider a – textual – query expressing an information need to which an answer is to be found by searching the documents. In 1965, Salton proposed that both documents and queries should use the same conceptual space [59], while in 1975 this idea was combined with the term-document matrix [62]. More than a decade later, Salton and Buckley re-used this framework, and gave a mathematical description which has since become known as the Vector Space Model (VSM) or Vector Space Retrieval [61].

In the following sections, the mathematical concepts of VSM are briefly introduced. It is also shown – with the help of an illustrative example – how the VSM approach conflicts with the mathematical notion of a vector space. It was this inconsistency, what inspired the work of this dissertation, i.e. to develop a new, discrepancy-free formal framework for IR which is introduced in Chapter 3.

1.1 The Vector Space Model of Information Retrieval

The formal mathematical framework for the classical Vector Space Model (VSM; [62]) of Information Retrieval (IR) is the orthonormal Euclidean space. (See [27] for a very instructive reading in this respect.) This means the following:

a) Let us consider a Euclidean space E – which is a very special linear (or vector) space – of dimension equal to the number of index terms, say n.

b) Each index term t_i (i=1,…,n) corresponds to a coordinate axis (or dimension) X_i of this space E, and is represented on that axis X_i by a point P_{ij} given by the

weight w_{ij} of that index term (in a document d_j): the point P_{ij} is conceived as being the end-point of a vector $\mathbf{P}_{ij}$ defined by the product between the weight w_{ij} and the unit length basis vector $\mathbf{e}_i$ on that axis, i.e., $\mathbf{P}_{ij}=w_{ij}\mathbf{e}_i$.

c) The index terms t_i (i=1,…, n) are considered to be independent of each other, this means that the corresponding coordinate axes are pair-wise perpendicular to one another.

d) Every document d_j (j=1,…, m) is represented as a point D_j in the space E given by the end-point of the vector $\mathbf{D}_j$ obtained as the vector sum of all the corresponding index term vectors, i.e., $\mathbf{D}_j=\sum_{i=1}^{n}\mathbf{P}_{ij}$.

For retrieval purposes, the query q is considered to be a document, and hence represented in that same space E (as being a vector $\mathbf{Q}=\sum_{i=1}^{n}\mathbf{P}_i$). In order to decide which document to retrieve in response to the query, the inner (also called scalar or dot) product $\mathbf{Q}\cdot\mathbf{D}_j$ between the query-vector $\mathbf{Q}$ and document-vector $\mathbf{D}_j$ is computed as a measure of how much they have in common or share.

Let us consider as an example the orthonormal Euclidean space of dimension two, E_2; its unit length and perpendicular basis vectors are $\mathbf{e}_1$=(1,0) and $\mathbf{e}_2$=(0,1). Let us assume that we have the following two index terms: t_1='computer' and t_2='hardware', which correspond to the two basis vectors (or, equivalently, to coordinate axes) $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively (Figure 1.1).

Consider now a document D being indexed by the term 'computer', and having the following weights vector: $\mathbf{D}$=(3,0). Let a query Q be indexed by the term 'hardware', and have the following weights vector: $\mathbf{Q}$=(0,2). The dot product $\mathbf{D}\cdot\mathbf{Q}$ is the following: $\mathbf{D}\cdot\mathbf{Q}=3\times0+0\times2=0$. (This means that the document D is not retrieved in response to the query Q.)

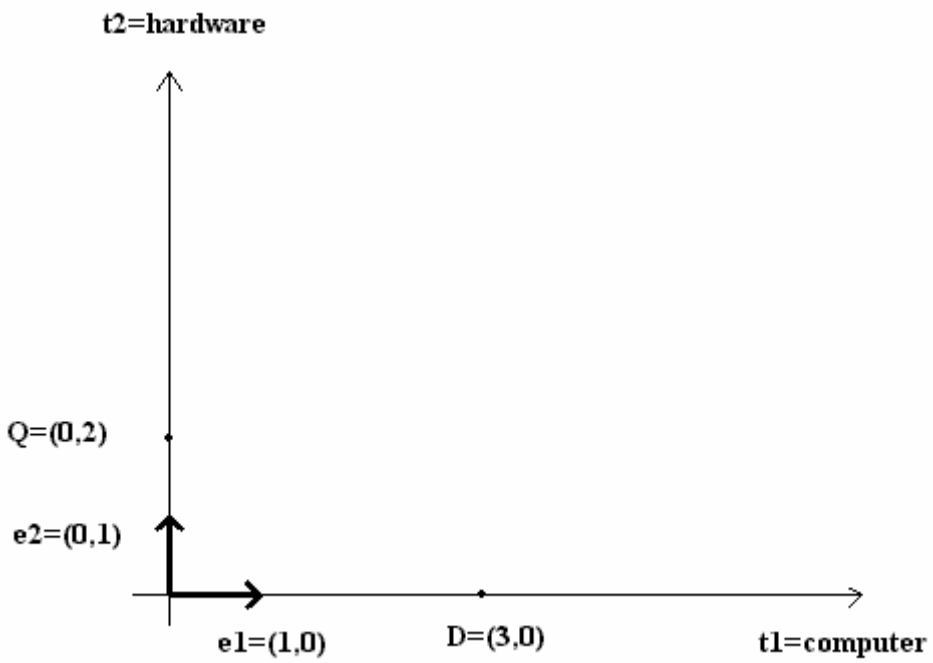


Figure 1.1 Document and query weight vectors. The document vector D(3;0) and query vector Q(0;2) are represented in the orthonormal basis (e_1;e_2). These basis vectors are perpendicular to each other, and have unit lengths. The dot product **D·Q** is the following: **D·Q**=3×0+0×2=0 (which means that the document *D* is not retrieved in response to the query *Q*).

1.2 Motivation: Is the dot product – dot product?

In [84] it is argued that: "the notion of vector in the VSM merely refers to data structure… the scalar product is simply an operation defined on the data structure…The main point here is that the concept of a vector was not intended to be a logical or formal tool", and shown why the VSM approach conflicts with the mathematical notion of a vector space.

In order to render and to illustrate the rightfulness of the concerns with the mathematical modelling as well as of the mathematical subtleties involved, let us enlarge our example of Figure 1.1. From the user's point of view, because the hardware is part of a computer, he/she might be interested to see whether a document *D* contains information also on hardware. In other words, he/she would not mind if the document *D* would be returned in response to the query *Q*. It is well-known that the term independence assumption is not realistic. The terms may depend on each other, and they often do in practice, as in our example, too. It is also known that the independence assumption can be counterbalanced, to a certain degree, in practice by, e.g., using thesauri. But can term dependence be captured and expressed in a vector space? One possible answer is as follows: instead of considering an orthonormal basis, let us consider a general basis (Figure 1.2).

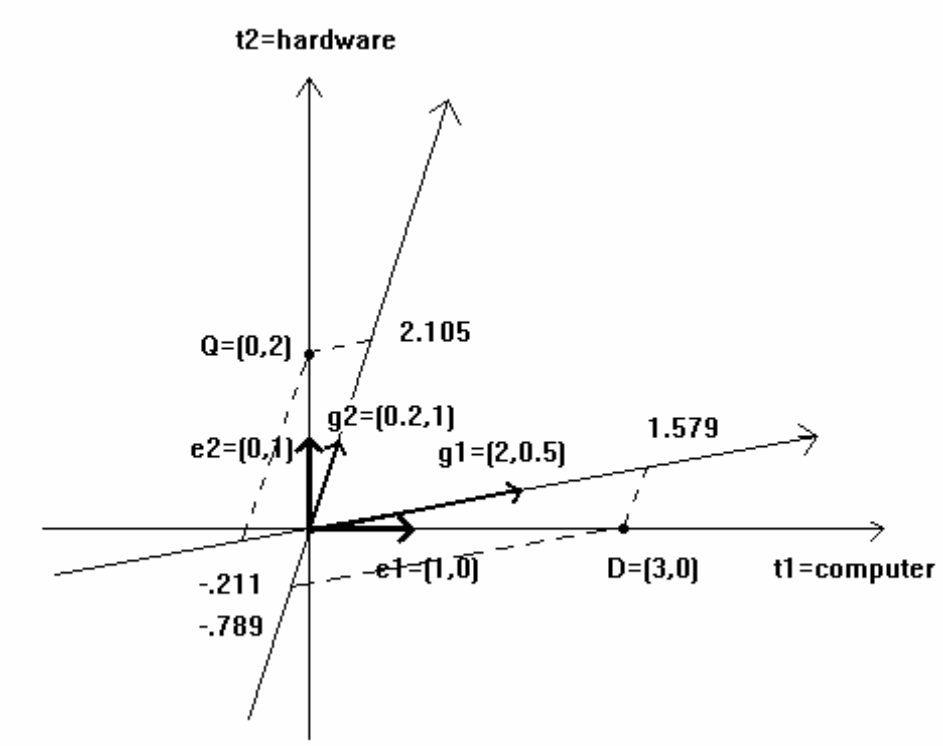


Figure 1.2 Document and query weight vectors. The document vector D(3;0) and query vector Q(0;2) are represented in the orthonormal basis (e_1;e_2). They are also represented in the general basis (g_1;g_2), these basis vectors are not perpendicular to each other, and do not have unit lengths. The coordinates of the document vector in the general basis will be D(1.579;-0.789), whereas those of the query vector will be Q(-0.211;2.105). The value of the expression D·Q viewed as an inner product between document D and query Q is always zero, regardless of the basis. But the value of the expression D·Q literally viewed as algebraic expression is not zero.

The basis vectors of a general basis need not be perpendicular to each other, and need not have unit lengths. In our example (Figure 1.2): the term 'hardware' is narrower in meaning than the term 'computer'. If orthogonal basis vectors are used to express the fact that two terms are independent, then a 'narrower' relationship can be expressed by taking an angle smaller than 90° (the exact value of this angle can be the subject of experimentation, but it is not important for the purpose of this example). So, let us consider the following two oblique basis vectors: let the basis vector $\mathbf{g}_1$ corresponding to term t_1 be $\mathbf{g}_1$=(2;0.5), and the basis vector $\mathbf{g}_2$ representing term t_2 be $\mathbf{g}_2$=(0.2;1). The coordinates $\mathbf{D}^i$ of the document vector **D** in the new (i.e., the general) basis are computed as follows (see e.g. [66] for a background and justification of the formulas subsequently used):

$$\mathbf{D}^i = (\mathbf{g}_i)^{-1} \times \mathbf{D} = (\mathbf{g}_1\ \mathbf{g}_2)^{-1} \times \mathbf{D} = \begin{pmatrix} 2 & 0.2 \\ 0.5 & 1 \end{pmatrix}^{-1} \times (3;\ 0)^{\mathrm{T}} =$$

$$= \begin{pmatrix} 0.526 & -0.105 \\ -0.263 & 1.053 \end{pmatrix} \times (3;\ 0)^{\mathrm{T}} = (1.579;\ -0.789),$$

whereas the coordinates $\mathbf{Q}^i$ of the query vector **Q** are as follows:

$$\mathbf{Q}^i = (\mathbf{g}_i)^{-1} \times \mathbf{Q} = (\mathbf{g}_1\ \mathbf{g}_2)^{-1} \times \mathbf{Q} = \begin{pmatrix} 2 & 0.2 \\ 0.5 & 1 \end{pmatrix}^{-1} \times (0;\ 2)^{\mathrm{T}} =$$

$$= (-0.211;\ 2.105).$$

Now, if the similarity function is interpreted – as is usual in IR – as being the expression of the dot product between the document vector and query vector, then the dot product **DQ** of the document vector **D** and query vector **Q** is to be computed relative to the new, general basis $\mathbf{g}_i=(\mathbf{g}_1\ \mathbf{g}_2)$; this computation proceeds as follows:

$$\mathbf{DQ} = \mathbf{D}^i \times \mathbf{g}_{ij} \times (\mathbf{Q}^j)^{\mathsf{T}} = (1.579;\ -0.789) \times \begin{pmatrix} g_1 g_1 & g_1 g_2 \\ g_2 g_1 & g_2 g_2 \end{pmatrix} \times$$

$$\times (-0.211;\ 2.105)^{\mathsf{T}} = (1.579;\ -0.789) \times \begin{pmatrix} 4.25 & 0.9 \\ 0.9 & 1.04 \end{pmatrix} \times$$

$$\times (-0.211;\ 2.105)^{\mathsf{T}} = 0.$$

It can be seen that the dot product of the document vector **D** and query vector **Q** is equal to zero in the new basis too; this means that the document is not retrieved in the general basis either. This should not be extraordinary because, as it is well-known, the scalar product is invariant with respect to the change of basis. This means that, under the inner product interpretation of similarity (i.e., if the similarity function is interpreted as being the dot product between two vectors), the no-hit case remains valid using also general basis!

1.3 The basis of the space: point of view

The change of basis represents a “point of view” from which the properties of documents and queries are judged. If the document is conceived as being a vector, i.e., it is the same in any basis (equivalently, its meaning, information content, or properties remain the same from any viewpoint) the inner product is also invariant, and hence so is the similarity function.

But then, what is the point in taking a general basis?

If we assume that the – meaning or information content of a – document and query do depend on the “point of view”, i.e. on the change of basis, then the properties of documents and queries may be found to be different in different bases. This is equivalent to not interpreting the similarity function as expressing an inner

product, rather being a numerical measure of how much the document and query share. Thus, the similarity, which formally looks like the algebraic expression of an inner product, is literally interpreted as a mere algebraic expression (or computational construct) being a measure of how much the document and query share, and not as expressing an inner product.

In this new interpretation, in our example of Figure 1.2 we obtain the following value for the similarity between document and query:

$$1.579\times(-0.211)+(-0.789)\times(2.105) = -1.994,$$

which is different from zero. (Subjectively, a numerical measure of similarity should be a positive number, although this is irrelevant from a formal mathematical point of view.) So,

(i) using a general basis to express term dependence,

(ii) and not interpreting similarity as being an inner product,

the document *D* is being returned in response to *Q*, as intended.

1.4 Kernel based methods

Kernel-based learning methods [36] have to be referenced here as an attempt to overcome the restrictions induced by the use of the Euclidean space as a mathematical framework in IR. In this approach, data items (documents) are mapped into high-dimensional spaces, where information about their mutual positions (inner products) is used for constructing classification, regression, or clustering rules. They consist of a general purpose learning module (e.g. classification or clustering) and a data-specific part, called the *kernel*, which defines a mapping of the data into the feature space of the learning algorithm.

Kernel-based algorithms utilize the information encoded in the inner-product between all pairs of data items, which is stored in the so called *kernel matrix*. This representation has the advantage of that very high dimensional feature spaces can be used, as the explicit representation of feature vectors (corresponding to data items, e.g. documents) is not needed. This kind of approach is applicable to different fields of science where methods are based on the inner products between vectors.

For example, the kernel corresponding to the feature space defined by VSM is given by the inner product between the feature (document) vectors:

$K(\mathbf{D}_1, \mathbf{D}_2) = \mathbf{D}_1^{\mathsf{T}} \times \mathbf{D}_2$.

The kernel matrix is the document by document matrix.

As showed in the example of Figure 1.1 classical VSM suffers from some drawbacks, in particular the fact that semantic relations between terms are not taken into account. In kernel based approach, this issue can be addressed by finding a mapping that captures some semantic information, with a “semantic kernel”, that computes the similarity between documents by also considering relations between

different terms. One possible approach is the *semantic smoothing for vector space model* [67], where a semantic network is used to explicitly compute the similarity level between terms.

In [20] a technique called *latent semantic kernels* is proposed based on latent semantic indexing (LSI) [23]. In this approach, the documents are implicitly mapped into a "semantic space", where documents that do not share any terms can still be close to each other if their terms are semantically related. In [20] good experimental results are also reported.

1.5 Information retrieval and measure theory

The Euclidean space as a mathematical/formal framework for IR is very illustrative and intuitive. But is there any real and actual connection between the mathematical concepts used (vector, vector space, scalar product) and IR notions (document, query, similarity)? In other words, for example, is a document or query a vector? May it be conceived to be a vector in the actual mathematical sense of the word? It can be seen that in the classical VSM there is a discrepancy between the theoretical (mathematical) model and the effective retrieval algorithm applied in practice. They are not consistent with each other: the algorithm does not follow from the model, and conversely, the model is not a formal framework for the algorithm. The modelling concerns justify the following question: is or should or can the VSM be really and actually based on the concept of inner product? In other words:

Is the inner product an underlying or necessary "ingredient" in IR?

In this dissertation, using the mathematical theory of measure, a proper answer will be given to this question for the first time. It will be shown that the answer is: no, the inner product is not, in general, an underlying ingredient in IR. Whether or not, it depends on how we conceive documents and queries. If they are conceived as entities whose content is susceptible to our interpretation (their meaning depends on our point of view), then the similarity function does not have the meaning of an inner product. If, however, they are viewed as entities bearing one fixed meaning, then the similarity function does have the meaning of an inner product. Moreover, also based on mathematical measure theory, novel retrieval methods are proposed, which are consistently derived from the mathematical framework introduced.

1.6 Organisation of the thesis

The remainder of the thesis is structured as follows.

In Chapter 2 the databases used for testing retrieval methods are described first. Four standard test collections, namely ADI, MED, TIME and CRAN are introduced as well as a Web collection named "vein.hu". The latter collection was constructed by our research group CIR (Center for Information Retrieval), and it can be used to evaluate effectiveness of retrieval methods developed especially for the Web. This is

followed by the description of the applied methods, that were used for evaluate the developed retrieval methods' effectiveness given in Chapters 5-6.

The next five Chapters present and discuss the results, which I obtained during my research:

Chapter 3 introduces a general and formal framework for IR as a concept based on widely accepted definitions of IR. Then, the concept of a mathematical measure is introduced in order to propose a mathematical definition of IR. This starts with describing in words the concepts used, and is followed by exact mathematical definitions.

In Chapter 4, the notion of a linear space is described in words first, which is followed by giving the exact mathematical definition. Then, a retrieval method in a linear space with general basis is proposed. Known retrieval methods (latent semantic indexing retrieval, classical vector space retrieval, generalised vector space retrieval) are integrated into the definition suggested in Section 3.2 by introducing a new principle: principle of object invariance (POI). Thus, these retrieval methods gain a correct formal mathematical background.

In Chapter 5, two novel retrieval methods, namely the Entropy- and the Probability-based retrieval methods are proposed as derived naturally from the definition introduced in Chapter 3.2. Then, experimental results on their relevance effectiveness are presented. In vitro measurements – test collections and computer programs were used under laboratory conditions (without user assessments) – were performed using the collections and methods introduced in Chapter 2 to evaluate Entropy- and Probability-based retrieval methods.

Chapter 6 introduces a new combined retrieval method which is partly derived from the definition introduced in Chapter 3.2 and is especially developed for the World Wide Web. This starts with the description of how the content and link importance of documents (Web pages) as well as similarity are calculated, and it is followed by the presentation of the steps of the combined method.

In Chapter 7, a search engine called WebCIR is introduced, which implements the Combined Importance-based Web retrieval and ranking method. This starts with an overview of the system architecture and is followed by a more detailed description of the main modules and functions. Querying, searching and ranking are explained and the user interface is also introduced. This is followed by the in vivo evaluation of the WebCIR search engine. Several evaluation methods were chosen, both with and without the need of user assessment. Results were compared to commercial search engines Yahoo!, Altavista, and MSN. After discussing the results, the Chapter ends with conclusions, observations and suggestions for further research.

Chapter 8 gives a summary of my results.

2 Methods applied and collections used in the experiments

2.1 Standard test collections

In IR, the evaluation of a retrieval method is usually based on a reference test database also called test collection and on an evaluation measure. Each test collection is manufactured by specialists, and has a fixed structure as follows:

- The collection of documents *d* are given.
- The set of information requests: queries *q* are given.
- The relevance list is given, i.e., it is exactly known which document is relevant to which query.

A number of test collections exist today [68]. A few of these collections are freely available on the Web, for example at:

- http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/
- http://www.cs.utk.edu/~lsi/corpa.html

The most popular standard test collections are: TREC, ADI, MED, CACM, CISI, TIME, REUTERS, INEX. These collections vary in size, topic and in the number of queries. I used the following four test collections for measuring relevance effectiveness: ADI, MED, TIME and CRAN.

2.1.1 ADI collection

The ADI collection contains 82 homogeneous English articles from computing journals with 35 queries. A detailed description about the collection can be found in Appendix A.1.

2.1.2 MED collection

MED is a collection of 1033 medical abstracts from the Medlars collection with 30 queries. A detailed description about the collection can be found in Appendix A.2.

2.1.3 TIME collection

Time is a collection of 423 articles from magazine Time including 83 queries and their relevance list. A detailed description about the collection can be found in Appendix A.3.

2.1.4 CRAN collection

CRAN is a collection of 1400 aerodynamics abstracts from the Cranfield collection including 225 queries with relevance assessments. A detailed description about the collection can be found in Appendix A.4.

2.2 The „vein.hu" collection

The World-Wide Web is a network of electronic documents stored on dedicated computers (Web-servers) around the world. Documents on the Web can contain different types of data, such as text, image, or sound. They are stored in units referred to as Web *pages*. Each page has a unique code, called URL (Universal Resource Locator), which identifies its location on a server.

Most Web documents are in HTML (Hypertext Markup Language) format, containing many tags. Tags can provide important information about the page. For example, the tag <*b*>, which is a bold typeface markup, usually increases the importance of the term it refers to, or the tag <*title*> defines a title text for the page which should increase the importance of the term(s) it refers to even more.

In traditional Information Retrieval, documents are typically well-structured. For example, scientific journals, books and newspaper articles have their typical formats and structures. Such documents are carefully written and are checked for grammar and style. As opposed to this, there does not exist a generally recommended or prescribed format which should be followed when writing a Web page. They are more diverse:

- they can be written in any language, moreover, several languages may be used within the same page,
- the grammar of the text in a page may not always be checked very carefully,
- the length of pages and the styles used varies to a great extent.

Web pages can be hyperlinked, which generates a linked network of Web pages. Factors like

- a Universal Resource Locator from a Web page to another page,
- anchor text (the usually underlined, clickable text in a Web page)

can provide additional information about the importance of the target page.

The standard test collections introduced in Section 2.1 – because of the aforementioned special characteristics of the Web – are not appropriate for forming judgements about the effectiveness of retrieval methods developed particularly for the Web.

In order to evaluate the effectiveness of WebCIR (a Web search engine introduced later in Chapter 7), we created a test collection by downloading pages from the sites of the University of Pannonia. This task (crawling) was carried out

using Nutch's crawler. Nutch [4][21] is an open source web search software package suitable for implementing and testing new IR methods. It is based on Lucene Java [2]. The collection reflects the state of the university domains "uni-pannon.hu" and "vein.hu" as of 13th April 2008. As the name change from "University of Veszprém" to "University of Pannonia" of the institute was still in progress at that time, crawling of both domains was necessary. Files having HTML, PDF or Microsoft Word Document formats were downloaded and indexed from 129 different sites of the domains, which resulted in 60,869 documents and 669,383 index terms. The majority of texts was Hungarian, while the rest English, German, and French.

2.3 Evaluation methods and measures

In this section, a brief description is provided about the relevance effectiveness measures, which were used to compare the newly developed retrieval methods introduced in Chapters 5 - 6.

2.3.1 Main concepts

The effectiveness of an information retrieval system (or method) means how well (or bad) it performs. Effectiveness is numerically expressed by effectiveness measures which are elaborated based on different categories such as [46]:

- Relevance,
- Efficiency,
- Utility,
- User satisfaction.

Relevance effectiveness is the ability of a retrieval method or system to return relevant answers. The traditional (and widely used) measures are the following:

- *Precision*: the proportion of relevant documents out of those returned.
- *Recall*: the proportion of returned documents out of the relevant ones.
- *Fallout*: the proportion of returned documents out of those nonrelevant.

Attempts to balance these measures have been made, and various other complementary or alternative measures have been proposed [19][78][12]. In Subsection 2.3.2, the three above mentioned, widely accepted and used measures as well as the precision-recall measurement method are introduced, as they were used to measure the relevance effectiveness of the developed Entropy- and Probability-based retrieval methods introduced in Chapter 5.

The in vivo measurement of a Web search engine's relevance effectiveness using traditional precision/recall measurement is known to be impossible [51]. Recall and fallout cannot be measured (however methods have been suggested to estimate it: [32][17][42][18][64]), since we do not know all the documents on the Web. This

means that the measurement of relevance effectiveness of search engines requires other measures than the traditional ones. The measurement of relevance effectiveness of a Web search engine is, typically (due to the characteristics of the Web), user centred [13]. It is an experimentally established fact that the majority of users examine, in general, the first two pages of a hit list [9][65]. Thus, the search engine should rank the most relevant pages in the first few pages. When elaborating such new measures, one is trying to use traditional measures (for example, precision which can be calculated also for a hit list of a search engine), and to take into account different characteristics of the Web. The methods used for evaluating the newly developed WebCIR Web search engine (introduced in Chapter 7) are described in Subsections 2.3.3 through 2.3.6.

2.3.2 Precision-recall method

The precision-recall measurement method is being used in the *in vitro* (i.e., under laboratory conditions, in a controlled and repeatable manner) measurement of relevance effectiveness [6]. In this measurement method, test collections are used (some introduced in Section 2.1).

Let D denote a collection of documents, q a query, and

- $\Delta \neq 0$ denote the total number of relevant documents to query q,
- $\kappa \neq 0$ denote the number of retrieved documents in response to query q,
- α denote the number of retrieved and relevant documents.

From the point of view of practice, it is reasonable to assume that the total number of documents to be searched, M, is greater than those retrieved, i.e., $|D| = M > \Delta$. The usual relevance effectiveness measures are defined formally as follows:

1. *Recall* ρ is defined as $\rho = \frac{\alpha}{\Delta}$.
2. *Precision* π is defined as $\pi = \frac{\alpha}{\kappa}$.
3. *Fallout* φ is defined as $\varphi = \frac{\kappa - \alpha}{M - \Delta}$.

