The Vector Space Model and some suggestions

Nikitas N. Karanikolas
Technological Educational Institute (TEI) of Athens
nnk@teiath.gr

Introduction

Information Retrieval is a term referred to the activity for discovering resources relevant to our information needs. Usually, we have to describe our information needs (formulate our query) and we are expecting to get a list or relevant resources (e.g. articles, pictures, etc). In case that the requested resources are documents (e.g. newspaper articles, scientific articles, regulations, judgements, statutes, technical manuals, culinary formulas, etc) we talk about *Text Retrieval*, but we also can use the broader term Information Retrieval. In case that the requested resources are images (e.g. pictures, photographs, paintings, graphs, medical images, etc) we talk about *Image Retrieval*. In this lecture we focus on Text Retrieval but some of the ideas can also apply for Image Retrieval.

The terms *Full Text Search* and *Full Text Retrieval* are used in order to emphasize that the indexing of documents/texts process (and consequently the search process) is based on the whole content of documents/texts. It is distinguished from indexing and search processes that are based on parts of the original documents/texts (e.g. using only titles or abstracts or selected sections).

The formulation of queries (for expressing the information needs) can follow some query language. For example *Boolean query* combine usual words with operators from the set {AND, OR, NOT, … }. Some other languages are more restrictive ones and demand from user to define the structural location (e.g. title, abstract, references, authors, …) where the word (or phrase) should exist in the document. More details about query formulation can be found in some other papers (Karanikolas 2011; Karanikolas and Skourlas 2011). Boolean queries do not provide some order between the retrieved documents. Usually, users define vary relaxed Boolean queries (by OR-ing terms) and get a list with many documents outside their information needs. In the other edge is the case that users define very restrictive Boolean queries (by AND-ing terms) and get a small subset of the documents that are relevant with their information needs.

To solve this problem, there exists the *Free Text Search* that allows the user to describe his/her needs by simply mentioning (place side by side) words. This approach returns a list of documents assigning a relevance value to each one of them. Usually, in the top of the list are the more relevant documents (having the greatest relevance value) and at the end are the less relevant ones (having the lowest relevance value). In this lecture we are talking about the Free Text Search (and Retrieval) approach. We actually suppose that the audience has knowledge of the *Vector Space Model* and the Free Text Search approach and we go further by