Figure 2.1 shows a visual representation of these measures. From the above definitions 1., 2., 3., it follows that:

- $0 \leq \rho \leq 1;\ 0 \leq \pi \leq 1$,
- $\rho = 0 \Leftrightarrow \pi = 0;\ \pi = 1 \Leftrightarrow \varphi = 0$,
- $\alpha = \kappa = \Delta \Leftrightarrow (\rho = \pi = 1 \wedge \varphi = 0)$.

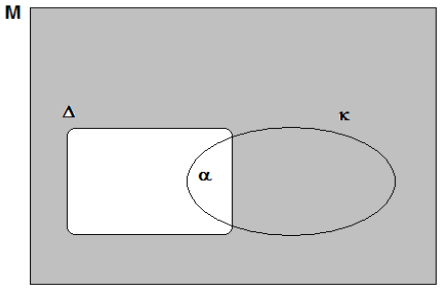


Figure 2.1 Visual representation of quantities which define precision, recall, fallout.

For every query, retrieval should be performed, using the retrieval method whose relevance effectiveness is to be measured. The hit list is then compared with the relevance list corresponding to the query under focus. The following recall levels are considered to be standard levels:

0.1; 0.2; 0.3; 0.4; 0.5; 0.6; 0.7; 0.8; 0.9; 1;

(these levels can also be given as %, for example 0.1 = 10%). For every query, pairs of recall and precision are computed. If the computed recall value is not standard (i.e. it is not in the list above), it is approximated. The precision values corresponding to equal recall values are averaged.

When the computed recall value r is not equal to a standard level, the following interpolation method can be used to calculate the precision value $p(r_j)$ corresponding to the standard recall value r_j:

$$p(r_j) = \max_{r_{j-1}<r\le r_j} p(r), \quad j = 1,\ldots,10.$$

It is known from practice that the values $p(r_j)$ are monotonically decreasing. Thus, the value $p(r_0)$ is usually determined to have $p(r_0) \ge p(r_1)$. For all queries q_i, the precision values $p_i(r_j)$ can be averaged at all standard recall levels as follows:

$$P(r_j) = \frac{1}{n}\sum_{i=1}^{n} p_i(r_j), \quad j = 0,...,10,$$

where n denotes the number of queries used. Figure 2.2 shows a typical precision-recall graph (for the test collection ADI).

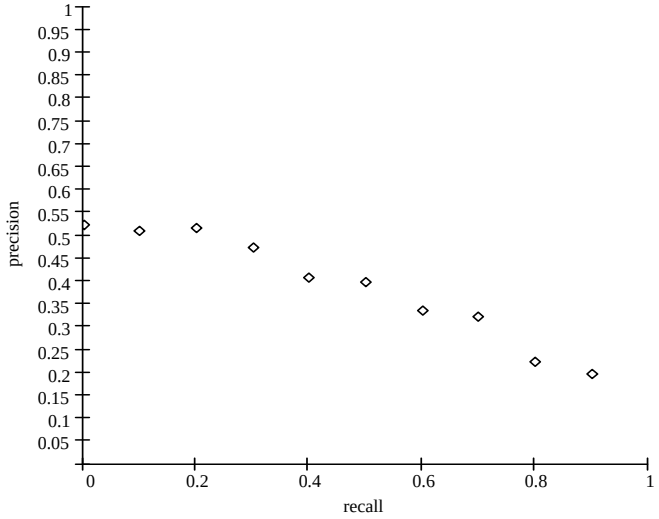


Figure 2.2 Typical precision-recall graph (for the test collection ADI)

The average of the values $P(r_j)$ is called MAP (*Mean Average Precision*). MAP can also be computed just at the recall values 0.3, 0.6, and 0.9.

2.3.3 MLS method

The MLS method [24], based on principles given in [42], measures the ability of a search engine to rank relevant hits within the first 5 or 10 hits, and involves user assessments. The MLS method is as follows:

1. Select search engine to be measured.
2. Define relevance categories, groups, and weights.
3. Give queries $\underline{Q}_i$ (i = 1,...,s).
4. Compute $P5_i$ and/or $P10_i$ for $\underline{Q}_i$ (i = 1,...,s).
5. The first 5/10-precision of the search engine is:

$$Pk = \frac{1}{s}\sum_{i=1}^{s} Pk_i \text{ , where } k = 5 \text{ or } k = 10.$$

In my experiments k=10. There are two relevance categories: irrelevance and relevance. The first ten hits are grouped into three groups as follows:

1. group: the first two hits,
2. group: the next three hits,
3. group: the rest of five hits.

Groups 1 and 2 are based on the assumption that, in practice, the most important hits are the first five (usually on the first screen) [24]. Hits within the same group receive equal weights. The weights reflect the fact that the user is more satisfied if the relevant hits appear on the first screen. According to [24] the group weights were chosen as follows: 20, 17, and 10, respectively. The *P*10 measure is as follows:

$$\frac{r\_hit_{1-2} \times 20 + r\_hit_{3-5} \times 17 + r\_hit_{6-10} \times 10}{141 - (miss\_hit_{1-2} \times 20 + miss\_hit_{3-5} \times 17 + miss\_hit_{6-10} \times 10)} \quad (2.1)$$

where

r_hit_{x-y} denotes the number of relevant hits within ranks x through y,

$miss\_hit_{u-v}$ denotes the number of missing hits within ranks u through v.

2.3.4 DCG method

The DCG (Discounted Cumulative Gain;[37]) method makes it possible to measure the cumulative gain a user obtains by examining the hits.. Given a ranked hit list H: 1,…, i,…,n, with the corresponding relevance degrees r_1,…,r_i,…,r_n. The gain is cumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks The DCG_p measure at rank p (p = 2,…,n) is defined as follows:

$$DCG_p = \sum_{j=1}^{p} \frac{2^{r_j} - 1}{\log_2(j+1)} \quad (2.2)$$

In my experiments r_i = 1 (relevant), r_i = 0 (irrelevant), and n = 5. For determining relevance degrees r_1,…,r_i,…,r_n relevance judgements are required, thus this method also involves user's assessments.

2.3.5 RC method

The RC (Reference Count; [86]) method allows ranking search engines without relevance judgements. Given a query Q and n search engines. Let $L_i = d_{1i}, \ldots, d_{ji},..., d_{mi}$ be the hit list returned by search engine i in response to Q (i = 1,…,n).

Let $o(d_{ji})$ denote the number of occurrences of d_{ji} in all other hit lists. The $RC_{Q,i}$ measure is calculated for a given query Q and search engine i as follows:

$$RC_{Q,i} = o(d_{1i}) + \ldots + o(d_{ji}) + ... + o(d_{mi}),\ i = 1,\ldots,n \quad (2.3)$$

Then, the value of $RC_{Q,i}$ should be computed for several queries, and an average should be taken:

$$RC_i = \frac{1}{s} \sum_{k=1}^{s} RC_{Q_k,i},$$

where s denotes the total number of queries. Finally, the search engines are ranked ascendingly on RC_i. In my experiments $m = 5$.

2.3.6 RP method

The RP method [24] can be used to compute a relative precision of a search engine compared to other (reference) search engine(s), without relevance judgement. Let q be a query. Let V be the number of hits returned by the search engine under focus, and T those hits out of these V that were ranked by at least one of the reference search engines within the first m of their hits. Then, $RP_{q,m}$ is calculated as follows:

$$RP_{q,m} = \frac{T}{V} \tag{2.4}$$

The value of relative precision should be computed for several queries, and an average should be taken. The steps for computing relative precision are as follows:

1. Select the search engine to be measured. Define queries q_i, $i = 1,...,n$.
2. Define the value of m; typically $m = 5$ or $m = 10$.
3. Perform searches for every q_i using the search engine as well as the reference search engine(s), $i = 1,...,n$.
4. Compute relative precision for q_i using eq. (2.4).
5. Compute average: $\frac{1}{n}\sum_{i=1}^{n} RP_{q_i,m}$

In my experiments $m = 5$.

3 A measure theoretic approach to information retrieval

In this chapter, different definitions of IR are recalled and analyzed. Based on the main common concepts they share, a new formal definition is given using the mathematical concepts of topological space and measure.

3.1 General formal framework for information retrieval

In this section, several – commonly accepted – definitions of IR are recalled first as they appeared in major works published in the field over the years. Note that these definitions are not definitions in a strict mathematical or logical sense, they are rather descriptions of what the concept of IR is or should be.

In 1965, Salton defines IR as follows [59]:

"The SMART retrieval system takes both documents and search requests in unrestricted English, performs a complete content analysis automatically, and retrieves those documents which most nearly match the given request."

In 1979, Van Rijsbergen gives the following definition [78]:

"In principle, information storage and retrieval is simple. Suppose there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by this question."

Some years later (in 1986), Salton phrases as follows [60]:

"An automatic text-retrieval system is designed to search a file of natural-language documents and retrieve certain stored items in response to queries submitted by the user."

The meaning of the word "certain" in the above quote is explained later on as follows:

"The effectiveness of a retrieval system is usually evaluated in terms of…recall and precision…Both query formulation and document representations can be altered to reach the desired recall and precision levels."

In 1999, Meadow et al. define IR as follows [46]:

"IR involves finding some desired information in a store of information or database. Implicit in this view is the concept of selectivity; to exercise selectivity

usually requires that a price be paid in effort, time, money, or all three. Information recovery is not the same as IR…Copying a complete disk file is not retrieval in our sense. Watching news on CNN…is not retrieval either…Is information retrieval a computer activity? It is not necessary that it be, but as a practical matter that is what we usually imply by the term."

In 1999, Berry & Browne formulate as follows [11]:

"We expect a lot from our search engines. We ask them vague questions … and in turn anticipate a concise, organised response. … Basically we are asking the computer to supply the information we want, instead of the information we asked for. … In the computerised world of searchable databases this same strategy (i.e., that of an experienced reference librarian) is being developed, but it has a long way to go before being perfected."

In the same year, Baeza-Yates and Ribeiro-Neto write [6]:

"In fact, the primary goal of an IR system is to retrieve all the documents which are relevant to a user query while retrieving as few non-relevant documents as possible."

In 2000, Belew, within his cognitive and articulate FOA (Finding Out About) framework, formulates retrieval in a pragmatic way as follows [8]:

"We will assume that the search engine has available to it a set of preexisting, 'canned' passages of text and that its response is limited to identifying one or more of these passages and presenting them to the users; see Figure 1.2." (Figure 1.2 shows a user having an information need, this need is being sent to a corpus of documents in the form of a query. Some process retrieves a subset of documents which is then sent back to the user.)

A few years later, in 2003, Baeza-Yates formulates similarly to his earlier view [7]:

"IR aims at modelling, designing, and implementing systems able to provide fast and effective content-based access to large amount of information. The aim of an IR system is to estimate the relevance of information items to a user information expressed in a query."

Taking a closer look at the above definitions given for IR, one can see that, in fact, they do not give *different* interpretations for IR, rather they all define IR *the same way*. All these definitions agree that: *IR means retrieving relevant documents in*

response to a query expressing a user's information need. In other words, given documents, users, information needs, and queries – retrieve relevant documents for a given query! Analysing the concepts occurring in this definition, one can group them into two classes as follows:

a) Class 1 (concepts assumed to be given): user, information need, query, document;

b) Class 2 (concepts that express operation, process): relevance, retrieve.

Adopting a mathematical, somewhat axiomatic, approach towards defining IR, Class 1 may be conceived as containing basic concepts, whereas Class 2 as expressing certain connections or relationship between them. Thus, the following purely formal and very general mathematical framework for IR can be formulated (Table 3.1).

Table 3.1 Formal mathematical framework for IR

Information Retrieval is a framework given by:	
Basic Concepts	Relationship
User, information need, query, document	For a given user, information need, and query, *there exists a corresponding* document.

The 'relationship' expresses a requirement, aim or wish. The word 'corresponding' should be understood as a synonym for appropriate, good, relevant. The term 'there exists' is to be interpreted as 'there exists at least one' (encompassing even the case when the only corresponding element is the empty set, i.e., no appropriate documents exist). Further, whether retrieval is query-driven, or query and user driven, or query and user and information need driven, or other mixture of these, is irrelevant from a formal mathematical point of view. Both the basic concepts and relationships should be viewed at an abstract level. In principle, it is actually irrelevant what the particular interpretations of the basic concepts and relationship are, or what the specific realizations or implementations of this latter might be. They may and should be interpreted abstractly, similarly to the way we interpret, for example, the basic notions (point, line, etc.) in the axiomatic theory of Euclidean Geometry. Likewise, the relationship may mean any kind of particular algorithm, relationship, method or process, similar to the free interpretation of the axioms (incidence, etc.) of Euclidean Geometry.

Usually, in practice, IR implies a computerised automatic retrieval system. The correlation degree between query and information need is a human rather than a computer matter (i.e., this degree can hardly be entirely automatised at our present knowledge). The extent to which a query reflects the information need chiefly depends on the user and less on a computer program. Thus, in a computerised

automatic retrieval system there only are two basic concepts: query and document, which may be referred to as objects in general. Alternatively, one may say that the concepts ‘user’ and ‘information need’ condensate into one single concept: ‘query’. So, one can re-define the formal framework of Table 3.1 as follows (Table 3.2):

Table 3.2 Formal mathematical framework for automatic computerised IR

Information Retrieval is a framework given by:	
Basic Concept	Relationship
Object	For a given object, there exists a corresponding object.

3.2 Measure theoretic definition of information retrieval [Thesis 1.a]

The objects in Table 3.2, let us denote them in general by *o*, may be viewed to form a set *O*. But it is reasonable to assume more than this. The objects *o* are not merely and simply elements of the set *O*: they are not independent and isolated elements gathered at random, rather they form some structure (from certain points of view such as, e.g., topic, application, etc.). In a very general way, the objects may be conceived as elements of some space having a structure. As these objects are – generically referred to as – documents, it is quite natural to assume that ‘putting together’ or unifying two objects yields a new one, and taking common parts or intersecting two objects results in another object. (For example, ‘putting together’ two documents on IR results in another document on IR, while taking their common parts will also be a document on IR. To what extent the resulting document is redundant or new, etc., is irrelevant at this point because we are interested in a formal approach.) The mathematical concept that may be used to model such a space is that of a topological space, which is mathematical formulation of the above properties (‘putting together’, common part).

Given a set X, and let $I = [0; 1] \subset \mathbf{R}$.

Let A denote a *fuzzy set* over X, i.e., A: $X \to I$ [88]. Then, I^X denotes the fuzzy power set of X, i.e., I^X denotes the set of all mappings A.

A collection $\Phi \subseteq I^X$ of fuzzy sets A is called a *fuzzy topology* if the following conditions hold:

1) $\mathbf{0}, \mathbf{1} \in \Phi$ ($\mathbf{0}$ denotes the empty fuzzy set $\varnothing$, i.e., $\mathbf{0}$: $X \to 0$; $\mathbf{1}$ denotes the crisp set X, i.e., $\mathbf{1}$: $X \to 1$);
2) $A_j \in \Phi, j \in J \Rightarrow \cup_j A_j \in \Phi$,
3) $A_1,\ldots, A_i, \ldots, A_n \in \Phi \Rightarrow \cap_i A_i \in \Phi$,

where J is an index set; $\cup$ and $\cap$ denote fuzzy union and intersection, respectively. We say that (X, Φ) is a fuzzy *topological space* on X.

If A is a crisp subset of X, i.e., when $I = \{0, 1\}$, the concepts of (classical) topology and topological space are obtained as special cases.

If, in addition to condition 1), the condition 2) is valid for countable J, and the complement A^C of any set A also belongs to Φ, the structure (X, Φ) is called a σ-*algebra*. $\Phi = I^X$ is a topology on X and is referred to as a *discrete topology*, which is also a σ-*algebra*.

The 'relationship' in Table 3.2 is an expression of retrieval: some mechanism or process ρ takes an object o and associates to it a subset O' of objects, $O' \subseteq O$, i.e., $O' \in 2^O$. The subset O' may even be empty, or it may consist of just one or several objects. One – generally accepted – way to characterise retrieval is to assume that it is based on what and how much the given object o (interpreted as being a query) and another object o_j (considered to be a document) share (formally: whatever the word 'share' is taken to mean). An abstract formulation of 'share' can be obtained by taking the common part of a document and query, mathematically the intersection $o \cap o_j$. Some measure μ can be used to express how much is being shared, i.e., $\mu(o \cap o_j)$. Intuitively, one would expect that such a measure μ be 'consistent' with the structure of the space in the following sense: if two objects do not have any common part (i.e., $o \cap o_j = \varnothing$) then the new object resulting from their union (i.e., $o \cup o_j$) will comprise all parts of both: $\mu(o \cup o_j) = \mu(o) + \mu(o_j)$. The mathematical concept that may be used to model such a measure is that of a mathematical measure, which is defined as follows.

A *measure* on a σ-algebra (X, Φ) is a countably additive real valued function $\mu: \Phi \rightarrow [0; \infty]$, i.e. [82]:

i) $\mu(\varnothing) = 0$;

ii) $A, B \in \Phi, A \cap B = \varnothing \Rightarrow \mu(A \cup B) = \mu(A) + \mu(B)$.

Looking now at the framework defined in Table 3.2 from a purely formal mathematical perspective, IR may be defined as follows:

Definition 3.1 *Let O be a space, and μ a measure on it.* Retrieval *is a function* $\rho: O \rightarrow 2^O$, $\rho(o) = \{o_j \mid \mu(o \cap o_j) \geq \theta, j = 1,\ldots, m\}$, *where* θ *is a threshold value.*

How can the space O be particularised? One possible way is as follows. Consider a set

$$T = \{t_1, t_2,\ldots, t_i,\ldots, t_n\} \tag{3.1}$$

of elements referred to as *terms* (or index terms). Then the space O can be taken as being the fuzzy discrete topology I^T, i.e., we consider the following space [83]:

$$(T, O), \quad O = I^T, I = [0,1]. \tag{3.2}$$

Any object $o_j \in O$ of this space can be represented as follows:

$$o_j = \{(t_i, \varphi_j(t_i)) \mid t_i \in T,\ i \in \{1,\ldots, n\}\},\ \varphi_j(t_i) \in [0,1] \tag{3.3}$$

where $\varphi_j(t_i)$ may be interpreted as the weight of index term t_i in object o_j. The intersection and union are defined as follows ([88], also used subsequently). The intersection of two objects o_p and o_q, notations: $o_p=\varphi_p(t_i)$ and $o_q=\varphi_q(t_i)$, can be defined as the algebraic product as follows:

$$o_p \cap o_q = \{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i)\} \tag{3.4}$$

while their union can be defined as the algebraic sum as follows:

$$o_p \cup o_q = \{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}. \tag{3.5}$$

4 Measure theoretic aspect of “classical” retrieval methods

The cardinality κ of an object o_j can be defined as the sum of the values of its membership function as follows:

$$\kappa(o_j) = \sum_{i=1}^{n} \varphi_j(t_i) \tag{4.1}$$

The empty object ∅’ is the following object ∅’=$\{(t_i,0)|t_i \in T\}$. It can be shown that:

Theorem 4.1 *The cardinality κ is a measure on* (T,O).

Proof. We have to show that the properties i) and ii) of Section 3.2 hold:

a) The cardinality of the empty object is equal to zero. This is immediate:

$$\kappa(\varnothing') = \kappa(\{(t_i, 0) \mid t_i \in T\}) = \Sigma_i \varphi_j(t_i) = \Sigma_i 0 = 0$$

b) The cardinality of two disjoint objects o_p and o_q is equal to the sum of their cardinalities. We have

$$o_p \cap o_q = \{ (t_i, \varphi_s(t_i)) \mid t_i \in T, \varphi_s(t_i) = \varphi_p(t_i) \cdot \varphi_q(t_i) \} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i) \cdot \varphi_q(t_i) = 0, \forall p, q.$$

Hence,

$$\kappa(o_p \cup o_q) =$$

$$= \kappa[\{(t_i, \varphi_s(t_i)) \mid t_i \in T, \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i) \cdot \varphi_q(t_i)\}] =$$

$$= \kappa[\{(t_i, \varphi_s(t_i)) \mid t_i \in T, \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= \kappa(o_p) + \kappa(o_q).$$

As a consequence of Theorem 4.1 the measure μ of Definition 3.1 can be taken to be the cardinality κ as defined in (4.1) as follows:

Lemma 4.1

$$\rho(o)=\{o_j|\mu(o \cap o_j)=\kappa(o \cap o_j)=\sum_{i=1}^{n} \varphi(t_i) \cdot \varphi_j(t_i) \geq \theta\} \tag{4.2}$$

Proof. According to Theorem 4.1 the cardinality κ is a measure. From relationship (3.4) the result is immediate.

4.1 Information retrieval in linear space with general basis

4.1.1 Mathematical concepts

In mathematics, a linear (or vector) space is a special set of objects related to – in many practical cases – the set of real (or complex) numbers. In that set, there is an operation defined between its objects, and this operation obeys certain requirements: it is commutative and associative, there exists a zero vector, and every vector has an ‘inverse’ or ‘opposite’. Further, there is an operation defined between the objects and real (or complex) numbers that satisfy a number of properties.

Example. Velocities (in Physics) form a linear space (over the set of real numbers). The positions of objects in the real world may be conceived as forming a linear space.

More exactly, a real *linear space* (or real *vector space*) over the set $\mathbf{R}$ of real numbers is the structure $(\mathsf{L}, \oplus, \otimes, \mathbf{R})$, where $\oplus$ and $\otimes$ denote two operations defined as follows: $\oplus: \mathsf{L} \times \mathsf{L} \to \mathsf{L}$, $\otimes: \mathbf{R} \times \mathsf{L} \to \mathsf{L}$, if the following properties hold:

i) $a \oplus b = b \oplus a, \ \forall a, b \in \mathsf{L}$ (commutativity);

ii) $\exists e \in \mathsf{L}$ such that $a \oplus e = a, \forall a \in \mathsf{L}$ (e is referred to as the null vector);

iii) $\forall a \in \mathsf{L}\ \exists a' \in \mathsf{L}$ such that $a \oplus a' = e$ (a' is the inverse of a);

iv) $a \oplus (b \oplus c) = (a \oplus b) \oplus c, \quad \forall a, b, c \in \mathsf{L}$ (associativity);

and for $\forall r, p \in \mathbf{R}$, $\forall a, b \in \mathsf{L}$ we have:

v) $(r + p) \otimes a = (r \otimes a) \oplus (p \otimes a)$;

vi) $r \otimes (a \oplus b) = (r \otimes a) \oplus (r \otimes b)$;

vii) $(r \times p) \otimes a = r \otimes (p \otimes a)$;

viii) $1 \otimes a = a$.

The linear space is briefly denoted by L. The concept of a linear space can be similarly defined also over the set of complex numbers. The elements of a linear space L are traditionally called *vectors*, and are usually denoted by small bold letters, e.g., $\mathbf{v}$, while the elements of $\mathbf{R}$ are called *scalars*.

Given the vectors $\mathbf{v}_1,\ldots,\mathbf{v}_m$. The expression $(r_1 \otimes \mathbf{v}_1) \oplus \ldots \oplus (r_m \otimes \mathbf{v}_m)$ is called a *linear combination* of these vectors, $r_1,\ldots,r_m \in \mathbf{R}$. When the linear combination is equal to e if and only if $r_1 = \ldots = r_m = 0$, then the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ are said to be *linearly independent*, and they are linearly dependent otherwise.

A set of linearly independent vectors forms an *algebraic basis* (shortly: basis) of L if any vector of the space can be written as a linear combination of them. Every linear space has at least one basis. Each basis contains the same number of vectors, this number is referred to as the dimension of the space. If $\mathbf{g}_1,\ldots,\mathbf{g}_n \in \mathsf{L}_n$ denotes basis vectors of an n-dimensional linear space L_n, then every vector $\mathbf{v} \in \mathsf{L}_n$ can be

written as a linear combination of the basis vectors as follows: $\mathbf{v} = (p_1 \otimes \mathbf{g}_1) \oplus \ldots \oplus (p_n \otimes \mathbf{g}_n)$, where the scalars $p_1,\ldots, p_n \in \mathbf{R}$ are called the *coordinates* of the vector $\mathbf{v}$; notations: $\mathbf{v} = (p_1,\ldots, p_n) = [p_1,\ldots, p_n]^T$, where ^{T} denotes transpose.

A subset $A \subseteq \mathsf{L}$ of the space L is called a *subspace* of L if the following properties hold:

(i) if $a, b \in A$, then $a \oplus b \in A$;

(ii) if $r \in \mathbf{R}$, and $a \in A$, then $r \otimes a \in A$.

Now, it is possible to relate any object o_j to an n-dimensional real linear space L as follows. Every term t_i, i=1,…,n, is assigned to the basis vector $\mathbf{g}_i$ of the space L. Consider an arbitrary object $o_j=\varphi_j(t_i)$. The values $\varphi_j(t_1),\ldots,\varphi_j(t_n)$ of the membership function can be used to form a vector $\mathbf{v}_j$ of the space L as the following linear combination: $\mathbf{v}_j=\varphi_j(t_1)\cdot\mathbf{g}_1+\ldots+\varphi_j(t_n)\cdot\mathbf{g}_n$. Thus, every object o_j may be viewed as corresponding to a vector $\mathbf{v}_j \in \mathsf{L}$ with co-ordinates $\varphi_j(t_i)$, i.e., $o_j \equiv \mathbf{v}_j=(\varphi_j(t_1),\ldots,\varphi_j(t_n))$.

Depending on certain properties of the linear space L (general, orthogonal or orthonormal basis, with or without inner product), different retrieval methods can be naturally defined.

4.1.2 Generalised Vector Space Model

In [84] it was shown why the VSM approach conflicts with the mathematical notion of a vector space. Further on, they rightly observed that the usual similarity functions (dot product, Dice- and Jaccard-coefficient) can be written also in general basis (not just in an orthonormal basis). They interpret the metric tensor G (that is $G = \mathbf{g}_{ij} = (\mathbf{g}_1 \ldots \mathbf{g}_n)^T \times (\mathbf{g}_1 \ldots \mathbf{g}_n)$, where $\mathbf{g}_1,\ldots,\mathbf{g}_n \in \mathsf{L}_n$ denotes basis vectors of an n-dimensional linear space L_n), which they refer to as correlation matrix, of the space as expressing correlations between the index terms t_i, i=1,…,n, viewed as basis vectors. G can be used as a model of term dependences:

$G=(\mathbf{t}_i\cdot\mathbf{t}_j)_{n\times n}$, where $\mathbf{t}_i$ denotes the basis vector corresponding to term t_i. Later, in [85], an automatic method is proposed to build the correlation matrix G of index terms t_i. The value of the similarity S between a document and query is computed as the product between

- the query vector $\mathbf{q}$ expressed in the general basis,
- the metric tensor G,
- and the document vector $\mathbf{d}$ in orthonormal basis

as follows: $S=\mathbf{q}^T\cdot G\cdot\mathbf{d}$. The method is referred to as the General VSM (GVSM), and good experimental results are also reported.

4.1.3 General basis-based retrieval method (GB method) [Thesis 1.b]

Given a linear space L with a general basis $\mathbf{g}_i$. The following retrieval method is proposed:

i) Let any object o_j correspond to a vector $\mathbf{v}_j$ in this space L such that the value $\varphi_j(t_i)$, i=1,…,n, corresponds to the i^{th} coordinate of the vector $\mathbf{v}_j$.

ii) The retrieval function is given by formula (4.2).

From the point of view of an implementation, the following retrieval algorithm may be formulated.

Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, j=1,…,m. Let $TD_{n\times m}=(f_{ij})_{n\times m}$ denote the frequency term-by-document matrix of term occurrences, i.e., f_{ij} denotes the number of times term t_i occurs in document D_j. Similarly, let us consider a query $Q=(f_1,\ldots,f_i,\ldots,f_n)$, where f_i denotes the number of times term t_i occurs in query Q. Compute now, using some weighting scheme, a term-by-document weight matrix $W_{n\times m}=(\mathbf{d}^{<j>})_m=(w_{ij})_{n\times m}$ for documents ($w_{ij}=\varphi_j(t_i)$), and one for the query $\mathbf{q}=(q_1,\ldots,q_i,\ldots,q_n)$.

Let us consider a general basis $\mathbf{g}_i$ now:

$$\mathbf{g}_i = (\mathbf{g}_1\ldots\mathbf{g}_n) = \begin{pmatrix} g_{11} & \ldots & g_{1n} \\ . & \ldots & . \\ g_{n1} & \ldots & g_{nn} \end{pmatrix}, \tag{4.3}$$

where the coordinates of the basis vector $\mathbf{g}_1$ are $g_{11},\ldots,g_{n1}$, and so on. Compute the coordinates $\mathbf{d'}^{<j>}=(w'_{1j},\ldots,w'_{ij},\ldots,w'_{nj})$ of every document D_j in the general basis as follows:

$$\mathbf{d'}^{<j>} = (\mathbf{g}_i)^{-1}\cdot\mathbf{d}^{<j>}, \tag{4.4}$$

where $(\mathbf{g}_i)^{-1}$ denotes the inverse of the basis tensor $\mathbf{g}_i$ (given by (4.3)). Similarly, the coordinates $\mathbf{q}'=(q'_1,\ldots,q'_i,\ldots,q'_n)$ of the query in the general basis are computed as follows:

$$\mathbf{q'} = (\mathbf{g}_i)^{-1}\cdot\mathbf{q} \tag{4.5}$$

The similarity s_j between a document D_j and query Q is computed using formula (4.2) of Lemma 4.1 as follows:

$$s_j = \sum_{i=1}^{n} q'_i \cdot w'_{ij} \tag{4.6}$$

The expression (4.6) for the similarity s_j *does not* have the meaning of an inner product between document and query vectors: it is simply a measure given by formula (4.2) according to Definition 3.1.

Note. The expression (4.6) would only be the expression of an inner product if the matrix $\mathbf{g}_{ij} = (\mathbf{g}_1 \ldots \mathbf{g}_n)^{\mathsf{T}} \times (\mathbf{g}_1 \ldots \mathbf{g}_n)$, called metric tensor, was in the expression (4.6), i.e., $s_j = \mathbf{d'}^{<j>} \times \mathbf{g}_{ij} \times \mathbf{q'}$.

4.2 Latent Semantic Indexing retrieval method [Thesis 1.b]

Retrieval in the Latent Semantic Indexing (LSI) model ([23],[10]) can be viewed as an application of Lemma 4.1 as follows: given an approximation matrix A_k (see below) of the term-by-document matrix A. Let **b** denote a basis of the column space of A_k. The retrieval function is the cardinality measure between

(i) the coordinates, in basis **b**, of the projection of the query vector **q** onto the column space of A_k,

(ii) and the coordinates of the columns of A_k in basis **b**.

Let us describe this in details. Let $A=(w_{ij})_{n\times m}$ be a term-by document matrix, where w_{ij} is the weight of term t_i in document D_j, and **q** be a query vector. The matrix A is decomposed using singular value decomposition (SVD):

$$A = U\cdot S\cdot V^{\mathsf{T}},$$

where U and V are orthogonal matrices defining the left and right singular values of A, respectively, whereas S is the diagonal matrix of singular values of A arranged in descending order from the top of the main diagonal downwards. The rank r_A of the matrix A is equal to the number of nonzero diagonal elements of S. The first r_A columns of U (from left to right) are a basis for the column space of A. The rank-k, $k\leq$rank(A), approximation of the matrix A is given by considering only the first k singular values in S:

$$\mathrm{A_k} = \mathrm{U_k}\cdot\mathrm{S_k}\cdot\mathrm{V_k}^{\mathsf{T}},$$

(equivalently, U_k contains the first k columns of U). The columns of the matrix A_k span a k-dimensional subspace of the column space of A. The columns of the matrix A_k are vectors whose coordinates are given by $S_k\cdot V_k^{\mathsf{T}}$ in the basis defined by the columns of U_k. The query is matched against the columns of A_k using formula (4.2) of Lemma 4.1, i.e., $\mathbf{q}^{\mathsf{T}}\cdot A_k$, which re-writes as follows:

$$\mathbf{q}^{\mathsf{T}}\cdot A_k =$$

$$= A_k^{\mathsf{T}}\cdot\mathbf{q} = (U_k\cdot S_k\cdot V_k^{\mathsf{T}})^{\mathsf{T}}\cdot\mathbf{q} = (V_k\cdot S_k^{\mathsf{T}}\cdot U_k^{\mathsf{T}})\cdot\mathbf{q} = (V_k\cdot S_k^{\mathsf{T}})\cdot(U_k^{\mathsf{T}}\cdot\mathbf{q}) =$$

$$= (S_k\cdot V_k^{\mathsf{T}})\cdot(U_k^{\mathsf{T}}\cdot\mathbf{q}).$$

The elements of the vector $U_k^{\mathsf{T}}\cdot\mathbf{q}$ are the coordinates in the basis defined by the columns of U_k of the projection $U_k\cdot U_k^{\mathsf{T}}\cdot\mathbf{q}$ of the query vector **q** onto the column space of A_k.

If the expression $\mathbf{q}^{\mathsf{T}}\cdot A$ is interpreted as having the meaning of a scalar product, then it should be the same as $\mathbf{q}^{\mathsf{T}}\cdot A_k$, i.e., $\mathbf{q}^{\mathsf{T}}\cdot A=\mathbf{q}^{\mathsf{T}}\cdot A_k$. We have equality when $A=A_k$,

which only occurs for k=rank(A). Otherwise, the expression $\mathbf{q}^T \cdot A_k$ does not have the meaning of a scalar product; it is a cardinality measure.

4.3 Information retrieval in linear space with inner product

A linear space with inner product is a very special kind of linear space in that it has an operation defined between any two vectors that exhibits the following two properties:

- it does not depend on the choice of the basis of the space,
- it is not a vector but a number.

Mathematically, the concept of scalar product is defined as follows. Given a real linear space $(\mathsf{L}, \oplus, \otimes, \mathbf{R})$. A mapping $\pi: \mathsf{L} \times \mathsf{L} \to \mathbf{R}$ satisfying the properties:

(i) $\pi(v \oplus w, u) = \pi(v, u) + \pi(w, u), \forall v, w, u \in \mathsf{L}$;

(ii) $\pi(r \otimes v, w) = r \times \pi(v, w), \forall r \in \mathbf{R}, \ \forall v, w \in \mathsf{L}$;

(iii) $\pi(v, w) = \pi(w, v), \forall v, w \in \mathsf{L}$;

(iv) $\pi(v, v) \geq 0, \forall v \in \mathsf{L}$;

(v) $\pi(v, v) = 0$ if and only if $\mathbf{v} = \mathbf{0}$ (**0** denotes the null vector of the space L),

is called a *scalar* (or *inner* or *dot*) *product*. Instead of $\pi(\mathbf{x}, \mathbf{y})$ the following shorter notations may be used: $(\mathbf{x},\mathbf{y})$, $\mathbf{x} \cdot \mathbf{y}$, $\langle \mathbf{x}|\mathbf{y} \rangle$. The space (L, π) is a linear space with inner product. An example for such a space is the Hilbert space. Originally, the Hilbert space meant the set of those real infinite series X (i.e., X=x_1,…, x_n,…) for which the sum of the squares of absolute values (i.e., $\Sigma_i |x_i|^2$) was convergent (i.e., the sum existed and was finite). Nowadays, the Hilbert space is regarded as being a more abstract concept. It is an important concept in modern mathematical analysis, but it has also been used to formalise, for example, quantum mechanics ([80]). With the use of a Hilbert space, a geometry for the space under focus can be developed. The concept of the Hilbert space can be mathematically defined as follows.

Given a linear space L. A function $\nu: \mathsf{L} \to [0; +\infty)$ is called a *pseudo-norm* if the following properties hold:

(i) if v is the null vector of L then $\nu(v) = 0$;

(ii) $\nu(r \otimes v) = |r| \times \nu(v), \forall r \in \mathbf{R}, \forall v \in \mathsf{L}$;

(iii) $\nu(v \oplus w) \leq \nu(v) + \nu(w), \forall v, w \in \mathsf{L}$.

If, in addition to these properties, the function ν obeys also the following property:

(iv) if $\nu(v) = 0$ then v is the null vector of L,

then the function ν is called a *norm*. Usually, the norm $\nu(\mathbf{v})$ is denoted by $\|\mathbf{v}\|$.

Example. Heading North with 90 km/h by car describes a velocity vector having norm (i.e., magnitude) 90 km/h. Notice that velocity has also a direction (apart from magnitude).

A linear space L with a norm is called a *normed* (linear) *space*. A normed linear space $(\mathsf{L}, \|.\|)$ defines a metric space (L, δ) with the metric $\delta(\mathbf{v},\mathbf{w})=\|\mathbf{v}\oplus(-1)\otimes\mathbf{w}\|$. The expression $\|\mathbf{v}\oplus(-1)\otimes\mathbf{w}\|$ is usually written as $\|\mathbf{v}-\mathbf{w}\|$, and the thus defined metric is called a *metric induced by the norm*.

Example. The distance of two objects in real space is a metric (called Euclidean metric).

Given a normed linear space $(\mathsf{L}, \|.\|)$. A sequence $x_1,\ldots, x_n,\ldots\in \mathsf{L}$ is said to be *convergent* if it has a *limit*, denoted by, say, L; i.e., $\forall\varepsilon>0 \;\; \exists n_\varepsilon\in\mathbf{N}$ such that $\|x_n-L\|<\varepsilon$, $\forall n>n_\varepsilon$, where $\mathbf{N}=\{1,2,\ldots,n,\ldots\}$ denotes the set of natural numbers.

The space is said to be *complete* (relative to the metric induced by the norm) if the *Cauchy principle of convergence* holds, i.e., the sequence $x_1,\ldots, x_n,\ldots\in \mathsf{L}$ is convergent if and only if $\forall\varepsilon>0 \;\; \exists n_\varepsilon\in\mathbf{N}$ such that $\|x_m-x_n\|<\varepsilon$, $\forall m,n>n_\varepsilon$. If the space L is complete relative to the metric induced by the norm $\|.\|$, then the space L is called a *Banach space*.

A Banach space in which the norm is defined using the scalar product as follow: $\|\mathbf{v}\| = (\mathbf{v}\cdot\mathbf{v})^{1/2}$, is called a *Hilbert space*. Two elements $\mathbf{u}$ and $\mathbf{v}$ of the Hilbert space L, $\mathbf{u},\mathbf{v}\in \mathsf{L}$, are said to be *orthogonal* if $(\mathbf{u},\mathbf{v})=0$; notation: $\mathbf{u}\perp\mathbf{v}$.

Two subsets A and B of L are said to be orthogonal, notation: $A\perp B$, if $\mathbf{u}\perp\mathbf{v}$, $\forall\mathbf{u}\in A$, $\forall\mathbf{v}\in B$. When the subset A consists of only one element, $A=\{\mathbf{u}\}$, then the notations $A\perp B$ and $\mathbf{u}\perp B$ are considered to be equivalent with each other. For a subset $A\subset\mathsf{L}$ of the Hilbert space L, the set $A^\perp=\{\mathbf{u}\in\mathsf{L} \mid \mathbf{u}\perp A\}$ is referred to as the *orthogonal complement* of A. The subset of all linear combinations of the elements of A is a subspace of the space L, and it is referred to as the *subspace generated* by A; notation: $Sp(A)$. It can be demonstrated that the following theorem holds ([31],[57]):

Theorem 4.2. Given a Hilbert space L, and a closed linear subspace $A\subset\mathsf{L}$. Then any element $\mathbf{u}\in\mathsf{L}$ can be represented as $\mathbf{u}=\mathbf{v}\oplus\mathbf{w}$ in a unique way, where $\mathbf{v}\in A$, $\mathbf{w}\in A^\perp$.

The element $\mathbf{v}$ in Theorem 4.2 is called the *projection* of $\mathbf{u}$ on A; notation: $\mathbf{v} = [A]\mathbf{u}$. An operation P defined as $P_A(\mathbf{x}) = [A]\mathbf{x}$, i.e., giving the projections of the elements $\mathbf{x}$ of a Hilbert space on A, is called a *projector*.

If the relevance measure μ in Lemma 4.1 is interpreted as being the expression of the inner product of the space L, then the following holds:

Theorem 4.3. *Given a real Hilbert space* L *for object-documents D having weight vectors* $\boldsymbol{w}\in\mathsf{L}$, *and an inner product similarity function* $\rho:\mathsf{L}\times\mathsf{L}\to\mathbf{R}_+$ ($\mathbf{R}_+$ *denotes the set of positive real numbers*). *Retrieval relative to an object-query Q represented as a vector* $\boldsymbol{q}\in\mathsf{L}$ *can be performed using a projector* P_A *as follows:* $\Re_Q=\{D|\boldsymbol{w}=P_A(\boldsymbol{w})+\boldsymbol{q}, A=\{\boldsymbol{q}\}^\perp\}$.

Proof. Retrieval of object-documents D represented as vectors $\mathbf{w}$ in response to an object-query Q represented as a vector $\mathbf{q}$ means constructing the set (inner product similarity function by assumption)

$$\mathfrak{R}_Q = \{D \mid \mathbf{q}\cdot\mathbf{w} \neq 0\}.$$

The orthogonal complement $A=\{\mathbf{q}\}^{\perp}$ (i.e., the set of object-documents which do not share common terms with the object-query) corresponding to the query Q is given by those object-documents D whose vectors $\mathbf{w}$ are perpendicular to $\mathbf{q}$, i.e.,

$$A = \{\mathbf{q}\}^{\perp} = \{\mathbf{w} \mid \mathbf{w} \perp \{\mathbf{q}\}\} = \{\mathbf{w} \mid \mathbf{w}\cdot\mathbf{q} = 0\}.$$

Hence, the set $\mathfrak{R}_Q$ is equal to the set of those object-documents whose vectors form the complement $\mathsf{C}_{\mathsf{L}}(A)$ of the set A relative to the space L (apart from the query Q itself, of course), i.e.,

$$\mathfrak{R}_Q = \{D \mid \mathbf{q}\cdot\mathbf{w} \neq 0\} = \{D \mid \mathbf{w} \in \mathsf{C}_{\mathsf{L}}(A) \setminus \{Q\}\}.$$

Because the set A is a closed linear subspace of the space L it follows that any element $\mathbf{w}\in\mathsf{L}$ of the space L can be uniquely written as $\mathbf{w}=\mathbf{v}+\mathbf{u}$, where $\mathbf{v}\in A$ and $\mathbf{u}\in A^{\perp}=\{\mathbf{q}\}$. The projector $P_A(\mathbf{w})$ for the elements $\mathbf{w}\in\mathsf{L}$ of the space L on the set A is defined as $P_A(\mathbf{w})=\mathbf{v}$, $\mathbf{v}\in A$. Thus, the set $\mathfrak{R}_Q$ contains those object-documents D whose vectors $\mathbf{w}$ are so projected on A that $\mathbf{u}=\mathbf{q}$, i.e.,

$$\mathfrak{R}_Q = \{D \mid \mathbf{w} = P_A(\mathbf{w}) + \mathbf{q}, A = \{\mathbf{q}\}^{\perp}\}.$$

Van Rijsbergen gives a geometrical treatment of retrieval [79], relating it to a concept of probability following von Neumann, in the very special case of a Hilbert space.

4.4 Information retrieval in orthonormal real linear space [Thesis 1.b]

The classical vector space model (VSM) of IR is a very special case of Theorem 4.2 in that the linear space L is the orthonormal Euclidean space. An intuitive model of an Euclidean space is the space we live our everyday lives in. Mathematically, it is defined as follows.

Given a linear space $(\mathsf{L}, \oplus, \otimes, \mathbf{R})$ as follows:

a) the set L is equal to the set of n-tuples $(v_1,\ldots,v_n)\in\mathbf{R}^n$ of real numbers, i.e, $\mathsf{L}=\mathbf{R}^n$;

b) the operation $\oplus$ is defined as follows: $\oplus=+$; $\mathbf{v}=(v_1,\ldots,v_n)$, $\mathbf{w}=(w_1,\ldots,w_n)$, $\mathbf{v}+\mathbf{w}=(v_1+w_1,\ldots,v_n+w_n)$;

c) the operation $\otimes$ is defined as follows: $\otimes=\times$; $r\times\mathbf{v}=(r\times v_1, \ldots, r\times v_n)$;

d) the norm is defined as the Euclidean length of a vector, i.e., $\|\mathbf{v}\|=\sqrt{\sum_{i=1}^{n} v_i^2}$.

The linear space ($\mathbf{R}^n$, +, ×, **R**) can be turned into a Hilbert space, and hence also into a Banach space, by defining the following operation as a scalar product:

$$(\mathbf{v},\mathbf{w}) = (v_1,\ldots,v_n)\cdot(w_1,\ldots,w_n) = v_1\times w_1+ \ldots +v_n\times w_n = \|\mathbf{v}\|\cdot\|\mathbf{w}\|\cdot\cos\varphi.$$

The space ($\mathbf{R}^n$, +, ×, **R**) with the above scalar product is usually referred to as an *Euclidean space*; notation: E_n. In a Euclidean space, the Euclidean distance between two vectors $\mathbf{v}=[v_1,\ldots,v_n]^T$ and $\mathbf{w}=[w_1,\ldots,w_n]^T$ is defined as follows:

$$\|\mathbf{v} +(-1)\times\mathbf{w}\| = \sqrt{\sum_{i=1}^{n}(v_i - w_i)^2} .$$

A measure of the angle φ between two vectors $\mathbf{v},\mathbf{w}\neq\mathbf{0}\in\mathbf{R}^n$ is a real number φ such that $\cos\varphi=(\mathbf{v},\mathbf{w})$ / ($\|\mathbf{v}\|\times\|\mathbf{w}\|$). Two vectors **v** and **w** are orthogonal to each other if $\cos\varphi=0$. Any *n*-dimensional Euclidean space E_n has an orthonormal (i.e., orthogonal and unit lengths) basis (there may be other bases, too, which need not be orthogonal or with unit lengths). A common orthonormal basis is: $\mathbf{e}_1=[1,0,0,\ldots,0]^T$, $\mathbf{e}_2=[0,1,0,\ldots,0]^T$, …, $\mathbf{e}_n=[0,0,0,\ldots,1]^T$, where $(\mathbf{e}_i,\mathbf{e}_j)=\delta_{ij}$ (δ_{ij} is the Kronecker delta symbol, i.e., $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ if $i \neq j$).

In 1975 Salton et al. [62], use a linear space as a formal framework for automatic indexing and retrieval. More than a decade later, in 1988, Salton and Buckley re-use this framework [61], and formulate as follows:

"In the late 1950's, Luhn first suggested that automatic text retrieval systems could be designed based on a comparison of content identifiers attached both to the stored texts and to the users' queries. The documents would be represented by term vectors of the form $D=(t_i,t_j,\ldots,t_p)$ where each t_k identifies a content term assigned to some sample document *D*. Analogously, a typical query vector might be formulated as $Q=(q_a, q_b,\ldots,q_r)$. A more formal representation of the term vectors is obtained by including in each term vector all possible content terms allowed in the system and adding term weight assignments to provide distinctions among terms. Thus, if w_{d_k} (or w_{q_k}) represents the weight of term t_k in document *D* (or query *Q*), and t terms in all are available for content representation, the term vectors for document and query can be written as $D=(t_0,w_{d_0}; t_1,w_{d_1};\ldots;t_t,w_{d_t})$ and $Q=(q_0,w_{q_0}; q_1,w_{q_1};\ldots;q_t,w_{q_t})$. Given the vector representations, a query-document similarity value may be obtained by comparing the corresponding vectors, using for example the conventional vector product formula similarity $(Q,D)=\Sigma w_{q_k}w_{d_k}$. When the term weights are restricted to 0 and 1 as previously suggested, the vector product measures the number of terms that are jointly assigned to query *Q* and document *D*. In practice it has proven useful to provide a greater degree of discrimination among terms assigned for content representation than is possible with weights of 0 and 1 alone. The weights could be allowed to vary continuously between 0 and 1, the higher weight assignment near 1

being used for the most important terms, whereas lower weights near 0 would characterize the less important terms. A typical term weight using a vector length normalisation factor is $w_{dk}/(\Sigma_{\text{vector}}(w_{di})^2)^{1/2}$ for documents. When a length normalized term-weighting system is used with the vector similarity function, one obtains the well-known cosine similarity formula.”

One can easily see that a document $D=(t_0,w_{d0};\ t_1,w_{d1};\ldots;t_t,w_{dt})$ and query $Q=(q_0,w_{q0};\ q_1,w_{q1};\ldots;q_t,w_{qt})$ are objects, and the similarity is a measure of the intersection. Any object o may be conceived as being an element of the real linear space $\mathbf{R}^n$, $n=t+1$, of n-tuples in $\mathbf{R}$, i.e., $(w_{o0},w_{o1},\ldots,w_{ot})\in\mathbf{R}^n$. The similarity is the inner product of this space. Because there is a one-to-one correspondence between the linear space $\mathbf{R}^n$ and the n-dimensional Euclidean space E_n, we may say that the formal framework of VSM is E_n. Thus, each term t_i corresponds to a basis vector $\mathbf{e}_i$ in the orthonormal linear space E_n. As E_n is at the same time a Hilbert space relative to similarity, Theorem 4.2 holds.

4.5 Principle of Object Invariance

Let us begin this part with a little physics. The concepts of position, translation, rotation, velocity, acceleration, force, etc. are physical concepts (and modelled formally as vectors). But they are not just abstract or mathematical concepts [30]. They reflect certain aspects of reality, and thus possess underlying properties such that the physical laws are the same in any coordinate system regardless of the basis of the space. For example, the position of a physical object in space does not depend on the angle from which we look at it or on the choice of the coordinate axes (i.e., on the choice of the basis of the space). The position of the physical object is invariant with respect to the basis of the space; the same holds for velocity, force, etc.. Such entities are referred to as vectors, for short. In other words, vectors are entities that possess an ‘identity’, and this identity is being preserved in any system or basis, i.e., it is invariant with respect to the change of the basis of the space. An immediate – but very important – consequence of this is that the scalar product of two vectors is also preserved, i.e, is invariant with respect to the choice of the basis of the space. In other words, apart from vectors, the scalar product is another quantity that is basis-invariant. The mathematical apparatus developed to correctly deal with the physical operations involved (e.g., the addition of velocities) is referred to as vector algebra or tensor calculus (see, for example, [41] or [66]).

As it is well-known, one of the basic concepts of IR is that of document, i.e., that of objects or entities to be searched. The notion of document is not merely a mathematical or abstract concept. Just as in physics, it is used to reflect certain aspects of reality. But as opposed to physics, a document need not have a basis-invariant ‘identity’ (meaning, content, property), it may depend on the point of view or on the judgement (or mathematically: on basis of the space) of the user. As a consequence, and as seen already, even if the space O is assumed to be, or is related to, a linear space, the similarity function (formula 4.2) need not necessarily be

viewed as being the expression of an inner product – this is rather an option or hypothesis that we may or may not accept, or accept to a certain extent. Thus, it is reasonable to introduce the following principle:

PRINCIPLE OF OBJECT INVARIANCE (POI). In a retrieval environment or system, i.e., given (O,ρ), the identity of objects is preserved with a probability π.

The objects may be documents and queries. The extent to which their identities are preserved, i.e., the extent to which they (their meaning, content) remain the same, can be characterised or given by a probability π.

The case when $\pi=1$ means that the identity of documents and queries does not change, it remains the same, regardless of the basis of the space; in one word: the documents and queries *are* conceived as vectors. In this case, the similarity function (formula 4.2) is the expression of an inner product, hence the similarity value between documents and queries does not change with the change of basis. An immediate and very important consequence is that using the orthonormal Euclidean space is as good as any other linear space (so, it is useless to consider other linear space)!

If, however, $\pi<1$, the identity of documents and queries does depend on a point of view or interpretation, i.e., on the basis of the space. Then, the documents *are not* vectors. But then, the similarity function is not the expression of an inner product! An immediate consequence is that it does matter whether the linear space used is orthonormal Euclidean, or Hilbert (which has an inner product) or Banach (which does not have an inner product).

Using a pseudocode, the POI can be expressed compactly and formally as follows:

IF $\pi = 1$ THEN (classical VSM)

- the documents and queries are basis-invariant, i.e., they preserve their identity regardless of the basis of the space (they *are* vectors!),
- the expression $\kappa(o \cap o_j)$ of similarity has the meaning of an inner product.

IF $\pi < 1$ THEN

- the documents and queries do depend on the basis of the space, (they *are not* vectors!),
- the expression $\kappa(o \cap oj)$ of similarity is a measure but it is *not* the expression of an inner product.

It is important to emphasise that the formula 4.2 as given in Lemma 4.1 (cardinality of intersection: $\kappa(o \cap o_j)$) remains valid for any value of the probability π. Thus, we may conclude that with POI, Definition 3.1, and formula (4.1):

a) the classical vector space retrieval model,

b) the latent semantic indexing retrieval model,

c) the generalised vector space retrieval model,

gain a correct formal mathematical formulation and background that is consistent with practice.

5 Entropy- and probability-based retrieval

In this Chapter, the Entropy- and the Probability-based retrieval methods are proposed as derived naturally from the definition introduced in Section 3.2.

5.1 Entropy-based information retrieval [Thesis 2.a]

Apart from the GB retrieval method in linear space with general basis proposed in subsection 4.1.3, another retrieval method can also be proposed as follows.

We particularise the set O of objects as in Chapter 3. Let $H(o_j)$ denote a fuzzy entropy of o_j and define it as follows:

$$H(o_j) = -\sum_{i=1}^{n} \varphi_j(t_i) \cdot \log_b(\varphi_j(t_i)) \tag{5.1}$$

where $b>1$.

It can be shown that:

Theorem 5.1. *The fuzzy entropy H is a measure on* (T,O).

Proof. We have to show that:

a) The entropy of the empty object is equal to zero. This is immediate:

$$H(\varnothing') = H(\{(t_i,0) \mid t_i \in T\}) = \lim_{\varphi_j(t_i)\to 0}\left(-\sum_{i=1}^{n} \varphi_j(t_i) \cdot \log(\varphi_j(t_i))\right) = 0.$$

b) The entropy of two disjoint objects o_p and o_q is equal to the sum of their entropies. We have

$$o_p \cap o_q = \{(t_i,\varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i)\} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i)\cdot\varphi_q(t_i) = 0,\ \forall p, q.$$

Hence:

$$H(o_p \cup o_q) =$$

$$= H[\{(t_i,\ \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}] =$$

$$= H[\{(t_i,\ \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= H(o_p) + H(o_q).$$

As a consequence of Theorem 5.1 the measure μ of Definition 3.1 can be taken to be the fuzzy entropy *H* as given by (5.1):

Lemma 5.1.

$$\rho(o) = \{o_j \mid \mu(o \cap o_j) = -\sum_{i=1}^{n} \varphi(t_i) \cdot \varphi_j(t_i) \cdot \log_b(\varphi(t_i) \cdot \varphi_j(t_i)) \geq \theta\} \qquad (5.2)$$

Proof. According to Theorem 5.1 the fuzzy entropy *H* is a measure. From relationship (3.4) the result is immediate.

5.1.1 Entropy-based retrieval method

From the point of view of an implementation, the following retrieval method may be formulated.

Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, $j=1,\ldots, m$. Let $W_{n\times m}=(w_{ij})_{n\times m}$ denote a term-by-document matrix, where $w_{ij}\in[0;1]$ is the weight of term t_i in document D_j. Given a query Q, query weights $q_1,\ldots,q_i,\ldots,q_n$ are computed; q_i denotes the weight of term t_i in Q. The similarity s_j between a document D_j and query Q and is computed using formula (5.2) as follows:

$$s_j = -\sum_{i=1}^{n} q_i \cdot w_{ij} \cdot \log(q_i \cdot w_{ij})$$

The similarity value s_j above is positive, and the ranking should be performed in increasing order. If the minus sign was omitted, then s_j would be negative, and the ranking should be performed in decreasing order.

Example. Let Q = (0.5; 0.2) and D_j = (1; 0.7). Then:

$$s_j = -0.5 \times 1 \times \log(0.5\times1) - 0.2 \times 0.7 \times \log(0.2\times0.7) =$$

$$= -0.5 \times \log(0.5) - 0.14 \times \log(0.14) = 1.25.$$

5.2 Probability-based information retrieval [Thesis 2.b]

Let $p(t_i)$ denote a Kolmogoroff probability [40] of term $t_i \in T$, $i=1,\ldots,n$, in O. In fuzzy set theory, the fuzzy probability $P(o_j)$ of an object $o_j=\{(t_i,\varphi_j(t_i))|t_i \in T,\ i \in \{1,\ldots,n\}\}$ is defined as follows:

$$P(o_j) = \sum_{i=1}^{n} \varphi_j(t_i) \cdot p(t_i) \tag{5.3}$$

It can be shown that:

Theorem 5.2. *The fuzzy probability P is a measure on* (*T*,*O*).

Proof. We have to show that:

a) The fuzzy probability of the empty object is equal to zero. This is immediate:

$$P(\varnothing') = P(\{(t_i, 0) \mid t_i \in T\}) = \sum_{i=1}^{n} 0 \cdot p(t_i) = 0.$$

b) The fuzzy probability of two disjoint objects o_p and o_q is equal to the sum of their fuzzy probabilities. We have

$$o_p \cap o_q = \{(t_i,\varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i)\cdot\varphi_q(t_i)\} = \varnothing' \Leftrightarrow$$

$$\Leftrightarrow \varphi_p(t_i)\cdot\varphi_q(t_i) = 0,\ \forall p, q.$$

Hence:

$$P(o_p \cup o_q) =$$

$$= P[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i) - \varphi_p(t_i)\cdot\varphi_q(t_i)\}] =$$

$$= P[\{(t_i, \varphi_s(t_i)) \mid t_i \in T,\ \varphi_s(t_i) = \varphi_p(t_i) + \varphi_q(t_i)\}] =$$

$$= \sum_{i=1}^{n} (\varphi_p(t_i) + \varphi_q(t_i))\, p(t_i) = P(o_p) + P(o_q).$$

As a consequence of Theorem 5.2 the measure μ of Definition 3.1 can be taken to be the probability P of (5.3):

Lemma 5.2.

$$\rho(o) = \{o_j \mid \mu(o \cap o_j) = P(o \cap o_j) = \sum_{i=1}^{n} \varphi(t_i) \cdot \varphi_j(t_i) \cdot p(t_i) \geq \theta\} \tag{5.4}$$

Proof. According to Theorem 5.2 the fuzzy probability P is a measure. From relationship (3.4) the result is immediate.

5.2.1 Probability-based retrieval method

From the point of view of an implementation, several retrieval methods may be formulated. Given a set of index terms $T=\{t_1,\ldots,t_n\}$ and a collection of documents D_j, $j=1,\ldots,m$. Let $TD_{n\times m}=(f_{ij})_{n\times m}$ denote the term-by-document frequency matrix where f_{ij} is the number of occurrences of term t_i in document D_j. The probability $p(t_i)$ of any term t_i may be calculated as follows:

$$p(t_i)=\frac{\sum_{j=1}^{m} f_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}} \tag{5.5}$$

Let $W_{n\times m}=(w_{ij})_{n\times m}$ denote a term-by-document weight matrix, where w_{ij} is the weight of term t_i in document D_j. Given a query Q, query weights $q_1,\ldots,q_i,\ldots,q_n$ are computed; q_i denotes the weight of term t_i in Q. The similarity s_j between a document D_j and query Q and is computed using formula (5.4) as follows:

$$s_j = P(Q\cap D)=\sum_{i=1}^{n} q_i\cdot w_{ij}\cdot p(t_i)$$

Following the Language Model [53], the conditional probability $P(Q|D)$ of document D generating query Q is considered, this may be defined using the formula for conditional probability as follows:

$$s_j = P(Q\mid D)=\frac{P(Q\cap D)}{P(D)}=\frac{\sum_{i=1}^{n} q_i\cdot w_{ij}\cdot p(t_i)}{\sum_{i=1}^{n} w_{ij}\cdot p(t_i)}$$

Alternatively, a hybrid similarity measure can also be defined by combining cardinality and probability as follows:

$$s_j=\frac{\kappa(Q\cap D)}{P(D)}=\frac{\sum_{i=1}^{n} q_i\cdot w_{ij}}{\sum_{i=1}^{n} w_{ij}\cdot p(t_i)}$$

Following the Probabilistic Model [56], the conditional probability $P(D|Q)$ of a document D given a query Q is considered, this may be defined using the formula for conditional probability as follows:

$$s_j = P(D \mid Q) = \frac{P(Q \cap D)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij} \cdot p(t_i)}{\sum_{i=1}^{n} q_i \cdot p(t_i)}$$

Alternatively, a hybrid similarity measure can also be defined by combining cardinality and probability as follows (referred to as KP method):

$$s_j = \frac{\kappa(Q \cap D)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij}}{\sum_{i=1}^{n} q_i \cdot p(t_i)} \tag{5.6}$$

5.3 Experimental results for Entropy- and Probability-based retrieval methods [Thesis 2.c]

Experiments were performed to estimate the relevance effectiveness of the following retrieval methods proposed in the present work:

1. General basis-based retrieval method (GB method, proposed in subsection 4.1.3),
2. Entropy-based retrieval method (E method, proposed in part 5.1.1),
3. Cardinality-probability retrieval method (KP method, proposed in part 5.2.1, formula 5.6).

The following standard test collections were used: ADI, MED, TIME, CRAN. In a pre-processing phase they were subjected to the usual Porter-stemming and stop-listing. Because there was no need for the pre-processing modules to be written in the same language, appropriate implementation could be used for each: Porter-stemming was implemented in C++ (fast running time) and stop-listing in PHP (convenient environment, low complexity). Table 5.1 shows the statistics of these test collections.

Table 5.1 Statistics of the test collections used in experiments. The columns correspond to number of documents, number of queries, number of identified index terms, average document and query lengths with their standard deviations respectively.

Test Coll.	No. Docs (d)	No. Qrys (q)	No. Terms (t)	Avg. No (t/d)	Std. Dev (t/d)	Avg. No. (t/q)	Std. dev (t/q)
ADI	82	35	791	21	7	6	2
MED	1 033	30	7 744	45	20	9	5
TIME	423	83	13 479	193	140	8	3
CRAN	1 402	225	4 009	49	21	8	3

In the experiments for the GB method, the coordinate axes corresponding to the following index terms (stemmed) were taken to be oblique to each other (expressing the fact that the terms are not independent): ADI: writer, book; MED: cell, patient; TIME: boarder, cradl; CRAN: program, computer. For MED and TIME, the two terms were selected because they were the two most frequent terms over the documents. For ADI and CRAN, the two terms were selected because they can be naturally related to each other in general (not just in these test collections). The computational details for the general basis-based GB retrieval method are explained using the test collection MED as an example. The term 'cell' corresponds to the following orthonormal basis vector in the classical vector space model:

$\mathbf{e}_{1108}$ = (0, …, 0, 1, 0, …, 0)
1^{st} 1108^{th} 7744^{th} position

whereas the term 'patient' to the following orthonormal basis vector in the classical vector space model:

$\mathbf{e}_{5637}$ = (0, …, 0, 1, 0, …, 0)
1^{st} 5637^{th} 7744^{th} position

The fact that the terms 'cell' and 'patient' are not independent of one another is modelled by taking oblique basis vectors as follows: instead of the basis vector $\mathbf{e}_{1108}$ a new basis vector $\mathbf{g}_{1108}$ (having unit length) is taken so that it forms an angle of α with the basis vector $\mathbf{e}_{5637}$ (Figure 5.1). Thus, instead of the basis vector $\mathbf{e}_{1108}$ we will have the new basis vector $\mathbf{g}_{1108}$ as follows:

$\mathbf{g}_{1108}$ = (0, …, 0, sin(α), 0, …, 0, cos(α), 0, …, 0)
1108^{th} position 5637^{th} position

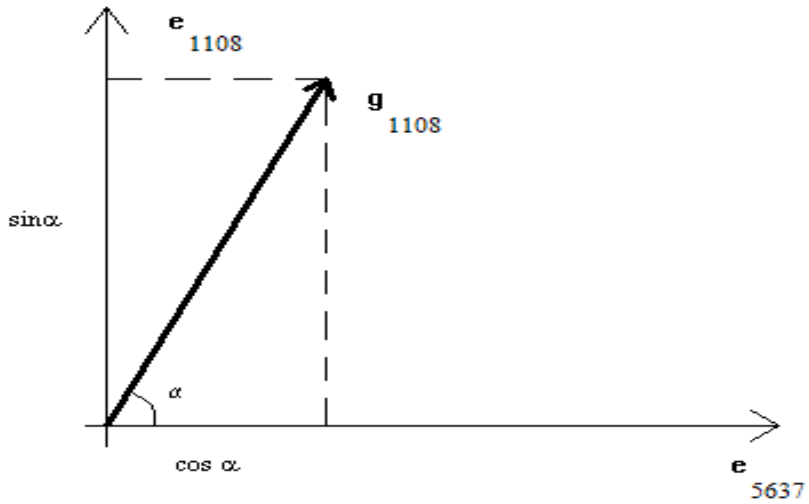


Figure 5.1 Orthonormal (e_{1108}, e_{5637}), and general (g_{1108}, e_{5637}) basis vectors

The columns $\mathbf{d}^{<j>}=(w_{1j},\ldots,w_{ij},\ldots,w_{nj})$, $j=1,\ldots, m$, of the term-by-document matrix are the coordinates of the documents in orthonormal basis. The new coordinates (i.e., new columns) $\mathbf{d'}^{<j>}=(w'_{1j},\ldots,w'_{ij},\ldots,w'_{nj})$ in the general basis $\mathbf{g}_i$

$$\mathbf{g}_i = (\mathbf{e}_1,\ldots, \mathbf{e}_{1107}, \mathbf{g}_{1108}, \mathbf{e}_{1109},\ldots, \mathbf{e}_{7744})$$

are computed as follows:

$$\mathbf{d'}^{<j>} = (\mathbf{g}_i)^{-1}\cdot\mathbf{d}^{<j>},$$

where $(\mathbf{g}_i)^{-1}$ denotes the inverse of the basis tensor. Thus, the term-by-document matrix in the general basis is obtained. Likewise, the new coordinates $\mathbf{q'}=(q'_1,\ldots,q'_n)$ of the initial query vector $\mathbf{q}=(q_1,\ldots, q_n)$ in the new basis $\mathbf{g}_i$ are computed as follows:

$$\mathbf{q'} = (\mathbf{g}_i)^{-1}\cdot\mathbf{q}.$$

The similarity ρ between the jth document and query Q is computed as follows:

$$\rho = \mathbf{d'}^{<j>}\cdot\mathbf{q'} = \sum_{i=1}^{n} w'_{ij}\cdot q'_i .$$

For each test collection, the normalised term frequency weighting scheme was used. The classical vector space (i.e., in orthonormal basis) retrieval method was also implemented and used as baseline. All three retrieval methods as well as the evaluation of retrieval effectiveness were performed using computer programs written in MathCAD (see Appendix A.). MathCAD saves considerable time during research/implementation phase, as it makes implementation and modification of algorithms possible in a pseudo-code like, easily interpretable way. The standard 11-point precision-recall values were computed (see section 2.3.2) for test collection for all documents and queries. Table 5.2 shows the mean average precision values. The results reported in Table 5.2 correspond to the case when the angle between the corresponding axes was taken to be equal to 60°, this value was meant to reflect the fact that the two terms were not independent from one another (the values 30° and 45° were also used in the experiments, but the results obtained were very similar to those reported in Table 5.2).

Table 5.2 Mean average precision obtained on standard test collections using the following retrieval methods: general basis based method (GB), entropy-based method (E), cardinality-probability method (KP). Baseline: VSM.

Test Coll.	**VSM**	**E**	**E over VSM**	**KP**	**KP over VSM**	**GB**
ADI	0.33	0.33	0%	0.35	+6%	0.33
MED	0.44	0.48	+9%	0.50	+14%	0.44
TIME	0.52	0.56	+8%	0.58	+12%	0.52
CRAN	0.18	0.20	+11%	0.20	+11%	0.18
			average=+7%		average=+11%	

Table 5.3 compares the results obtained to with those obtained using LSI with normalised term frequency [23].

Table 5.3 Mean average precision obtained on standard test collections using the following retrieval methods: general basis based method (GB), entropy-based method (E), cardinality-probability method (KP). Baseline: LSI.

Test Coll.	**LSI**	**E**	**E over LSI**	**KP**	**KP over LSI**	**GB**	**GB over LSI**
ADI	0.30	0.33	+10%	0.35	+17%	0.33	+10%
MED	0.48	0.48	0%	0.50	+4%	0.44	–9%
TIME	0.32	0.56	+75%	0.58	+81%	0.52	+62%
CRAN	0.25	0.20	–25%	0.20	–25%	0.18	–38%
			average=+15%		average=+19%		Average=+5%

The experimental results obtained from the in vitro measurements show that the Entropy- and Probability-based methods outperform both the classical vector space (in case of all test collections) and LSI methods (in three from the four collections; LSI appears to perform very well on the CRAN collection, probably due to its structure and content), and so, it can be concluded that the measure theoretic approach and definition of information retrieval (introduced in section 3.2) offers a good basis for proposing new and efficient retrieval methods.

6 Combined importance-based information retrieval [Thesis 3.a]

Web information retrieval has long been faced with trying to find the most relevant pages out of the many billions of Web pages in response to a user's query. Modern Web search engines typically use a mixture of retrieval methods partly based on classical methods (the computation of a similarity between the content of the query and pages) and partly on properties of the Web graph (methods based on the link structure of the Web).

In [72]-[73], methods based on alternative document models are proposed to enhance the PageRank method. Web pages are treated as individual HTML documents, all HTML files in the same directory are treated as a single document, and all HTML files with the same domain name are treated as a single document. The same holds for all pages belonging to one university domain name. The PageRank method is applied at directory, domain and university level. In [75], a new approach is proposed for the automatic building of a ranking function from a set of user examples. The user provides a set of requirements exemplified by some Web pages. Based on this user feedback, a customised ranking method is proposed as a generalisation of the PageRank method. In [34], the Web is conceived as a manifold. Similarities between query and pages are modelled by the geodesic distance which is transformed into Euclidean distance on the tangent plane of the manifold. In [81], a personalised retrieval method is proposed. The user's query is enhanced with information from the user's profile. PageRank values are computed based on a link matrix at domain level. Pages are then ranked using a linear combination of PageRank values and vector similarities (between query and page).

Link-based methods are usually combined with similarity methods to account for cases like high degree pages (e.g., index pages) without meaningful content, young pages with good content (but hardly receiving any links), etc..

In this Chapter, a new combined method is proposed for Web retrieval and ranking as a combination of content and link importance as well as similarity. The effectiveness of the method is evaluated (see Chapter 7.7) using the university domains *vein.hu* and *uni-pannon.hu* of the University of Pannonia, and compared to the commercial search engines MSN, Altavista and Yahoo!.

6.1 Content importance

The phrases "content" and "content-based" are used frequently nowadays in several areas with different meaning.

For example in the field of content industry *content* denotes any type of mass media, on-line games, mobile contents, e-books, internet broadcasting, e-music, etc..

Content-based Image Retrieval (CBIR) operates on different principle from text indexing. Primitive features characterizing image *content*, such as colour, texture, and shape, are computed for both stored and query images, and used to identify images most closely matching the query [38]. Semantic features such as the type of object present in the image are harder to extract, this is still an active research topic.

In everyday language, according to the Oxford English Dictionary, information is "communicating of the knowledge or 'news' of some fact or occurrence". Only a document of a certain news value is informative to the reader, thus information is a semantic property of a document, and should be distinguished from Shannon information [50].

In the following, the word *content* should be understood as semantic content, having the synonym of meaning, which is (roughly) called information in everyday language.

The content importance of a Web page can be defined and computed as follows: Given a set X and the real interval I = [0; 1] as defined in chapter 3.2. A fuzzy set A over X is defined as the mapping A: $X \rightarrow I$. Given a set $T = \{t_1, t_2,\ldots,t_i,\ldots,t_n\}$ of elements referred to as *terms*. According to equation 3.3 a *document* (Web page) may be interpreted as being the fuzzy set $d_j = \{(t_i, \varphi_j(t_i))|t_i \in T, i = 1,\ldots,n; \varphi_j(t_i) \in [0; 1]\}$ over T; $j = 1,\ldots,m$. The *cardinality* κ of document d_j is defined according to eq. 4.1 as follows:

$$\kappa(o_j) = \sum_{i=1}^{n} \varphi_j(t_i)$$

Let $TD_{n\times m} = (f_{ij})_{n\times m}$ denote the *term-by-document frequency matrix*, where f_{ij} denotes the number of occurrences of term t_i in document d_j. Let $W_{n\times m} = (w_{ij})_{n\times m}$ denote a *term-by-document weight matrix* obtained from the matrix TD (where $w_{ij} \in [0; 1]$ expresses how much term t_i "characterises" document d_j). The membership function $\varphi_j(t_i)$ may be taken to be the weight w_{ij}, i.e., $\varphi_j(t_i) = w_{ij}$. The fuzzy probability P_j of document d_j is defined by equation 5.3:

$$P_j = P(d_j) = \sum_{i=1}^{n} \varphi_j(t_i) \cdot p(t_i) \tag{6.1}$$

where $p(t_i)$ denotes a frequency-based probability of term t_i, which is calculated according to equation 5.5:

$$p(t_i) = \frac{\sum_{j=1}^{m} f_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij}}$$

If a Web page W_j is interpreted as being a fuzzy set d_j over a set T of terms, its fuzzy probability P_j may be taken as being proportional to (or an indication or

measure of) the chance that the page, as a fuzzy event, will occur. The fuzzy probability of a page is equal to zero if the page does not have any content (it is without meaning, the weights of all terms are zero). Thus, the fuzzy probability P_j is interpreted as a measure of its content importance.

6.2 Similarity measure

Given a query Q, and let $q_1,\ldots,q_i,\ldots,q_n$ denote query weights. In the well-known probabilistic model of retrieval, the conditional probability of a document D given a query Q, $P(D|Q)$, is estimated using Bayes' Theorem: $P(D|Q)P(Q) = P(Q|D)P(D)$. The right-hand side may be viewed, in a sense, as the probability of Q and D occurring simultaneously. As an analogy to the probabilistic model, a similarity function ρ_j (corresponding to $P(D|Q)$) between a document and a query may be proposed by fuzzyfication as follows: the document and the query as simultaneous fuzzy events are expressed using the cardinality of the algebraic fuzzy intersection between them, "normalised" by the fuzzy probability $P(Q)$ of the query, as given in eq. 5.6.

$$\rho_j = \frac{\kappa(Q \cap d_j)}{P(Q)} = \frac{\sum_{i=1}^{n} q_i \cdot w_{ij}}{\sum_{i=1}^{n} q_i \cdot p(t_i)}$$

6.3 Link importance

The PageRank method [14][52] is well known, still it is briefly recalled here in order to fix the ideas. In the PageRank method, a Web page's importance is determined by the importance of Web pages linking to it. The PageRank value R_i of a Web page W_i is given by the following equation:

$$R_i = \alpha(1 - d) + \beta d \sum_{W_j \in B_i} R_j C_j^{-1} + \gamma E \qquad (6.2)$$

where C_j denotes the number of outgoing links from page W_j, B_i denotes the set of pages W_j pointing to W_i, $d \in [0; 1]$ is a damping factor, E is the personalization matrix, whereas α, β, and γ are real parameters. The R_i PageRank values can be calculated using a simple iterative method which corresponds to the principal eigenvector of the normalized link matrix of the Web. Faster methods have been also introduced [33][39] to calculate the PageRank vector for large repositories of Web pages.

Let $G = (V, A)$ denote the directed graph of the Web, where the set $V = \{W_1, W_2,\ldots,W_N\}$ of vertices denotes the set of Web pages. The set A of arcs consists of the

directed links (given by URLs) between pages. Let $M = (m_{ij})_{N\times N}$ denote a square matrix attached to graph G such that:

$$m_{ij} = \begin{cases} \frac{1}{L_j} & \text{if there is a link from } W_j \text{ to } W_i, \\ 0 & \text{otherwise.} \end{cases}$$

The elements of matrix M may be interpreted in the following way: the entry m_{ij} is the probability with which page W_i follows page W_j during a walk on the Web (i.e., the probability with which, during a navigation on the Web, a surfer jumps from page W_j to page W_i). It may happen that a page W_j does not have any outgoing links (i.e., its outdegree is null). Such a page is referred to as a dangling page. The columns corresponding to dangling nodes in matrix M are replaced by columns containing all $1/N$.

$$m'_{ij} = \frac{1}{N}, \quad \text{where } i = 1,\ldots,N, \text{ page } W_j \text{ is a dangling page.}$$

Using matrix M', a new matrix, M'', is computed as follows:

$$M'' = \alpha M + (1 - \alpha)M', \quad 0 < \alpha < 1.$$

A typical value for α is $\alpha = 0.85$. Because the elements of matrix M'' are the coefficients of the right hand side of equation 6.2, this can be re-written in matrix form as $M'' \times \mathsf{R} = \mathsf{R}$, where R denotes the vector of PageRank values, i.e., $\mathsf{R} = (R_1,\ldots,R_i,\ldots,R_N)^{\mathrm{T}}$.In practice, the vector R can be computed using the following iteration: $\mathsf{R}_0 = (1/N,\ldots, 1/N)^{\mathrm{T}}$, $M'' \times \mathsf{R}_{i-1} = \mathsf{R}_i$, $i = 1,\ldots,K$, or until a pre-set degree of convergence occurs (i.e., $\|\mathsf{R}_{i-1} - \mathsf{R}_i\| < \varepsilon$). In [33] it is concluded that when calculating PageRank using the iterative method, the ordering which results after 25 iterations agrees very closely to the ordering induced by 100 iterations on what the top pages are.

6.4 Combined Importance-based Web retrieval and ranking method

The combined importance Ψ of a Web page W is defined as being a function F of its link importance L (e.g., defined by the PageRank value R of W) and of its content importance P: $\Psi = \Psi(L, P)$. An analytic form for Ψ can be obtained by accepting some reasonable assumptions (without restricting generality, it can be assumed that both P and L are normalised: $P, L \in [0; 1]$):

Assumption 1. It is straightforward to require that the combined importance of an isolated page without content be null:

$$\Psi(0, 0) = 0.$$

Assumption 2. If a Web page does not carry any meaning (practically it does not have any content), i.e., $P = 0$, its combined importance should vanish even if it is highly linked:

$$\Psi(L, 0) = 0.$$

Assumption 3. Further, from zero link importance ($L = 0$) need not necessarily follow a vanishing combined importance Ψ when the content importance does not vanish (this may be the case of a "young" Web page which is an isolated node of the Web graph but which may carry important meaning). Formally:

$$\Psi(0, P) \neq 0, P \neq 0.$$

Assumption 4. It also seems naturally to require that the combined importance of a page increase with its content importance P for the same link importance L; the same should hold also for L. Formally:

$$P_1 < P_2 \Rightarrow F(L, P_1) < F(L, P_2),$$

$$L_1 < L_2 \Rightarrow F(L_1, P) < F(L_2, P).$$

A possible and simple analytical form for Ψ that satisfies these assumptions is as follows:

$$\Psi(L, P) = \gamma PL + aP = P(\gamma L + a), \tag{6.3}$$

where the parameters a and γ are introduced to "maintain" a balance between content importance and link-based importance.

6.4.1 Web Retrieval and Ranking method

Based on content and link importance as well as on similarity, the following Web retrieval and ranking method is proposed:

1. Construct the Web graph G for the Web pages under focus, W_j, $j = 1,\ldots,N$.
2. Construct the link matrix M'' corresponding to graph G (Section 6.3).

3. Compute a link importance L_j for page W_j, $j = 1,\ldots,N$; e.g., using eq. (6.2).
4. Construct a set of terms $T = \{t_1,\ldots,t_i,\ldots,t_n\}$.
5. Construct the term-by-page frequency matrix: $(f_{ij})_{n\times N}$.
6. Compute the frequency-based probabilities $p(t_i)$ of terms using eq. (5.5); $i=1,\ldots,n$.
7. Define membership functions (weights) $\varphi_j(t_i)$, $j = 1,\ldots,N$; $i = 1,\ldots,n$ ($\varphi_j(t_i) = w_{ij}$) using some weighting scheme.
8. Calculate the content importance P_j of page W_j, $j = 1,\ldots,N$, using eq. (6.1).
9. Compute the combined importance Ψ_j for page W_j, $j = 1,\ldots,N$, using eq. (6.3) in section 6.4.
10. Enter query Q.
11. Construct, based on the similarity eq. (5.6), the set of pages that match the query: $\{W_j \mid \rho_j \neq 0,\ j = 1,\ldots, J\}$.
12. Compute an aggregated importance S_j for pages W_j, $j = 1,\ldots,J$, as follows: $S_j = \alpha\Psi_j + \beta\rho_j$.
13. Rank pages $W_1,\ldots,W_J$ descendingly on their aggregated importance $S_1,\ldots,S_J$ to obtain a hit list H.
14. Show the hit list H to the user.
15. (Iterate from step 10.)

7 WebCIR – a search engine using the combined importance-based method

In this section, a Web search engine called WebCIR is introduced. WebCIR implements the Combined Importance-based Web retrieval and ranking method given in 6.4. This system was implemented in order to make possible the evaluation of efficiency and relevance effectiveness of the suggested method.

7.1 Web Search Engine architecture

The literature dealing with search engine design is, unfortunately, scarce. In addition to research at universities and laboratories, many commercial companies have worked on search engines. However, these companies usually do not disclose their techniques and resulting performance because of obvious commercial reasons. So the exact methods and techniques employed by search engine companies are not known, making it harder to conclude what the best practices are nowadays. The pioneering work in search engine design has been done by the founders of Google in 1998 [14]. In their work, the foundations for a scalable hypertext search engine are laid out, which employs techniques such as link structure analysis, link-text indexing, etc. which are nowadays widely applied. In a more recent paper [5], an overview of Web search engine design is provided, with practical experience drawn from the Stanford WebBase project [69]. A generic search engine architecture is introduced, which is examined component by component. The most common design and implementation techniques for the components performing crawling, local Web site storage, indexing, and link analysis are presented.

Before the architecture of WebCIR is described, it is necessary to understand how a Web search engine is typically built. Based on [5], an overview of the most important components of the search engine is discussed in the following.

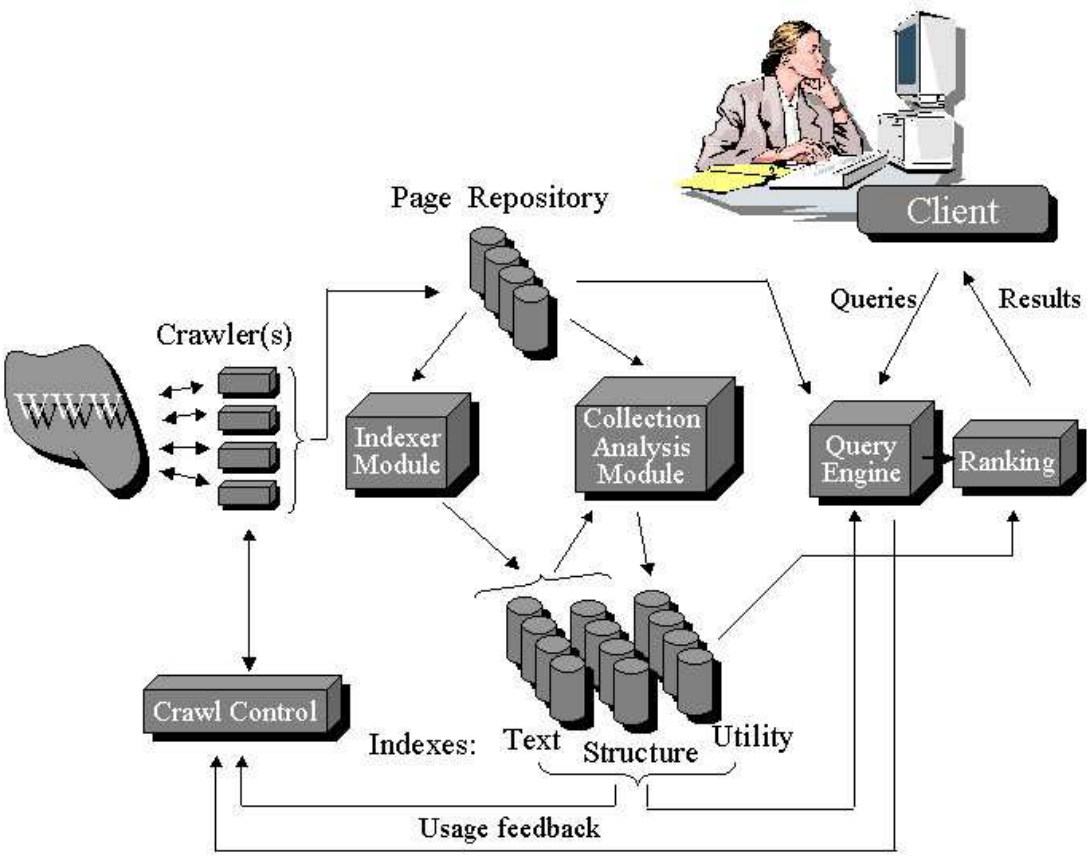


Figure 7.1 General search engine architecture [5].

Figure 7.1 shows the general structure of a Web search engine. Every engine gathers data from the Web using a *crawler module*. Crawlers are programs that visit Web pages (similarly to a user), downloading them and storing them in the page repository. Crawlers are given a starting set of URLs to visit, which they retrieve from the Web. They also extract links from the retrieved pages, which are then fed back into the crawl control module. The crawl control module determines what URLs should be visited next, and instructs the crawlers which links are to be retrieved. After the search engine has done at least one complete crawling cycle the indexer module can begin its work.

The *indexer* extracts all the words from each page and records, resulting in an inverted file structure (the text index shown in Figure 7.1) which can provide all the URLs that point to pages containing a specific word. Text indexing of the Web poses several challenges such as greatly varied, frequently changing content, billions of documents, while also calling for special kinds of indexes. For example, the collection analysis module, being responsible for the creation of the various indexes, might create a structure index too for storing the linkage information between Web pages. The collection analysis module might use the text and structure indexes to create the *utility index*. Utility indexes may provide auxiliary information about pages, such as the number of images they contain, or their “importance” calculated using a metric. Pages must be stored during a crawling and indexing run. The page *repository* in Figure 7.1 represents this collection, which is generally used for

temporarily storing Web pages. However, in order to serve results pages fast, search engines might cache the pages they visited beyond the time they are indexed.

The *query module* is responsible for collecting and fulfilling search requests from the users. It consults the indexes to find pages which match the user's query. Result sets are typically very large because of the vast amount of information available on the Web, and because users generally enter only one or two keywords as the query [9] [65].

The *ranking module* has the task to sort these results such that the top ones are likely those pages which the user is looking for. The query module has to deal with a difficult situation: the short queries entered by users [9] [65] match many documents, and traditional IR techniques used for similarity calculation are not able to filter sufficiently enough irrelevant pages from the result set. As such, a combination of methods from traditional IR and Web specific factors must be used. These Web specific factors might include linkage information between pages and on-page factors, such as text size or position.

7.2 WebCIR's architecture

After considering the advantages and disadvantages of different IR libraries (like Lucene[2], MG4J [47], Egothor [29] and Xapian [71]) we have chosen Nutch [4][21] to build the system. Nutch is an open source web search software suitable for testing new IR methods. It supports out-of-the-box infrastructure needed for Web-specifics, such as a crawler, a link-graph database, parsers for HTML and other document formats, and most importantly, it includes a *distributed computing framework* which makes it possible to process very large data sets with it.

Figure **7.2** shows the high-level system architecture of WebCIR. The parts below the dashed line have been implemented, while the tasks of the components above the line were carried out by Nutch. The individual components are described in the following.

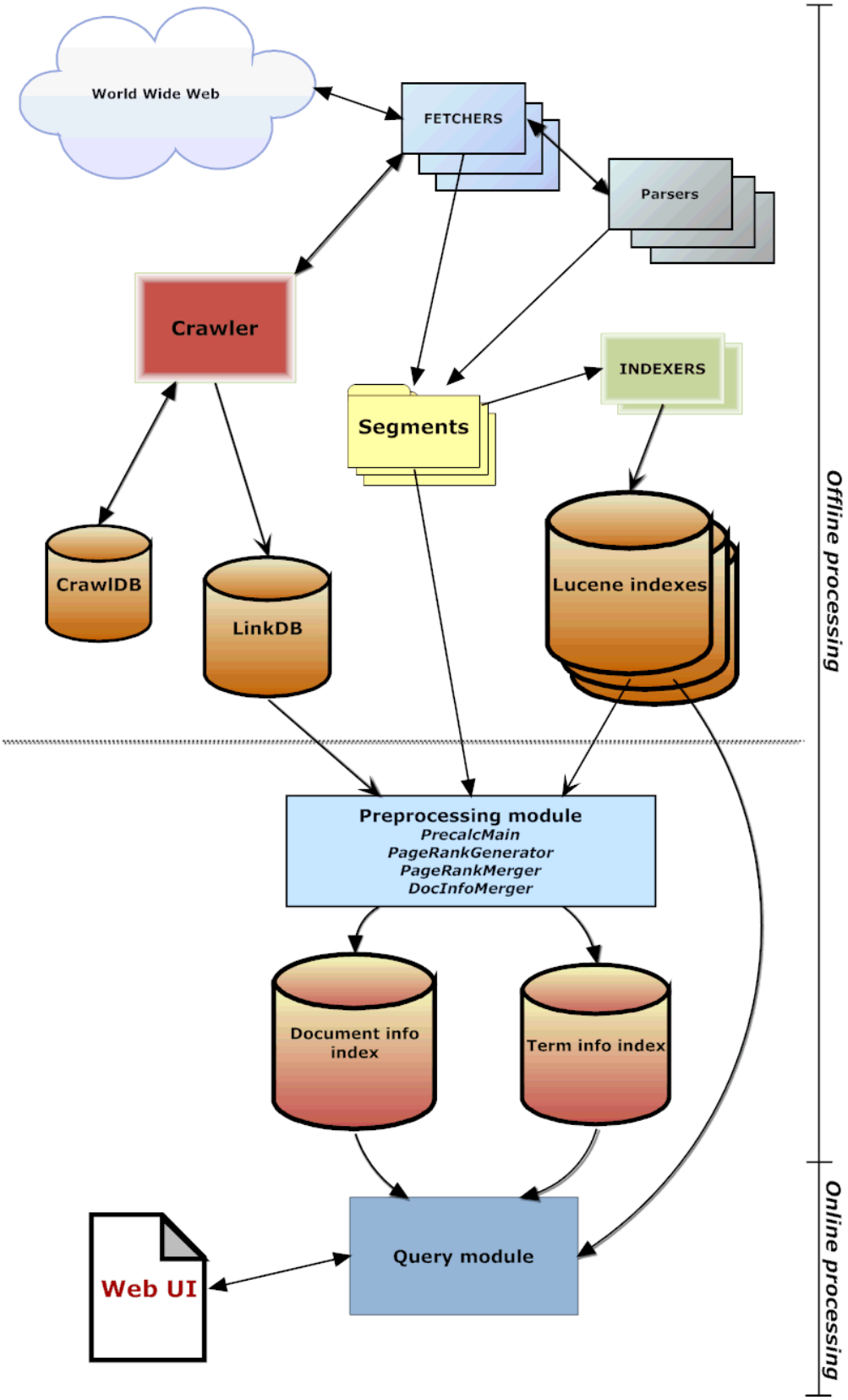


Figure 7.2 System architecture of WebCIR. The components above the dashed line are that of Nutch's (CrawlDB, LinkDB, crawler, fetchers, indexers, parsers), while the components below the line (the preprocessing- and query module) have been implemented.

7.2.1 CrawlDB

The CrawlDB (or page DB) contains information about the status of individual pages, such as page DB status (fetched, removed, injected, etc.), page signature, fetch status (successfully fetched, redirected, etc.), time info and other metadata.

7.2.2 Crawler Module

The crawling (downloading of Web pages) is controlled by Nutch's crawler. This tool takes some starting pages and the depth of recursion along with certain fine-tuning options and recursively downloads the pages using the fetchers, which actually do the downloading from the Internet.

If the page DB contains no links, it has to be bootstrapped with some known URLs to start crawling from. In this case these URLs are injected into the page DB. The generator checks the page DB and selects the best scoring pages to fetch, creating a new segment. Then the fetchers gather these selected pages and store them on disk (batch-mode crawling policy, see [5]). Right after the content have been downloaded by the protocol plugin (e.g., HTTP or FTP), it is parsed by a parser configured for the given document type. When fetching is finished, the page DB is updated with the contents of the new segment. Each fetching session generates a segment, which can be processed in the next phase.

Since an URL pointing to the same Web page can have multiple text forms, each of the above steps employ URL normalization to yield a canonical representation. For example, the URLs http://www.uni-pannon.hu:80/ and http://www.uni-pannon.hu/index.php represent the same Web page. It is important to ensure that URLs are handled in a consistent way, since nonnormalized URLs can have an adverse effect on the results of indexing or analysis.

Nutch's crawler supports both the "crawl & stop", and the "crawl & stop using threshold" crawler model, as described in [5]. For simplicity, WebCIR does only complete crawls: that is, the page repository is always replaced by the results of a subsequent crawl. A good summary of possible strategies for page selection and page refreshing when crawling are presented in [5] [16].

7.2.3 LinkDB

The *LinkDB* stores the inlinks of pages identified by their URL and anchor text associated with each link. This database is used for anchor text indexing and can be used to perform link-based analysis, as it is needed in the first three step of the Combined Importance-based Web retrieval and ranking method introduced in subsection 6.4.1:

1. Construct the Web graph G for the Web pages under focus, W_j, $j = 1,\ldots,N$.
2. Construct the link matrix M'' corresponding to graph G (Section 6.3).
3. Compute a link importance L_j for page W_j, $j = 1,\ldots,N$; e.g., using eq. (6.2).

…

7.2.4 Indexer Module

The data downloaded by the crawler is processed by the various plugins configured for Nutch, which by default populate a Lucene index from the content of the pages using the parsers. This data is then used by the *preprocessing module*.

In Lucene, documents consist of fields, which represent a named sequence of terms, while terms are simple strings usually representing a single word [3]. The same string in two different fields (e.g., in the fields named ”url” and ”title”) is considered a different term. This data structure makes it possible to store many properties of a document, such as title, URL, content, link-text from links pointing to it (anchors), etc., in a generalized way. Also, at query time the hits occurring in different fields can be weighted differently, like the terms matching the query in the title of the document can be given a higher weight than those found in the body of the document.

The Lucene index stores statistics about terms in order to ease term-based searching. Lucene’s index is an inverted index (because for a given term it can list the documents which contain it). In Lucene, indexes may be composed of several sub-indexes, or segments. Each segment is a fully independent index, which can be searched separately. Indexes are modified by either (i) creating new segments by adding new documents, or by (ii) merging existing segments.

The data stored here are needed by the *preprocessing module* to calculate the content importance P_j of pages (see section 6.1 and eq. (6.1)), in steps 4-8 of the Combined Importance-based Web retrieval and ranking method introduced in subsection 6.4.1:

4. Construct a set of terms $T = \{t_1,\ldots,t_i,\ldots,t_n\}$.
5. Construct the term-by-page frequency matrix: $(f_{ij})_{n\times N}$.

6. Compute the frequency-based probabilities $p(t_i)$ of terms using eq. (5.5); i=1,...,n.
7. Define membership functions (weights) $\varphi_j(t_i)$, j = 1,...,N; i = 1,...,n ($\varphi_j(t_i) = w_{ij}$) using some weighting scheme.
8. Calculate the content importance P_j of page W_j , j = 1,...,N, using eq. (6.1).

7.2.5 Preprocessing Module

Using the data stored in *linkDB* and Lucene indexes, the *preprocessing module* performs all calculations (needed in the steps of the Combined Importance-based Web retrieval and ranking method), which can be done offline. It calculates:

- the Kolmogoroff probabilities of terms (step 6.), the
- term membership functions (step 7.), the
- link importance L_j – particularly the PageRank values – of pages (step 3.), the
- content importance P_j – particularly the fuzzy probabilities – of pages (step 8.) and the
- combined importance Ψ_j values for pages (step 9.).

This module is implemented as a set of programs using the MapReduce programming model [22]. The results of these computations are stored in the *document*- and *term info indexes*.

7.2.6 Document Info Index

Using the terminology of [5], the document info index is a utility index. It keeps additional information about documents not stored in the Lucene index. This additional data structure is needed because of numerous reasons. Firstly, the Lucene index supports the addition of document fields only at indexing time, after the index is merged, no more field additions are possible. Secondly, in order to calculate the fuzzy probabilities of documents, all of the terms in the index have to be known beforehand because otherwise we cannot calculate their Kolmogoroff probabilities. Thus, the indexing phase needs to be completed before the calculation of term- and fuzzy probabilities. Thirdly, we would like to support the easy modification of the membership functions in the implemented Web retrieval and ranking method (see Section 6.4), so we have to keep the fuzzy probability data separate from the Lucene index.

The document info index is a fixed-width ISAM (index sequential access mode) index ordered by the identifier of the documents (docID). Each entry of it includes

the following information about a document: the docID, the fuzzy probability, the length normalization factors and the PageRank value. Most of this information is generated by MapReduce tasks so they are available in separate files, but instead, these pieces of data are collected into one data structure such that the information needed for ranking can be obtained with one disk seek at search time. This kind of optimization is needed to limit response time.

7.2.7 Term Info Index

The term info index is also a fixed-width ISAM index, ordered by termID. At present, it contains the Kolmogoroff probabilities of terms only, but it might be used for storing other kinds of term-level information later (e.g., stem of the term, statistics). The term info index is also a utility index.

7.2.8 Query Module

The query module takes the Lucene index and the document information data created by the *preprocessing module* to answer queries. It performs the following steps of the Combined Importance-based Web retrieval and ranking method introduced in subsection 6.4.1:

10. Enter query Q.
11. Construct, based on the similarity eq. (5.6), the set of pages that match the query: $\{W_j \mid \rho_j \neq 0,\ j = 1,\ldots, J\}$.
12. Compute an aggregated importance S_j for pages W_j ,$j = 1,\ldots,J$, as follows: $S_j = \alpha\Psi_j + \beta\rho_j$.
13. Rank pages $W_1,\ldots,W_J$ descendingly on their aggregated importance $S_1,\ldots,S_J$ to obtain a hit list H.
14. Show the hit list H to the user.

Note that the pre-set values of parameters α, β used by the computation of the aggregated importance of pages in step 12. is possible to change online for each query separately, e.g. for advanced users to fine tune the relation of combined importance and similarity.

7.3 Querying

This subsection starts with the presentation of WebCIR's query syntax and it is followed by the introduction of the used query expansion techniques.

7.3.1 Query syntax

WebCIR supports single- and multi-term queries. Single term queries match documents containing a given term (e.g., ”university” or ”science”), while multi-term queries match documents containing *all of* the terms. Thus, multi-term queries are implicit boolean queries having the AND operator between the query terms. The query is lowercased upon entering the system. Then, query expansion is applied (detailed in the next subsection) through lemmatization and automatic accenting. Query expansion can be disabled for a term by preceding it with a plus sign ”+” which can be used to restrict search for the term precisely as entered. Note that because of the possibility to turn off query expansion and to force searching for the query term exactly as given, stemming (detailed in the next subsection) cannot be performed at indexing time.

7.3.2 Query expansion

Query expansion means reformulating an original seed query to improve retrieval performance [28][54]. Common query expansion techniques used in Web search engines include the following:

- Finding synonyms of words,
- handle various morphological forms of words by stemming,
- fixing spelling errors and automatically searching for the corrected words,
- re-weighting the terms in the original query.

Query expansion is invoked to increase the quality of search results, because it is assumed that users do not always formulate the query using the ”best” terms: users can enter query terms which are not available in the lexicon, or which can only be found in a low number of documents. The goal of query expansion is to *increase recall, while not decreasing precision*. By finding more matching documents possibly having more matching terms, the ranking system has a chance to migrate documents with higher density up in the search results, leading to a higher quality of search results in spite of the increased recall.

In WebCIR two techniques were implemented to generate alternate terms for a query term: lemmatization and automatic accenting. The latter can be considered as correcting certain types of misspellings (like typing without accents or using the wrong accents).

Stemming and lemmatization

Stemming is the process of reducing words to their stem, root or base form. It is important to note that the stem need not be identical with the morphological root of the word, however, related words should be reduced to the same stem. In contrast, lemmatization is the process of reducing a word to its lemma or normalized, dictionary form. Consequently, the main difference between the two methods is that by lemmatization we always get a meaningful form of the word, while stemming

tends to truncate words which does not necessarily yield a meaningful dictionary word [54] [74]. The most well-known English stemming algorithm is that of Porter's [44]. It is a suffix stripping algorithm and thus does not rely on a lookup table of inflected forms or root form relations.

After empirical evaluations we concluded that results obtained by lemmatization were much more useful for query expansion. (Stemming produced too many improper non-dictionary forms of the words.) Because the majority of the texts in the "vein.hu" collection is Hungarian (see section 2.2), the technique worked out for query expansion using lemmatization has been especially targeted to the Hungarian language. For the lemmatization task the Hunspell [35] library was chosen, because it has been specifically built with the Hungarian language in mind (it can support any language through dictionaries as well). Hunspell uses a word dictionary and an affix dictionary to perform morphological analysis for lemmatization; it is able to determine what parts of speech a word might have in case of ambiguity, dissect compound words and do spell checking, among others.

Alternate query terms are generated by lemmatization as follows: we are given the lexicon, i.e. the set of index terms (in case of the "vein.hu" collection it contains about 670.000 terms, see chapter 2.2). The lemma and the lemma's part of speech for each query term are determined. Then the query is expanded with all the words from the lexicon, which start with the same lemma and have the same part of speech.

The lemmatization method was designed especially for the Hungarian language. The noise resulting from the presence of foreign language texts – as their proportion was below 1% in the "vein.hu" collection – was ignored.

Automatic accenting

Because it is often easier to type words without accents, people sometimes omit some or all accents from a word or mistype accents [54]. Because accents are extensively used in the Hungarian language, we used automatic accenting to remedy this problem and to improve recall and provide more relevant results. For each query term the *query module* generates every possible permutation of accents, and checks whether any of these variations can be found in the lexicon. If a variation is found in the lexicon, it is added to the query as an alternate term (identical to using logical OR operations) for the original term. For example, in case of the query "ösztöndíj" ("stipend") the following alternate terms were generated based on the "vein.hu" collection (see section 2.2): "ösztöndij", "osztondij". As such, the boolean query identical to what the original single-term query achieves is: ("ösztöndíj" OR "ösztöndij" OR "osztondij"). The permutation is done by replacing each vowel in a term with every possible accented or non-accented vowel. For example, for the vowel "o" the following accented vowels are tried out: "ó", "ö", "ő".

7.4 Searching

Search-related tasks are performed by the *Query module* (see Figure 7.2). This module is responsible for gathering matching documents and ranking them, be them local or distributed. Upon a search request the following operations are performed:

1. The Lucene index reader, the *document*- and *term info index* reader are opened.
2. The query is parsed.
3. Query expansion is applied and accented permutations and inflected forms of query terms are generated. The query is expanded with those which can be found in the lexicon.
4. Term information (Kolmogoroff probability) is read in for each query term.
5. For each term, list of documents is located in the inverted index using the pointer in the term dictionary.
6. Similarity is calculated between the matching documents in the aforementioned list and the query terms, then the combined rank is computed for these documents.
7. Hit list is presented.

Proximity Matching

As it was concluded in [65], users tend to avoid using the ”phrasifying” operator ”” to explicitly specify their intent to search for a phrase. Instead, they assume that the search engine will return documents which contain the query words in each other’s proximity. As such, the query can be considered an implicit phrase, and proximity matching can be used to exploit this user assumption.

Proximity matching techniques are based on that intuitive understanding, that in a relevant document, query terms appear relatively close to each other and not in completely unrelated parts of the document. Documents, which contain the query words nearer to each other, should get higher score during similarity calculation to the query. Exact phrase matches get the highest score. When a multi-term query is executed, the searcher looks for documents which contain all the query terms. Each time such a document has been found, the similarity value is calculated for it, taking into consideration partial hits too, i.e., when only a part of the original phrase can be reconstructed from consequent words occurring in a section of the document.

[70] gives a good summary of term-proximity measures. In [55], it was conducted that by applying a term-proximity scoring heuristic generally tends to improve retrieval effectiveness. In addition, in [15] it was concluded, that term proximity becomes more important as the size of the text collection increases, thus applying term-proximity scoring in the *Query module* of WebCIR was beyond question.

7.5 Ranking

The purpose of the ranking system is to order by relevance the thousands or millions of documents matching a query. Several heuristics are used for calculating these relevance values, based on which the documents are sorted and presented to the user.

However, this task is made difficult because of several reasons. Because of the already huge and exponential growth of Web collections the ranking algorithm needs to scale well. Another difficulty arises from the way search engines are used: most of the time Web users submit queries consisting of only one or two terms. Queries are very subjective, the set of documents considered relevant to a query change from person to person. Not to mention that the majority of users only look at the first search results page [9][65].

Term weighting

In order to get better ranking the length-normalized TF-IDF (term frequency/inverse document frequency) weighting was used instead of simple length-normalized term frequency (TF) weighting [61] for the membership function (see Section 6.1). The inverse document frequency (idf) is a measure of the general importance of a term t_i, obtained by dividing the number of all documents ($m = |D|$) by the number of documents containing the term, and then taking the logarithm of that quotient, that is:

$$\mathrm{idf}_{ij} = \log \frac{m}{F_i}$$

where F_i denotes the number of documents containing term t_i. Using the above formula, the TF-IDF weighting of a term t_i can be obtained as a simple product $\mathrm{f}_{ij} \times \mathrm{idf}_{ij}$. Employing TF-IDF weighting as the membership function is a good choice because a high weight value is only reached by a high term frequency in the document under focus and a low document frequency of the term in the whole document set; as such, common terms tend to be filtered out by lower weights.

Another enhancement is the use of term weighting to distinguish between the same terms occurring in different parts or fields of a document, because, for example, a term occurring in the URL of the document can be considered more important than occurring in the body. The weight for each field type is applied when calculating fuzzy probabilities and these weights are also used for calculating the similarity values during search. Four types of fields are considered: terms occurring in the *title*, in the *body*, or in the *URL* of the document, and terms occurring in links pointing to the document (*anchor text*).

Summarizing the above, when considering the term t_i occurring on Web page W_j , the membership function (calculated by the p*reprocessing module* in step 7. of the Combined Importance-based Web retrieval and ranking method) used by WebCIR is defined as follows:

$$\varphi_j(t_i) = w_{t_i} \frac{f_{ij} \times \left(\log \frac{m}{F_i} \right)}{\sqrt{\sum_{k=1}^{n} \left(f_{kj} \times \left(\log \frac{m}{F_k} \right) \right)^2}}$$

where F_k denotes the total number of documents containing term t_k, w_{t_i} the term weight, and m the total number of pages considered. The weighting scheme used can be seen in Table 7.1. The weights were tuned empirically, taking Lucene's default field weight values as a starting point.

Table 7.1 Term weighting scheme.

Field type	w_{t_i}
content	1.0
title	1.35
URL	4.0
anchor text	1.7

7.6 User interface

A search engine's user experience is made up of two parts: the user interface and the functionality [63]. Since both are equally important, we used the best of the layouts used by popular public search engines and the well-tried colours for presenting results pages. As such, the user interface of WebCIR gives the impression of being easy to learn to use as well as actually easy to use, yet it offers the functionality expected from a Web search engine.

The starting page of WebCIR is shown in Figure 7.3. By clicking on the "Advanced Options" link, the parameter a of the combined importance function (see section 6.4) and the parameters α, β of the aggregated importance function (see step 12 in subsection 6.4.1) can be also set there besides entering a query.

Figure 7.3 Starting page of WebCIR. Here the query can be entered, and the parameters of the importance functions may be tweaked.

After submitting the query by clicking on the ‘search’ button, retrieval is performed and the search results page is shown to searchers (Figure 7.4).

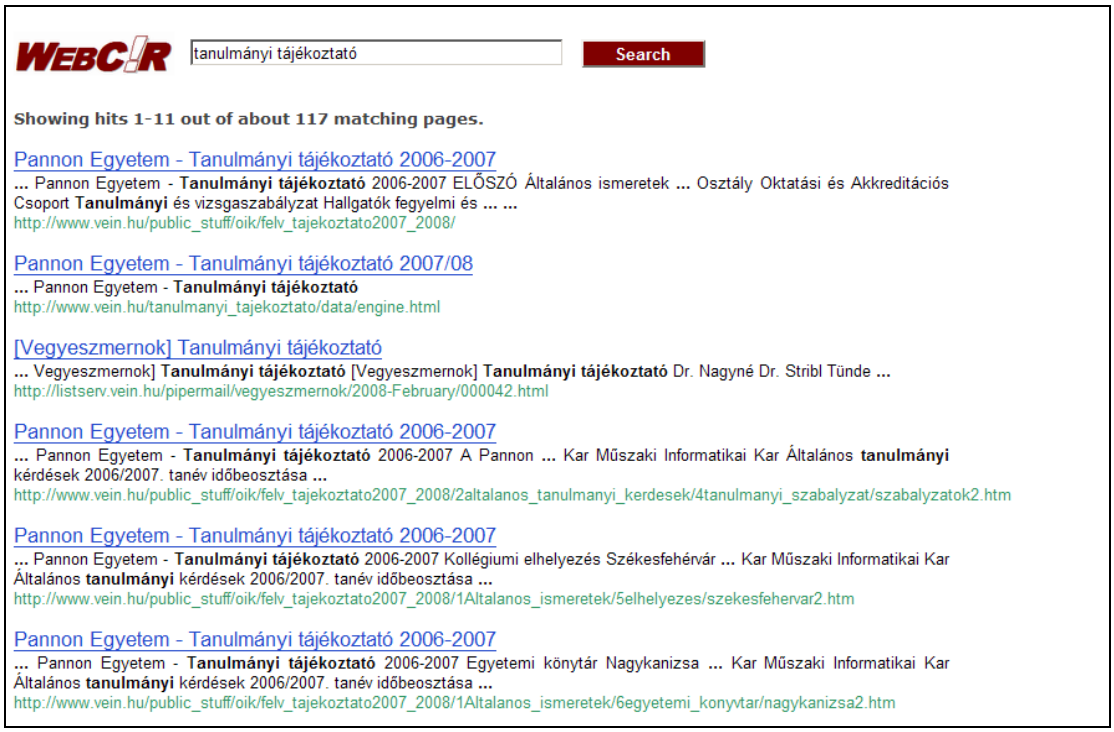


Figure 7.4 Sample search results for query: ”tanulmányi tájékoztató”.

Figure 7.4 shows an example result page of WebCIR for the query ”tanulmányi tájékoztató” (“study guide”). On one search results page 10 hits can be found, which are, of course, ordered by decreasing – aggregated importance values meaning – relevance. For each hit, the title, the summary and the URL is shown. Summarization is done by Nutch’s summarizer plugin. It works by retokenizing a text string generated from the entire content of the original document, and extracting a minimal set of excerpts containing five words of context on each side of a hit [21]. These excerpts are then ranked based on their length and the number of hits they contain. Finally, the summary is truncated to a maximum of twenty words.

The estimated total number of hits is displayed on the top of the search results page. At the bottom of the page, the user can jump to individual results pages using the page number links, or, if available, to the next or the previous results page. Of course, a new query can also be submitted on this page, without returning to the previous one.

For implementing the user interface of WebCIR Sun’s Java Servlet Pages (JSP) technology was used. The JSP pages were deployed on an Apache Tomcat servlet container during testing.

7.7 WebCIR's evaluation [Thesis 3.b]

It is well-known that the *in vivo* measurement of relevance effectiveness of a Web search engine poses several problems. For example, recall cannot be calculated. Neither can precision, in many cases (due to the high numbers of hits returned, which are practically impossible to assess).

The effectiveness of WebCIR was evaluated using four measurement methods and it was compared to the effectiveness of the commercial search engines MSN [48], Altavista [1], and Yahoo! [87].

7.7.1 Evaluation methodology

For evaluation, two types of methods were used: (i) MLS and DCG methods (see subsections 2.3.3 and 2.3.4), which require user's assessments and (ii) RC and RP methods (see subsections 2.3.5 and 2.3.6), which do not require relevance judgements. Test queries were chosen and fed into the search engines: WebCIR, MSN, Altavista and Yahoo!. WebCIR was running on the "vein.hu" collection (see section 2.2), the others' results were restricted to the vein.hu domain. For the MLS and DCG evaluation methods real users were needed, to whom a special evaluation software was also implemented in favour of the easy and objective relevance judgements. The relevance judgements together with the hit lists given for the test queries were stored in a database. Test queries, WebCIR's settings and the user assessment are introduced more detailed in the followings.

7.7.1.1 Test queries

The test queries were selected from the official query log of the University of Pannonia: the 32 most frequent ones as of 4th of April 2008 were used in the measurements.

7.7.1.2 Parameters of the Combined Importance-based Web retrieval and ranking method

In WebCIR, for the computation of combined importance the parameter values were fine-tuned (during manual experiments) and set as follows: a = 0.25, α = 100,000, β = 0.5, γ = 100,000. For PageRank calculation the following (generally used) parameter values were used: α =1/numDocs (numDocs denotes to the total number of Web pages), d = 0.85, β = 1, γ = 0. The calculation was done using the iterative method with 20 iterations.

7.7.1.3 Hit list assessment

The hit lists were assessed by real users (students of the Faculty of Information Technology on the 10th of April, 2008) using a measurement software specially developed for the purpose of these evaluations.

Figure 7.5 Screenshot of the measurement software developed for assessing hit lists.

A query was chosen randomly from the test queries by the software, and was sent to the first search engine in the background. Then, the users were shown the hit list (with the query as well), they did not know from which search engine it came, they just assessed every hit. A hit list (of one of the search engines) for the query "kollégium" can be seen on Figure 7.5, by clicking on a hit the corresponding Webpage opens in a new window. Hits could be judged to be *relevant* or *non relevant* with the help of a combo box on the right side of the hits. After marking all the hits, the next engine's hit list could be evaluated by clicking on the "tovább" button on the bottom.

7.7.2 Results of the evaluation

In this subsection, the numerical results of the used evaluation methods are presented.

7.7.2.1 Results obtained with the MLS Method

This subsection presents the numerical results obtained with the MLS method (see subsection 2.3.3). Appendix C shows the values of *P*10 for all queries and search engines. Summing it up the rank of search engines is as follows:

Search Engine	**MLS measure**	
WebCIR	0.66	100%
Altavista	0.6	91%
Yahoo!	0.6	91%
MSN	0.58	88%

7.7.2.2 Results obtained with the DCG Method

This subsection presents the numerical results obtained with the DCG method (see subsection 2.3.4). Appendix D shows the values of the DCG measure for all queries and search engines. Summing it up the rank of search engines is as follows:

Search Engine	**DCG Measure**	
Yahoo!	52.31	100%
Altavista	49.9	95%
MSN	49.76	95%
WebCIR	48.47	93%

7.7.2.3 Results obtained with the RC Method

This subsection presents the numerical results obtained with the RC method (see subsection 2.3.5). Appendix E shows the values of the RP measure for all queries and search engines. Summing it up the rank of search engines is as follows:

Search Engine	**RC Measure**	
Altavista	151	100%
Yahoo!	130	86%
MSN	94	62%
WebCIR	73	48%

7.7.2.4 Results obtained with the RP Method

This subsection presents the numerical results obtained with the RP method (see subsection 2.3.6). Appendix F shows the values of the RP measure for all queries and search engines. Summing it up the rank of search engines is as follows:

Search Engine	**RP Measure**	
Altavista	0.49	100%
Yahoo!	0.41	84%
MSN	0.25	51%
WebCIR	0.21	43%

7.7.3 Discussion

The effectiveness of WebCIR was estimated using four measurement methods (MLS, DCG, RC, RP) by performing *in vivo* experiments involving real users. Exactly the same experiments were performed also on the commercial search engines Yahoo!, Altavista, and MSN in order to compare the results. (However, Google has a significant interest in the search engine market; unfortunately, we could not use it in the comparison due to technical (and financial) reasons.)

The MLS and DCG measurement methods are based on user assessments. According to the MLS measure, Altavista and Yahoo! have equal effectiveness (0.6), MSN seems to be very slightly below this value (0.58), while WebCIR performed best (having the smallest standard deviation), quite above this value: 0.66. According to the DCG measure, Yahoo! performs slightly better than all other search engines, while these perform practically equally well.

The RC and RP methods are not using user assessments, they allow to mutually compare the hit lists of several search engines, and thus obtain a ranking of their effectiveness (relative to each other).

What catches the eye first is that Altavista and Yahoo! Perform very "close" to each other. This is not surprising because, as it is well-known, Altavista's search results are powered by Yahoo!'s technology. Thus, these two search engines re-inforce each other's RC and RP measures. WebCIR's and MSN's effectiveness are roughly at the same level, MSN slightly outperforming WebCIR.

In order to diminish the influence of the mutual bias between Altavista and Yahoo!, the RC and RP measures were re-computed after Altavista's hit lists were excluded. The thus obtained RC and RP values are shown in Appendix G and Appendix H. The new rank of search engines is now as follows:

Search Engine	**RC Measure**
MSN	64 100%
WebCIR	55 86%
Yahoo!	41 64%

Search Engine	**RP Measure**
MSN	0.23 100%
WebCIR	0.2 87%
Yahoo!	0.13 57%

It can be seen that in this case the situation changed: Yahoo! shifted down to the last position, with a considerable loss, while MSN and WebCIR perform quite close to each other. However, the standard deviation of RP for MSN is larger than the standard deviation of RP for WebCIR.

To sum up the measurement results, we may conclude that

- In the experiments involving real users, WebCIR seems to perform best, followed immediately by MSN, while Altavista and Yahoo! are third but perform equally well.

- In the experiments comparing search engines to each other (no users involved, bias excluded), MSN seems to perform best (note that with the largest standard deviation), followed immediately by WebCIR, while Yahoo! is in third position (with a considerable difference).

Thus, one may say that WebCIR offers a very competitive retrieval and ranking technology (for domain retrieval).

7.7.4 Future work

This subsection presents some issues, which have been considered but – because they were not needed at the present state of the research – not actualized yet. These are described in the following and a possible solution is proposed for each.

7.7.4.1 Normalizing the values of the Combined importance function

Currently, content importance (fuzzy probabilities) and link importance (PageRank) values are not normalized. The parameters a and γ of the combined importance function Ψ (see section 6.4) have been tuned especially for the "vein.hu" collection (see section 2.2), which was used for testing. Without normalization the parameters have to be tuned for every document collection, because of the possibly differing orders of magnitude of these values from collection to collection. For example, in case of a repository consisting of pages from a single host only, the calculated PageRank values for the pages might be smaller by orders of magnitude from those calculated based on a larger link graph.

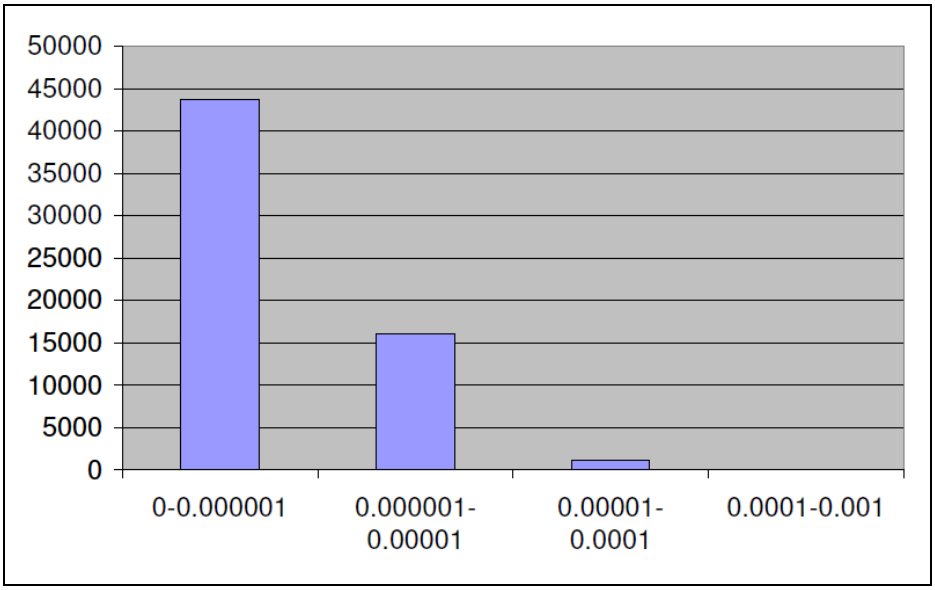


Figure 7.6 The distribution of fuzzy probabilities.
The fuzzy probability values were split into intervals represented on the horizontal axis; cardinalities of the intervals are represented on the vertical axis.

The distribution of fuzzy probabilities can be seen in Figure 7.6. The higher fuzzy probability values are very sparse, while the majority of documents have fuzzy

probability values which fall into the lowest interval. One way to solve the normalization problem for fuzzy probability values could be to segment them and assign a weight to each of these segments. An obvious assignment of weights is shown in Table 7.2. Also, the weights could be interpolated between the weight of the previous interval and the weight corresponding to the interval into which the fuzzy probability falls, as to get a continuous weight function.

Table 7.2 Example weights for the segmented fuzzy probabilities.

Fuzzy probability	Weight
[0; 0.0000001[	0
[0.0000001; 0.000001[	0.25
[0.000001; 0.00001 [	0.5
[0.00001; 0.0001[	0.75
[0.0001; 1]	1

The distribution of PageRank values follows a power law distribution [77]. To normalize these values, the same segmenting and weighting approach described above could be applied.

7.7.4.2 Spamming

A great challenge in ranking is spam. Since the inclusion of a Web site in the first results page increases traffic to it, page authors, especially those with commercial interests, deliberately try to deceive search engines to improve their ranking. Several techniques exist to mislead search engines, such as content manipulation (e.g., placing popular terms on the page in a non-disturbing way), link manipulation (e.g., employing link farms or doorway pages), or cloaking, and the combination of these. In fact, the ”war” between spammers and Web search service providers is ever-continuing: as soon as a spamming technique is defended by the target search service, spammers come up with new methods or variants of their methods to circumvent the counteract. A more detailed description of the Web search engine spam problem can be found in [45].

Searching inside a given domain usually does not need the use of any anti-spamming techniques. For example, in a university domain, as it was WebCIR tested on, pages with commercial interest are not typical, as well as the competition for the first places. At the present state WebCIR is vulnerable to spam since there are no heuristics applied to combat it. For example, WebCIR is prone to content manipulation because of the way fuzzy probabilities are calculated. Recall from Section 6.1 that fuzzy probabilities are calculated from the Kolmogoroff probabilities

of the terms appearing on the page and their frequency. As such, repeating the same term multiple times on a page increases the content importance, thence the combined importance of the page in question. Moreover, because of the increased frequency of the intentionally repeated term, its frequency-based Kolmogoroff probability will increase too, thus increasing the page's fuzzy probability even further.

One possible solution for this problem could be the modification of the membership function to increase linearly first then taper off quickly. However, determining the inflection point for this function raises some questions such as whether the threshold number should be uniform for all documents or whether it should depend on the whole document collection or on just the document itself. Answering these questions and countering Web spam is left for future work.

7.7.4.3 Other directions of experimentation

Further work may include testing other ways to compute term probabilities, together with their effect on fuzzy probabilities.

Other analytical forms for the global importance function should be experimented with. Also, different values for the parameters a, α, β, and γ may be tested and fine-tuned. Larger scale experiments would be helpful in assessing how the retrieval method proposed scales up with very high number of Web pages.

Other ways to calculate the link-based importance, e.g., SALSA, HITS, I^2R [25] as well as the content-based importance may also be tried out.

8 Conclusions

In this chapter, the main contributions and the proposed theses – both in English and Hungarian – are summarized in the next sections, and then the publications related to this dissertation are listed.

8.1 Theses

1. *General formal framework for information retrieval*

Taking the definitions given for Information Retrieval (IR), they do not give different interpretations for IR, rather they all define IR the same way. In my dissertation a general formal framework for IR has been given.

(a) The concept of retrieval has been defined based on the mathematical measure theory. Then, documents (and queries) were particularised using fuzzy set theory [Chapter 3.2]. As a result, the retrieval function was conceived as the cardinality of the intersection of two fuzzy sets [Lemma 4.1].

(b) It has been shown that using the concepts of this general framework the generalised vector space retrieval model, the latent semantic indexing retrieval model and the classical vector space retrieval model gain a correct formal mathematical formulation and background that is consistent with practice [Chapter 4].

2. Entropy- and probability-based retrieval methods

The measure theoretic view (proposed in (1.a)) makes it possible to consistently formulate new retrieval methods. By taking fuzzy entropy and fuzzy probability as measures, new retrieval methods have been given, which are both consistent with their mathematical background.

(a) Entropy-based retrieval method has been given by taking fuzzy entropy as measure in the retrieval function. [Chapter 5.1]

(b) Probability-based retrieval method has been given by taking fuzzy probability as measure in the retrieval function. [Chapter 5.2]

(c) Effectiveness of the methods has been measured; experimental results using standard test collections have been reported. The experiments showed that enhancements from +5% to +19% can be obtained in average (over VSM and LSI), which indicates that the approach introduced in (1.a) offers a good basis for proposing new and better retrieval methods. [Chapter 5.3]

3. Combined importance-based method for the retrieval and ranking of Web pages

Owe to the special properties of the World Wide Web, modern Web search engines typically use a mixture of retrieval methods partly based on classical methods and partly on properties of the Web graph.

(a) Using the concepts introduced in (1.a) and (2.b) a new method has been proposed for the retrieval and ranking of Web pages based on content importance, link importance, and topical similarity. The method is implemented in a search engine called WebCIR. [Chapters 6.4 and 7]

(b) Four measurement methods have been used, involving human assessors as well, to evaluate the effectiveness of WebCIR, which was compared to the effectiveness of Altavista, Yahoo!, and MSN. The results show that the Combined importance-based method is a very competitive Web page retrieval and ranking method. [Chapter 7.7]

8.2 Tézisek magyar nyelven

Az értekezés új tudományos eredményei az alábbiakban foglalhatók össze:

1. Információ-visszakereső módszerek egységes keretrendszere

Az információ-visszakeresésre adott definíciókat megvizsgálva észrevehetjük, hogy azok nem különböző interpretációi az IR-nek, hanem nagyon hasonlóak. Ezt alapul véve megadtam az információ-visszakeresés egységes formális keretrendszerét.

(a) Megadtam a visszakeresés elvének matematikai mértékelméleten alapuló definícióját. A dokumentumokat (és a keresőkérdéseket) a fuzzy halmazelmélet segítségével határoztam meg [Chapter 3.2]. Majd a visszakeresést, mint két fuzzy halmaz metszetének számosságával definiált függvényt tekintettem. [Lemma 4.1].

(b) Megmutattam, hogy az így megadott egységes keretrendszerben, az általánosított vektortér-modellt, a rejtett szemantikus indexelést (LSI) és a klasszikus vektortér-modellt újradefiniálva azok helyes matematikai formalizmust kapnak, melyek konzisztensek a gyakorlattal [Chapter 4].

2. Entrópia- és valószínűség alapú visszakereső módszerek

Az új mértékelméleti megközelítés lehetővé teszi további, új és az elmélettel konzisztens visszakereső módszerek megadását. A fuzzy entrópiát és a fuzzy valószínűséget alapul véve új visszakereső módszereket adtam meg, melyek konzisztensek a matematikai hátterükkel.

(a) A visszakereső függvényben a fuzzy entrópiát véve mértéknek megadtam az Entrópia-alapú visszakereső módszert [Chapter 5.1].

(b) A visszakereső függvényben a fuzzy valószínűséget véve mértéknek megadtam a Valószínűség-alapú visszakereső módszert [Chapter 5.2].

(c) A módszerek relevancia-hatékonyságát sztenderd teszt-kollekciókon mértem. A gyakorlati eredmények alapján a VSM és LSI módszerekéhez képest átlagosan 5% és 19% közti hatékonyság növekedést tapasztaltam, mely azt mutatja, hogy a mértékelméleti megközelítésen alapuló egységes

keretrendszer jó alapja lehet új és hatékony visszakereső módszerek kifejlesztésének [Chapter 5.3].

3. Kombinált fontosság-alapú Webes információ-visszakereső módszer

A World Wide Web speciális tulajdonságai miatt a modern webes keresők jellemzően olyan visszakereső módszereket használnak, melyek részben klasszikus visszakereső módszereken, részben pedig a Webgráf speciális tulajdonságain alapulnak.

(a) Az (1.a) és (2.b) tézispontokban megfogalmazott keretrendszert és valószínűség alapú módszert használva új webes információ-visszakereső módszert adtam meg, mely tartalmi- és link alapú fontosságon, valamint hasonlóságon alapul. A módszert a WebCIR nevű keresőmotorban implementáltam [Chapters 6.4 and 7].

(b) A WebCIR kereső relevancia-hatékonyságának kiértékelésére 4 különböző módszert alkalmaztam, majd az eredményeket az Altavista, Yahoo!, és MSN keresők eredményeivel hasonlítottam össze. A kísérletek eredményei azt jelzik, hogy a Kombinált fontosság-alapú Webes visszakereső módszer versenyképes alternatívát jelenthet [Chapter 7.7].

8.3 Publications

8.3.1 Publications directly related to the thesis

[P1] DOMINICH, S., **KIEZER, T.**, ERDÉLYI, M. (2008). WebCIR: Web ranking and search engine using combined method. *Studies on information and knowledge processes 13.* Infota, pp.: 53-74, ISBN **[thesis 2, 3]**

[P2] DOMINICH, S., **KIEZER, T.** (2007). A Measure Theoretic Approach to Information Retrieval. *Journal of the American Society for Information Science and Technology*. John Wiley & Sons, Vol. 58, no 8, pp.: 1108-1122, ISSN 1532-2882, IF=1.773. **[thesis 1, 2]**

8.3.2 Other publications relevant to the thesis

[P3] DOMINICH, S., GÓTH, J., **KIEZER, T.** (2006). Web-based Neuroradiological Information Retrieval System using three methods to satisfy different user's aspect. *Computerized Medical Imaging and Graphics*, ISSN 0895-6111, pp: 263-272, IF=1.090.

[P4] DOMINICH, S., **KIEZER, T.** (2005). Hatványtörvény, „kis világ” és magyar nyelv. *Alkalmazott Nyelvtudomány*, pp: 5-25, ISSN 1587-1061.

[P5] DOMINICH, S., GÓTH, J., **KIEZER, T.** (2005). NeuRadIR: A Web-Based NeuroRadiological Information Retrieval System. *ERCIM News*, vol. 61., pp:52-53, ISSN 0926-4981.

[P6] DOMINICH, S., GÓTH, J., M. HORVÁTH, **KIEZER, T.** (2005). 'Beauty' of the World Wide Web – Cause, Goal, or Principle. *Lecture Notes in Computer Science*, Springer Verlag, Volume 3408/2005, pp:67-80, ISSN 0302-9743, IF=0.515.

[P7] DOMINICH, S., GÓTH, J., **KIEZER, T.**, SZLÁVIK, Z. (2004). Entropy-based interpretation of Retrieval Status Value-based Retrieval, and its application to the computation of term and query discrimination value. *Journal of the American Society for Information Science and Technology*. John Wiley & Sons, Vol. 55, no 7, pp: 613-627, ISSN 1532-2882, IF=1.773.

Bibliography

[1] Altavista Search Engine, http://www.altavista.com/

[2] Apache Lucene: an open source information retrieval library. `http://lucene.apache.org/`.

[3] Apache Lucene: Index File Formats. `http://lucene.apache.org/java/docs/fileformats.html`.

[4] Apache Nutch: an open source web-search software. `http://lucene.apache.org/nutch/`.

[5] Arasu, A., Cho, J., Garcia-Molina, H., Paepcke, A., and Raghavan, S., (2001) Searching the Web. *ACM Transactions on Internet Technology*, 1(1) pp.:2–43.

[6] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley Longman Publishing Co. Inc.

[7] Baeza-Yates. R. (2003). Information retrieval in the Web: Beyond current search engines. *International Journal of Approximate Reasoning*, vol. 34, pp: 97-104.

[8] Belew, R.K. (2000). *Finding Out About*. Cambridge University Press.

[9] Bernard J. Jansen, Spink, A., and Saracevic. T., (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management*, 36(2) pp: 207–227.

[10] Berry, M.W., Drmac, Z., Jessup, E.R. (1999). Matrices, Vector Spaces, and Information Retrieval. *SIAM Review*, vol. 41, no. 2, pp: 335-362.

[11] Berry, W.M., Browne, M. (1999). *Understanding Search Engines*. SIAM, Philadelphia.

[12] Bollmann-Sdorra, P. and Raghavan, V.V., (1993). On the Delusiveness of Adopting a Common Space for Modelling Information Retrieval Objects: Are Queries Documents? *Journal of the American Society for Information Science*. 44(10), pp: 579–587

[13] Borlund, P., (2003). The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*. **8**(3), pp:1–66.

[14] Brin, S., and Page, L. (1998). The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Proceedings of the 7th World Wide Web Conference*, Brisbane, Australia, 14-18 April, pp: 107-117.

[15] Büttcher, S., Clarke, C.L.A., Lushman, B.: Term proximity scoring for Ad-Hoc retrieval on very large text collections. In: SIGIR' 06, New York, NY, ACM Press (2006) 621–622

[16] Cho, J., García-Molina, H., and Page, L., (1998). Efficient crawling through URL ordering. *Computer Networks and ISDN Systems*, 30(1–7) pp.:161–172.

[17] Chu, H. and Rosenthal, M., (1996). Search Engines for the World Wide Web: A Comparative Study and Evaluation Methodology. *Proceedings of the American Society for Information Science Annual Meeting*, 33, pp: 127–135.

[18] Clark, S.J., Willett, P., (1997). Estimating the recall performance of Web search engines. In: *Aslib Proceedings*. **49**(7), pp.: 184–189

[19] Cooper, W. S., (1968). Expected search length: a single measure of retrieval effectiveness based on the weak ordering action of retrieval systems. *American Documentation*. 19, pp: 30–41.

[20] Cristianini, N., Shawe-Taylor, J., Lodhi, H. (2002). Latent semantic kernels. *Journal of Intelligent Information Systems*, vol. 18, no. 2-3, pp:127–152.

[21] Cutting D. Sitaker K. Khare, R. and A. Rifkin. (2004). Nutch: A flexible and scalable open-source web search engine. *Technical report*.

[22] Dean, J., and Ghemawat, S., (2004) MapReduce: Simplified data processing on large clusters. *Proceedings of the 6th Symposium on Operating Systems Design and Implementation*, San Francisco, CA.

[23] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, vol. 41, pp: 391-407.

[24] Dominich, S. (2003). Connectionist Interaction Information Retrieval. *Information Processing and Management*. Elsevier, vol 39, no 2, pp: 167-194

[25] Dominich, S. (2008). *The Modern Algebra of Information Retrieval*. Springer-Verlag, Berlin Heidelberg.

[26] Doob, J.L. (1994). *Measure Theory*. Springer Verlag.

[27] Dubin, D. (2004). The most influential paper Gerard Salton never wrote. Library Trends, Spring. http://www.findarticles.com/p/articles/mi_m1387/is_4_52

[28] E. N. Efthimiadis., (1996). Query expansion. In M. E.Williams, *Annual Review of Information Science and Technology*, 31, pp.:121–187.

[29] Egothor Home :: egothor search engine. `http://www.egothor.org/.`

[30] Feynman, R.P., Leighton, R.B., Sands, M. (1964). *The Feynman Lectures on Physics*. Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, USA.

[31] Folland, G.B. (1984). *Real analysis: modern techniques and their applications*. John Wiley and Sons, New York.

[32] Gordon, M., and Pathak, P., (1999). Finding information on the World Wide Web: the retrieval effectiveness of search engines. *Information Processing and Management*, 35, pp: 141–180.

[33] Haveliwala, T., (1999). Efficient Computation of PageRank. *Technical Report* 1999-31.

[34] Hua, H., Boqin, F. (2005). Web retrieval algorithm based on differential manifold. *Journal of Xi'an Jiaotong University*, 39(2), pp: 130–134.

[35] Hunspell: open source spell checking, stemming, morphological analysis and generation.
`http://hunspell.sourceforge.net/`.

[36] J. Shawe-Taylor, N. Cristianini. (2004). Kernel Methods for Pattern Analysis. *Cambridge University Press*, New York, NY, USA.

[37] Jarvelin, K., Kekalainen, J. (2002). Cumulated Gain-based Evaluation of IR Techniques. *ACM TOIS*, vol. 20, no. 20, pp: 422-446

[38] Kato, T. (1992) Database architecture for content-based image retrieval. *Image Storage and Retrieval Systems*, Proc SPIE 1662, pp: 112-123.

[39] Klein, D., Manning, C., Haveliwala, T., Kamvar, S., and Golub, G., (2003). Computing pagerank using power extrapolation. *Stanford University Technical Report*.

[40] Kolmogoroff, A. (1950). *Foundation of Probability*. New York.

[41] Lánczos, K. (1970). *Space through the Ages*. Academic Press, Inc., London.

[42] Leighton, H. V., & Srivastava, J. (1999). First twenty precision among world wide web search services (Search Engines). *Journal of the American Society for Information Science*, 50(10), pp: 870-881.

[43] Luhn, H.P., (1966). Keyword-in-Context Index for Technical Literature (KWIC Index). In: *Readings in Automatic Language Processing*, ed by Hays, D. D. (American Elsevier Publishing Company, Inc.), pp.: 159–167

[44] M. F. Porter, (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

[45] M. Henzinger, R. Motwani, and C. Silverstein., (2002). Challenges in web search engines. *ACM SIGIR Forum*, 36(2), pp.: 11–22.

[46] Meadow, C.T., Boyce, B.R. and Kraft, D.H. (1999). *Text Information Retrieval Systems*. Second edition, Academic Press, San Diego, CA.

[47] MG4J: Managing Gigabytes for Java.
`http://mg4j.dsi.unimi.it`.

[48] Microsoft Live Search, http://search.msn.com/

[49] Nallapati, R. (2004). Discriminative Models for Information *Retrieval. Proceedings of the ACM SIGIR International Conference on Research and Development in Information Retrieval*. Sheffield, United Kingdom, pp:

[50] Nöth, W. (1995) *Handbook of Semiotics*. Indiana University Press, Bloomington.

[51] Oppenheim, C., Morris, A., and McKnight, C., (2000). The evaluation of WWW search engines. *Journal of Documentation*, **56**(2), pp: 190–211.

[52] Page, L., Brin, S., Motwani, R., and Winograd, T., (1998). The PageRank Citation Ranking: Bringing Order to the Web. *Technical report, Stanford Digital Library Technologies Project*.

[53] Ponte, J.M., Croft, W.B. (1998). A Language Modelling Approach to Information Retrieval. *Proceedings of the ACM SIGIR International Conference on the Development and Research in Information Retrieval*, Melbourne, Australia, pp: 275-281.

[54] Raghavan, P., Manning, C., D. and Schütze., H., (2008). *Introduction to Information Retrieval*, pp.:19–47,177–194. Cambridge University Press.

[55] Rasolofo, Y., Savoy, J.: Term proximity scoring for keyword-based retrieval systems. In: ECIR. (2003) 207–218

[56] Robertson, S.E., Sparck-Jones, K. (1977). Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, vol. 27.

[57] Rudin, W., (1966). *Real and Complex Analysis*. McGraw Hill, New York.

[58] S. E. Robertson. The probabilistic ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[59] Salton, G. (1965). Automatic Phrase Matching. In: Hays. D. G. (ed.) *Readings in Automatic Language Processing*, American Elsevier, New York, 1966. pp:169-188.

[60] Salton, G. (1986). Another look at automatic text-retrieval systems. *Communications of the ACM*, vol. 29, no. 7, pp: 648-656.

[61] Salton, G., and Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, vol. 24, no. 5, pp: 513-523.

[62] Salton, G., Wong, A., and Yang, C.S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM*, vol. 18, no. 11, pp: 613-620.

[63] Search User Interface and User Experience - SearchTools Report. `http://www. searchtools.com/info/user-interface.html.`

[64] Shafi, S. M., Rather, R. A. (2005). Precision and Recall of Five Search Engines for Retrieval of Scholarly Information in the Field of Biotechnology. *Webology*, 2 (2), Article 12.

[65] Silverstein, C., Henzinger, M., Marais, H., and Moricz, M., (1998). Analysis of a very large AltaVista query log. *Technical Report* 1998-014, COMPAQ Systems Research Center, Palo Alto, Ca, USA .

[66] Simmonds, J.G. (1982). *A Brief on Tensor Analysis*. Springer Verlag.

[67] Siolas, G., d'Alch´e Buc, F. (2000). Support Vectors Machines Based on a Semantic Kernel for Text Categorization. *In Proceedings of the International Joint Conference on Neural Networks*, vol. 5, pp: 5205.

[68] Sparck Jones, K. and van Rijsbergen, C.J. (1976). Progress in Documentation. *Journal of Documentation*, 32(1), pp: 59-75.

[69] Stanford WebBase project.
`http://www-diglib.stanford.edu/~testbed/doc2/WebBase/.`

[70] Tao, T., Zhai, C., (2007). An Exploration of Proximity Measures in Information Retrieval. SIGIR'07, July 23–27, Amsterdam, The Netherlands.

[71] The Xapian Project.
`http://xapian.org/.`

[72] Thelwall, M., Vaughan, L. (2004). New versions of PageRank employing alternative Web document models. *ASLIB Proceedings*, 56(1), pp: 24-33.

[73] Thelwall, M., Wilkinson, D. (2003). Three target document range metrics for university Web sites. Journal of the American Society for *Information Science and Technology*, vol. 54, no. 6, pp: 489-496.

[74] Tikk, D., (2007). *Szövegbányászat*, pp.: 25–62. TypoTeX, Budapest.

[75] Tsoi, C.A., Scarselli, F. (2006). Computing customised Page Ranks. *ACM Transactions on Internet Technology*, 6(4), pp: 381–414.

[76] Turtle, H., *Inference Networks for Document Retrieval*. Ph.D. thesis, Department of Computer Science, University of Massachusetts, Amherst, MA 01003, 1990. Available as COINS Technical Report 90-92.

[77] Upstill, T., Craswell, N., and Hawking, D., (2003). Predicting fame and fortune: PageRank or indegree? *In Proceedings of the Australasian Document Computing Symposium, ADCS2003*, Canberra, Australia, pp.: 31–40.

[78] Van Rijsbergen, C.J. (1979). *Information Retrieval*. Butterworth, London.

[79] Van Rijsbergen, C.J. (2004). *The Geometry of IR*. Cambridge University Press, Cambridge, U.K.

[80] Von Neumann, J. (1927). Mathematische Grundlagen der Quantumphysik. *Göttinger Nachrichten*, Math.-Phys., Klasse, pp: 1-46.

[81] Wang, H., Guo, Y., Feng, B. (2006). Optimising personalized retrieval system based on Web ranking. *Lecture Notes in Computer Science, LNCS 3967*, Springer Verlag, pp: 629–640.

[82] Wang, Z., Klir, G.J. (1991). *Fuzzy Measure Theory*. Plenum Press, New York.

[83] Won, K.M. (1996). On Fuzzy s-open Maps. *Kangweon-Kyungki Mathematical Journal*, 4, no. 2, pp: 135-140.

[84] Wong, S.K.M., Raghavan, V.V. (1984). Vector space model of information retrieval – a re-evaluation. *Proceedings of the 7th ACM SIGIR International Conference on Research and Development in Information Retrieval*. Kings College, Cambridge, England, pp: 167-185.

[85] Wong, S.K.M., Ziarko, W., Wong, P.C.N. (1985). Generalized Vector Space Model in Information Retrieval. *Proceedings of the 8th ACM SIGIR International Conference on Research and Development in Information Retrieval*. New York, ACM Press, pp: 18-25.

[86] Wu, S., Crestani, F. (2003). Methods for ranking information retrieval systems without relevance judgements. *ACM SAC '03*, March 9-12, Melbourne, Florida, USA, pp: 811-816.

[87] Yahoo! Search Engine, http://search.yahoo.com/

[88] Zimmerman, H.-J. (1996). *Fuzzy set theory – and its applications*. Kluwer Academic Publishers, Norwell-Dordrecht.

Appendix A.

A.1. ADI collection

The ADI collection contains 82 homogeneous English articles from computing journals with 35 queries. The test collection contains the following files:

- **adi.all**: documents
- **adi.que**: queries,
- **adi.rel**: relevance list,
- **adi.bln**: list of Boolean queries.

The documents file (adi.all) contain text articles and generally some structured fields in addition, as for example:

- **.I**: serial number of the document
- **.T**: title of the document
- **.A**: author/authors of the document
- **.W**: text of the document.

When some of these additional pieces of information about the article are not known, the corresponding field is missing from the document. Figure A.1 shows the 50th document of the collection.

.I 50
.T
english-like systems of mathematical logic for content retrieval .
.A
H. G. BOHNERT
T. J. WATSON
.W
an english-like system of mathematical logic is a formally defined set of sentences whose vocabulary and grammar resemble english, with an algorithm which translates any sentence of the set into a notation for mathematical logic . objectives, accomplishments, and problems in the construction of such languages in project logos are discussed .

Figure A.1 Structure of the 50th document in the ADI test collection. The tag .I denotes the serial number, .T the title, .A the author(s) and .W the content/text of the document.

The structure of the queries is similar to the documents, except they do not contain (*.A*) author/authors and (*.T*) title fields. Figure A.2 shows an excerption from the adi.que file; *.I3* is a mixed query including question and declarative sentence too, *.I16* is a simple query including only a question and *.I34* is a simple query including a declarative sentence.

```
.I 3
.W
What is information science?  Give definitions where possible.
…
.I 16
.W
What systems incorporate multiprogramming or remote stations in information
retrieval?  What will be the extent of their use in the future?
…
.I 34
.W
Methods of coding used in computerized index systems.
```

Figure A.2 Excerption from the adi.que file in the ADI test collection.
The tag .I denotes the serial number and .W the text of the query.

Figure A.3 shows a fraction of the relevance assessments (adi.rel file) illustrating which answers are relevant to which queries, for example: documents *.I3*, *.I43*, *.I45*, *.I60* are relevant to the query *.I3*.

```
…
3    3
3    43
3    45
3    60
…
16   18
16   55
…
34   1
34   15
34   39
…
```

Figure A.3 Excerption of the relevance assessments file,
showing the relevant documents for queries .I3, .I16, and .I34.
For example, for query .I16 there are two relevant documents: .I18 and .I55.

A.2. MED collection

MED is a collection of 1033 medical abstracts from the Medlars collection with 30 queries. It consist of the following files:

- **med.all**: documents,
- **med.que**: queries,
- **med.rel**: relevance list

The documents file (med.all) contains document texts and serial numbers.

- **.I**: serial number of the document
- **.W**: text of the document.

Figure A.4 shows the 8th document as an example of the collection.

.I 8
.W
essential fatty acids and acids with trans-configuration in the subcutaneous and visceral fat of the newborn .
we made an investigation of the subcutaneous and visceral fat in the newborn . we estimated the contents of linolic and linolenic acid and of acids with trans-configuration spectrophotometrically . we were able to show the penetration of these acids through the placental barrier . the essential fatty acid contents of fat in the newborn is low . in immature ones about 7-14 g, there is a rising trend.

Figure A.4 Structure of the 8th document in the MED test collection.
The tag .I denotes the serial number and .W the content/text of the document.

The structure of the queries and relevance list is similar to those of the ADI collection which were introduced in Appendix A.1., in Figure A.2 and Figure A.3.

A.3. TIME collection

Time is a collection of 423 articles from magazine Time including 83 queries and their relevance list. It consist of the following files:

- **Time.all**: documents,
- **Time.que**: queries,
- **Time.rel**: relevance list,
- **Time.stp**: stop list containing words which occur in documents but should be ignored.

The documents file (Time.all) contains articles from the year 1963 along with some extra information which includes a serial number, the exact date and a page number when and where the article had been appeared. Figure A.5 shows a fraction of the 106th document as an example of the collection.

**TEXT 106 02/22/63 PAGE 030*

[…] *HUNGARY'S FARSANG WAS TRADITIONALLY A TIME TO BLOW OFF STEAM BEFORE THE ONSET OF LENT'S RIGORS . IT WAS BANNED BY HUNGARY'S RED RULERS . BUT NOW, WITH THEIR TOLERANCE, FARSANG (PRONOUNCED FORSHONG), IS MAKING A COMEBACK* **[…]** *HUNGARY'S FESTIVAL PALES BY COMPARISON WITH THE OLD DAYS, WHEN MAGYAR ARISTOCRATS WOULD SPIT ON A 100-FORINT NOTE (WORTH ABOUT $12.50), SLAP IT ON A GYPSY'S FOREHEAD, AND DEMAND PASSIONATE VIOLINPLAYING UNTIL THE SPITTLE DRIED AND THE NOTE FELL OFF .* **[…]** *DON'T LET ALL THIS GAIETY FOOL YOU, " A BUDAPEST WRITER WARNED AN AMERICAN VISITOR AFTER A FARSANG BALL . " THE YOUNG PEOPLE ARE GAY BECAUSE THEY ARE YOUNG . THE OLD PEOPLE THEY ARE GAY BECAUSE THEY DON'T KNOW WHAT COMES TOMORROW . "*

Figure A.5 Structure of an article in the TIME test collection.
Every document is preceded by an information line starting with a "*" character, which is followed by the serial number, the exact date and a page number informing when and where the article appeared in the magazine.

An excerption of the Time.que file containing the queries can be seen on Figure A.6. Every query is identified by a serial number. The relevance list has the structure showed on Figure A.7; each line starts with the query's serial number, and is followed by serial numbers of relevant documents.

```
*FIND    17
 WITHDRAWAL BY THE SULTANATE OF BRUNEI FROM THE PROPOSED FEDERATION OF MALAYSIA .

*FIND    18
 RUSSIA'S REFUSAL TO CONTRIBUTE FUNDS FOR THE SUPPORT OF UNITED NATIONS PEACEKEEPING FORCES .
```

Figure A.6 Excerption of Time.que file containing the queries for TIME test collection. Queries are preceded by information lines starting with "*FIND" strings, followed by the serial number of the given query. The next line contains the query itself.

```
…
17 303 358
18 356
19 99 100 195 267 344
…
```

Figure A.7 Excerption of Time.rel file. Each line starts with the query's serial number and is followed by serial numbers of relevant documents. As it can be seen, there are two relevant documents for query #17: documents #303 and #358

The TIME stoplist (Figure A.8) contains stop words which should be ignored during processing the documents and queries. These are usually high frequent words bearing very low or no meaning (in general or specially in the given context)[78], like:

```
A
ABOUT
ABOVE
ACROSS
…
BACK
BAD
BE
…
TIME
…
```

Figure A.8 Excerption of the TIME stoplist

A.4. CRAN collection

CRAN is a collection of 1400 aerodynamics abstracts from the Cranfield collection including 225 queries with relevance assessments. It consists of the following files:

- **cran.all**: documents
- **cran.qry**: queries,
- **cran.rel**: relevance list.

The documents file (cran.all) contain text articles and some structured fields in addition, as for example:

- **.I**: serial number of the document
- **.T**: title of the document
- **.A**: author/authors of the document
- **.B**: source of the document, e.g.: journal name, year, page number
- **.W**: text of the document.

Figure A.9 shows the 36th document as an example of the collection.

.I 36
.T
supersonic flow around blunt bodies .
.A
serbin,h.
.B
j. ae. scs. 25, 1958, 58.
.W
supersonic flow around blunt bodies .the newtonian theory of impact has been shown to be useful for pressure calculations on the forward facing part of bodies moving at high speed . it is now a familiar practice to use this information to calculate nonviscous velocities at the wall and then to estimate rates of heat transfer . this procedure is perhaps open to question,. heat-transfer rates depend on velocity gradients which are not given by the newtonian analysis . nor can one obtain information on boundary-layer stability or all the body stability derivatives . it seems, therefore, inevitable that, as design proceeds with these hypersonic missiles, there will be a greater need for more accurate aerodynamic theories either to predict what will happen in unfamiliar flight conditions or to effect an extrapolation from a known test result to the design condition .

Figure A.9 Structure of the 36th document in the CRAN test collection. The tag .I denotes the serial number, .T the title, .A the author(s), .B the source and .W the content/text of the document.

The structure of the queries and relevance list is similar to those of the ADI collection which were introduced in Appendix A.1.

Appendix B.

MathCAD programs, which were used for evaluating GB, E and KP methods.

1. Define document term weighting schemes DTW(p,M,n,m):

p = 1, fully weighted (tfc), i.e., [term_freq x log(n/Fi)]; length normalised TFxIDF ;
p = 2, standard normalised frequency, (txc); length normalised term frequency;
p = 3, classical term frequency inverse document frequency, TFxIDF (tfx);
p = 4, best weighted probabilistic, (nxx); 0.5+0.5*tf/max(tf);
p = 5, entropy weighting; log(tf)*(1-log(SUM(pij*log(pij)/log(ndocs)));
p = 6, BM25
M = raw term frequency matrix having n raws and m columns

$$
\begin{aligned}
&DTW(p,M,n,m) := \\
&\quad \text{for } i \in 1..n \\
&\quad\quad \text{for } j \in 1..m \\
&\quad\quad\quad W_{i,j} \leftarrow 0 \\
&\quad \text{if } p = 1 \\
&\quad\quad \text{for } i \in 1..n \\
&\quad\quad\quad F_i \leftarrow 0 \\
&\quad\quad\quad \text{for } j \in 1..m \\
&\quad\quad\quad\quad F_i \leftarrow F_i + 1 \quad \text{if } M_{i,j} \neq 0 \\
&\quad\quad \text{for } j \in 1..m \\
&\quad\quad\quad L \leftarrow 0 \\
&\quad\quad\quad \text{for } i \in 1..n \\
&\quad\quad\quad\quad L \leftarrow L + \left(M_{i,j} \cdot \log\left(\frac{n}{F_i} \right) \right)^2 \quad \text{if } F_i \cdot M_{i,j} \neq 0 \\
&\quad\quad\quad \text{for } i \in 1..n \\
&\quad\quad\quad\quad W_{i,j} \leftarrow M_{i,j} \cdot \frac{\log\left(\frac{n}{F_i} \right)}{\sqrt{L}} \quad \text{if } L \cdot F_i \cdot M_{i,j} \neq 0 \\
&\quad W
\end{aligned}
$$

2. Define query term weighting schemes QTW(p,M,n,m):

p = 1, fully weighted (nfx);
p = 2, classical term frequency IDF, (tfx);
p = 3, binary term indpendence (bpx);
M = raw term frequency matrix having n raws and m columns

$$
\begin{aligned}
&QTW(p,M,n,m) := \\
&\quad \text{for } i \in 1..n \\
&\quad\quad \text{for } j \in 1..m \\
&\quad\quad\quad W_{i,j} \leftarrow 0 \\
&\quad \text{for } j \in 1..m \qquad\qquad \text{if } p = 2 \\
&\quad\quad L \leftarrow 0 \\
&\quad\quad \text{for } i \in 1..n \\
&\quad\quad\quad L \leftarrow L + \left(M_{i,j} \right)^2 \quad \text{if } M_{i,j} \neq 0 \\
&\quad\quad \text{for } i \in 1..n \\
&\quad\quad\quad W_{i,j} \leftarrow \frac{M_{i,j}}{\sqrt{L}} \quad \text{if } L \cdot M_{i,j} \neq 0 \\
&\quad W
\end{aligned}
$$

3. Read in term-document raw term frequency (i.e., number of occurrence) matrix TD(nxm):

$TD := READPRN("med\_all\_matrix\_stemmed\_stopped.txt")$

No of terms: $N := rows(TD)$ $N =$ ▮ $i := 1..N$

No of docs: $M := cols(TD)$ $M =$ ▮ $j := 1..M$

Probability distribution of terms: total number of occurrences of all terms: $s_j := \sum TD^{\langle j \rangle}$ $S := \sum s$

probability of each term: $TT := TD^T$ $p_i := \frac{\sum TT^{\langle i \rangle}}{S}$

4. Apply weighted scheme for documents: $D := QTW(2, TD, N, M)$

General basis : t(1108)="cell", t(5637)="patient". "cell" will be oblique to "patient" at 60 degrees.

$a := 1108$ $b := 5637$ $alpha := \frac{\pi}{3}$ $s := \frac{1}{\sin(alpha)}$ $s =$ ▮ $c := \frac{-\cos(alpha)}{\sin(alpha)}$ $c =$ ▮

Document vectors D in general basis: $DG := D$ $DG_{a,j} := s \cdot D_{a,j}$ $DG_{b,j} := c \cdot D_{a,j} + D_{b,j}$

5. Read in term-query frequency matrix TQ(nxm):

$TQ := READPRN("med\_qry\_matrix\_stemmed\_stopped.txt")$

Number of queries: $MQ := cols(TQ)$ $MQ =$ ▮ $jq := 1..MQ$

6. Apply weighted scheme for queries: $Q := QTW(2, TQ, N, MQ)$

Query vectors Q in general basis: $QG := Q$ $QG_{a,jq} := s \cdot Q_{a,jq}$ $QG_{b,jq} := c \cdot Q_{a,jq} + Q_{b,jq}$

7. Compute similarity inner product SIM(MxMQ) between all documents and all queries:

$SIM := D^T \cdot Q$ $rows(SIM) =$ ▮ $cols(SIM) =$ ▮

Similarity in general basis: $SIMGB := DG^T \cdot QG$

Cardinality and probability: $P_{j,jq} := \frac{Q^{\langle jq \rangle} \cdot D^{\langle j \rangle}}{Q^{\langle jq \rangle} \cdot p}$

8. Rank order SIM descendingly:

```
rank_desc(X) := | for i ∈ 1..M
                |   RR_{i,1} ← i
                |   RR_{i,2} ← X_i
                | for i ∈ 1..M − 1
                |   for j ∈ i + 1..M
                |     if RR_{i,2} < RR_{j,2}
                |       v1 ← RR_{i,1}
                |       v2 ← RR_{i,2}
                |       RR_{i,2} ← RR_{j,2}
                |       RR_{i,1} ← RR_{j,1}
                |       RR_{j,2} ← v2
                |       RR_{j,1} ← v1
                | RR
```

$RSIM_{jq} := rank\_desc\left(SIM^{\langle jq \rangle}\right)$

$RSIMGB_{jq} := rank\_desc\left(SIMGB^{\langle jq \rangle}\right)$

$RSIMKP_{jq} := rank\_desc\left(P^{\langle jq \rangle}\right)$

9. Read in relevance matrix R: R := READPRN("MED.REL")

RN := rows(R) RN = ▮ ir := 1..RN RM := cols(R) RM = ▮ jr := 1..RM

```
REL := | k ← 1
       | r_k ← R_{1,1}
       | for ir ∈ 2..RN
       |   if R_{ir,1} ≠ r_k
       |     k ← k + 1
       |     r_k ← R_{ir,1}
       | r
```

10. Compute precision-recall for similarity-based retrieval: FS := 1033

```
PR_SIM :=  for q ∈ 1..MQ
             no_of_rel_doc ← 0
             for ir ∈ 1..RN
               if REL_q = R_(ir,1)
                 no_of_rel_doc ← no_of_rel_doc + 1
                 r_(no_of_rel_doc) ← R_(ir,2)
             no_ret_sim ← 0
             for i ∈ 1..FS
               for j ∈ 1..no_of_rel_doc
                 if ⌈RSIM(REL_q)⌉_(i,1) = r_j
                   no_ret_sim ← no_ret_sim + 1
                   pr ← round(no_ret_sim·10 / no_of_rel_doc)
                   P_(REL_q,pr+1) ← no_ret_sim / i
           for q ∈ 1..MQ
             for pr ∈ 10,9..1
               P_(REL_q,pr) ← P_(REL_q,pr+1) if P_(REL_q,pr) = 0
           for pr ∈ 1..11
             p_pr ← mean(P^⟨pr⟩)
           p
```

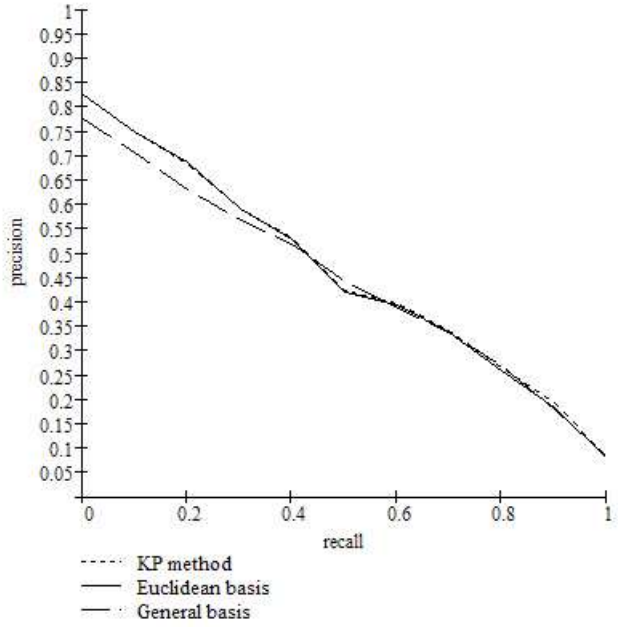
```
PR_SIMGB:= for q ∈ 1..MQ
             no_of_rel_doc ← 0
             for ir ∈ 1..RN
               if REL_q = R_(ir,1)
                 no_of_rel_doc ← no_of_rel_doc + 1
                 r_(no_of_rel_doc) ← R_(ir,2)
             no_ret_sim ← 0
             for i ∈ 1..FS
               for j ∈ 1..no_of_rel_doc
                 if ⌈RSIMGB(REL_q)⌉_(i,1) = r_j
                   no_ret_sim ← no_ret_sim + 1
                   pr ← round(no_ret_sim·10 / no_of_rel_doc)
                   P_(REL_q,pr+1) ← no_ret_sim / i
           for q ∈ 1..MQ
             for pr ∈ 10,9..1
               P_(REL_q,pr) ← P_(REL_q,pr+1) if P_(REL_q,pr) = 0
           for pr ∈ 1..11
             p_pr ← mean(P^⟨pr⟩)
           p
```

```
PR_SIMKP:= for q ∈ 1..MQ
             no_of_rel_doc ← 0
             for ir ∈ 1..RN
               if REL_q = R_(ir,1)
                 no_of_rel_doc ← no_of_rel_doc + 1
                 r_(no_of_rel_doc) ← R_(ir,2)
             no_ret_sim ← 0
             for i ∈ 1..FS
               for j ∈ 1..no_of_rel_doc
                 if ⌈RSIMKP(REL_q)⌉_(i,1) = r_j
                   no_ret_sim ← no_ret_sim + 1
                   pr ← round(no_ret_sim·10 / no_of_rel_doc)
                   P_(REL_q,pr+1) ← no_ret_sim / i
           for q ∈ 1..MQ
             for pr ∈ 10,9..1
               P_(REL_q,pr) ← P_(REL_q,pr+1) if P_(REL_q,pr) = 0
           for pr ∈ 1..11
             p_pr ← mean(P^⟨pr⟩)
           p
```

11. Plot precision-recall graph:

mean(PR_SIM) = mean(PR_SIMGB) = mean(PR_SIMKP) =

r := 1.. 11



12. Compute amount of information INF(MxMQ) associated to documents D and queries Q:

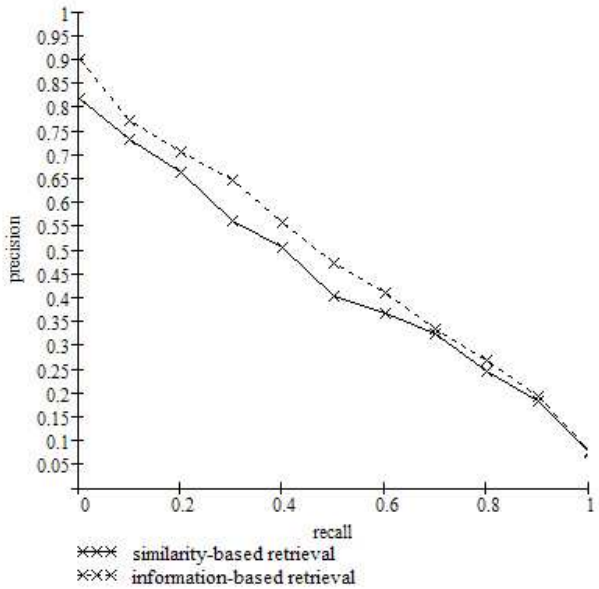
INF :=
for j ∈ 1.. M
  for jq ∈ 1.. MQ
    p ← 1
    for i ∈ 1.. N
      if $Q_{i,jq} \wedge D_{i,j}$
        $v \leftarrow D_{i,j} \cdot Q_{i,jq}$
        $p \leftarrow p \cdot v^{v}$
    $inf_{j,jq} \leftarrow \ln(p)$
–inf

13. Rank order INF descendingly: $RINF_{jq} := rank\_desc\left(INF^{\langle jq \rangle}\right)$

14. Compute precision-recall values PR_INF for information-based retrieval

```
PR_INF := | for q ∈ 1..MQ
          |   no_of_rel_doc ← 0
          |   for ir ∈ 1..RN
          |     if REL_q = R_(ir,1)
          |       no_of_rel_doc ← no_of_rel_doc + 1
          |       r_(no_of_rel_doc) ← R_(ir,2)
          |   no_ret_sim ← 0
          |   for i ∈ 1..FS
          |     for j ∈ 1..no_of_rel_doc
          |       if [RINF(REL_q)]_(i,1) = r_j
          |         no_ret_sim ← no_ret_sim + 1
          |         pr ← round(no_ret_sim·10 / no_of_rel_doc)
          |         P_(REL_q, pr+1) ← no_ret_sim / i
          | for q ∈ 1..MQ
          |   for pr ∈ 10,9..1
          |     P_(REL_q, pr) ← P_(REL_q, pr+1) if P_(REL_q, pr) = 0
          | for pr ∈ 1..11
          |   p_pr ← mean(p^⟨pr⟩)
          | p
```

15. Plot precision-recall graph: mean(PR_INF) = ∎ r := 1..11



Appendix C.

Results obtained with the MLS method

Query	WebCIR	Yahoo!	Altavista	MSN
dax4p1-albérlet	1	0,141843972	0,141843972	1
dax4p1-egyetemünk	0,453900709	0,070921986	0,120567376	0,191489362
dax4p1-erasmus	1	0,929078014	0,879432624	1
dax4p1-kollégiumi várólista	0,5	0	0,350877193	0
dax4p1-regatta	1	0,333333333	0,283687943	0,595744681
dax4p1-telefonkönyv	0,141843972	0,262411348	0,191489362	0,617021277
dax4p1-vizsgaszabályzat	0,858156028	0,737588652	0,737588652	0,666666667
FIYRG5-ösztöndíj számítás	0,435643564	0,435114504	0,41221374	0,212765957
FIYRG5-tanulmányi ösztöndíj	0,120567376	0,737588652	0,475177305	0,404255319
FIYRG5-vízesés modell	1	0,929078014	0,879432624	1
FIYRG5-záróvizsga jelentkezés	1	0,787234043	0,929078014	0,929078014
LGNTAK-albérlet	1	0,141843972	0,141843972	1
LGNTAK-egyetemi telefonkönyv	0,638297872	0,858156028	1	1
LGNTAK-tdk	0,716312057	0,858156028	0,808510638	0,929078014
LGNTAK-záróvizsga tételek	0,787234043	0,666666667	0,787234043	0,645390071
P1Z3VF-dékáni titkárság telefon	0,404255319	1	0,858156028	0,929078014
P1Z3VF-gazdasági informatika tárgy	0,475177305	0,475177305	0,404255319	0,546099291
P1Z3VF-köztársasági ösztöndíj pályázati lap	0,78021978	0,603960396	0,504950495	0,305785124
P1Z3VF-moodle	0,858156028	0,141843972	0,120567376	0,595744681
P1Z3VF-szociális támogatás	0,546099291	0,666666667	0,716312057	0,929078014
P9BDT4-államvizsga beosztás	0,540540541	0,90990991	0,90990991	0,540540541
P9BDT4-egyetem fotógaléria	1	0,305785124	0,305785124	0,262411348
P9BDT4-information retrieval	0,666666667	0,858156028	0,787234043	0,354609929
P9BDT4-karrier iroda	0,141843972	0,262411348	0,241134752	0
P9BDT4-levlist	0,354609929	0,382978723	0,404255319	0,141843972

P9BDT4-orvosi rendelés	0	0	0	0
P9BDT4-tdk	0,595744681	0,716312057	0,858156028	0,496453901
q2irvj-levlista feliratkozás	1	1	1	0,77027027
q2irvj-tanév időbeosztása	0,333333333	0,879432624	1	0,262411348
u3619x-karrier iroda	1	0,404255319	0,382978723	0,354609929
u3619x-költségtérítési díj	0,929078014	0,858156028	0,929078014	0,879432624
u3619x-rektori körlevél	0,649122807	0,929078014	0,929078014	0,406593407
u3619x-tavaszi szünet	0,564885496	0,858156028	0,929078014	0,645390071
u3619x-tdk	0,716312057	0,858156028	0,808510638	0,808510638
X7GCUE-áthallgatás	0,564885496	0,574468085	0,524822695	0,546099291
X7GCUE-gazdaságinformatikus kreditek	1	1	1	1
P10 (average)	**0,660357954**	**0,599275524**	**0,604256667**	**0,582401438**
Standard deviation	**0,296432307**	**0,321527031**	**0,321632889**	**0,324209681**

Appendix D.

Results obtained with the DCG method

Query	WebCIR	Yahoo!	Altavista	MSN
dax4p1-albérlet	2,948459119	1	1	2,948459119
dax4p1-egyetemünk	0,817529365	0	0,5	0,430676558
dax4p1-erasmus	2,948459119	2,948459119	2,517782561	2,948459119
dax4p1-regatta	2,948459119	1,061606312	1,630929754	1,817529365
dax4p1-telefonkönyv	1	1,386852807	0,5	1,430676558
dax4p1-vizsgaszabályzat	2,948459119	2,561606312	2,561606312	1,886852807
FIYRG5-ösztöndíj számítás	0,930676558	2,017782561	1,817529365	0
FIYRG5-tanulmányi ösztöndíj	0,5	2,448459119	1,130929754	0,430676558
FIYRG5-záróvizsga jelentkezés	2,948459119	2,948459119	2,948459119	2,948459119
LGNTAK-albérlet	2,948459119	1	1	2,948459119
LGNTAK-egyetemi telefonkönyv	1,630929754	1,948459119	2,948459119	2,948459119
LGNTAK-tdk	1,317529365	2,948459119	2,448459119	2,948459119
LGNTAK-záróvizsga tételek	2,948459119	2,448459119	2,948459119	2,317529365
P1Z3VF-dékáni titkárság telefon	2,130929754	2,948459119	2,948459119	2,948459119
P1Z3VF-gazdasági informatika tárgy	0,386852807	1,017782561	0,386852807	1,061606312
P1Z3VF-moodle	1,948459119	1	0,5	2,448459119
P1Z3VF-szociális támogatás	2,130929754	2,517782561	2,948459119	2,948459119
P9BDT4-information retrieval	2,517782561	2,948459119	2,948459119	1
P9BDT4-karrier iroda	0	1,061606312	0,930676558	0
P9BDT4-levlist	1	1,930676558	2,130929754	1
P9BDT4-tdk	0,817529365	1,948459119	1,948459119	1,630929754
u3619x-karrier iroda	2,948459119	2,061606312	1,930676558	1,630929754
u3619x-költségtérítési díj	2,948459119	2,948459119	2,948459119	2,517782561
u3619x-tavaszi szünet	2,561606312	2,948459119	2,948459119	2,948459119
u3619x-tdk	1,317529365	2,948459119	2,448459119	2,561606312
X7GCUE-áthallgatás	0,930676558	1,317529365	0,930676558	1,061606312
DCG	**48,47509271**	**52,31634109**	**49,90164029**	**49,7629934**

Appendix E.

Results obtained with the RC method

Query	WebCIR	Yahoo!	Altavista	MSN
albérlet	4	3	3	4
áthallgatás	3	6	7	6
ösztöndíj számítás	4	5	5	0
dékáni titkárság telefon	2	4	3	1
doktori iskola	3	2	5	2
egyetemünk	3	5	5	3
egyetemi kiadó	2	1	4	3
egyetemi telefonkönyv	5	3	3	5
erasmus	1	0	2	1
etv regatta	7	9	9	7
gazdasági informatika tárgy	0	6	6	4
hefop pályázat	0	4	4	2
information retrieval	4	2	6	4
karrier iroda	3	7	7	3
költségtérítési díj	0	5	5	2
kollégium	0	4	3	1
levlist	3	4	5	2
moodle	1	0	1	2
neptun	0	4	4	0
nyelviskola	0	4	4	4
regatta	2	4	4	2
szmsz	0	7	6	5
szociális támogatás	4	6	6	4
tanulmányi ösztöndíj	1	4	4	1
tavaszi szünet	5	6	9	8
tdk	3	4	6	3
tdk eredmények	2	1	3	2
telefonkönyv	1	3	3	1
ven	0	5	4	3
vizsgaszabályzat	4	4	4	4
záróvizsga jelentkezés	6	4	7	5
záróvizsga tételek	0	4	4	0
RC	**73**	**130**	**151**	**94**

Appendix F.

Results obtained with the RP method

Query	WebCIR	Yahoo!	Altavista	MSN
albérlet	0,5	0,3	0,4	0,5
áthallgatás	0,111111	0,4	0,6	0,5
ösztöndíj számítás	0,333333	0,444444	0,555556	0
dékáni titkárság telefon	0,2	0,4	0,4	0,1
doktori iskola	0,3	0,4	0,6	0,2
egyetemünk	0,1	0,4	0,3	0,2
egyetemi kiadó	0,1	0,3	0,5	0,5
egyetemi telefonkönyv	0,5	0,2	0,5	0,3
erasmus	0,3	0	0,3	0,1
etv regatta	0,5	0,6	0,6	0,4
gazdasági informatika tárgy	0	0,5	0,6	0,3
hefop pályázat	0	0,4	0,4	0,2
information retrieval	0,2	0,3	0,8	0,2
karrier iroda	0,1	0,5	0,5	0,1
költségtérítési díj	0,1	0,5	0,5	0,2
kollégium	0,2	0,4	0,5	0,2
levlist	0,2	0,4	0,3	0,4
moodle	0	0,5	0,5	0,2
neptun	0,3	0,5	0,8	0,5
nyelviskola	0,6	0,5	0,5	0,8
regatta	0,2	0,4	0,5	0,1
szmsz	0,2	0,5	0,5	0,2
szociális támogatás	0,1	0,4	0,4	0,1
tanulmányi ösztöndíj	0,1	0,4	0,6	0,2
tavaszi szünet	0,222222	0,6	0,7	0,3
tdk	0,1	0,2	0,2	0,1
tdk eredmények	0,2	0,4	0,6	0,1
telefonkönyv	0,1	0,5	0,4	0,3
ven	0,3	0,5	0,2	0,2
vizsgaszabályzat	0,3	0,5	0,7	0,2
záróvizsga jelentkezés	0	0,6	0,6	0
záróvizsga tételek	0,3	0,4	0,4	0,4
RP	**0,211458**	**0,417014**	**0,498611**	**0,253125**
Standard Deviation	**0,155747**	**0,124785**	**0,149518**	**0,174105**

Appendix G.

Results obtained with the RC method after Altavista's hit lists were excluded.

Query	WebCIR	Yahoo!	MSN
albérlet	4	0	4
áthallgatás	2	3	3
ösztöndíj számítás	2	2	0
dékáni titkárság telefon	2	1	1
doktori iskola	1	0	1
egyetemünk	2	2	2
egyetemi kiadó	1	0	1
egyetemi telefonkönyv	4	2	4
erasmus	0	0	0
etv regatta	5	4	5
gazdasági informatika tárgy	0	2	2
hefop pályázat	0	1	1
information retrieval	2	0	2
karrier iroda	2	2	2
költségtérítési díj	0	1	1
kollégium	2	1	1
levlist	1	1	2
moodle	0	0	0
neptun	2	2	2
nyelviskola	4	3	5
regatta	0	1	1
szmsz	3	2	3
szociális támogatás	1	0	1
tanulmányi ösztöndíj	1	1	2
tavaszi szünet	2	0	2
tdk	1	0	1
tdk eredmények	1	0	1
telefonkönyv	0	3	3
ven	3	2	3
vizsgaszabályzat	3	2	3
záróvizsga jelentkezés	0	0	0
záróvizsga tételek	4	3	5
RC	**55**	**41**	**64**

Appendix H.

Results obtained with the RP method after Altavista's hit lists were excluded.

Query	WebCIR	Yahoo!	MSN
albérlet	0,5	0	0,5
áthallgatás	0,111111	0,2	0,4
ösztöndíj számítás	0,333333	0,222222	0
dékáni titkárság telefon	0,2	0,1	0,1
doktori iskola	0,2	0,1	0,2
egyetemünk	0,1	0,1	0,2
egyetemi kiadó	0,1	0	0,3
egyetemi telefonkönyv	0,4	0,1	0,3
erasmus	0,2	0	0
etv regatta	0,5	0,3	0,4
gazdasági informatika tárgy	0	0,2	0,2
hefop pályázat	0	0,1	0,2
information retrieval	0,2	0	0,2
karrier iroda	0,1	0,1	0,1
költségtérítési díj	0,1	0,2	0,2
kollégium	0,2	0,1	0,1
levlist	0,2	0,1	0,4
moodle	0	0	0,2
neptun	0,3	0,3	0,2
nyelviskola	0,6	0,3	0,8
regatta	0,2	0,1	0,1
szmsz	0,2	0,1	0,2
szociális támogatás	0,1	0	0,1
tanulmányi ösztöndíj	0,1	0,1	0,2
tavaszi szünet	0,222222	0,2	0,3
tdk	0,1	0	0,1
tdk eredmények	0,1	0,1	0,1
telefonkönyv	0,1	0,3	0,3
ven	0,3	0,1	0,2
vizsgaszabályzat	0,2	0,3	0,2
záróvizsga jelentkezés	0	0,1	0
záróvizsga tételek	0,3	0,2	0,4
RP (average)	**0,195833**	**0,128819**	**0,225**
Standard Deviation	**0,148717**	**0,099735**	**0,16264**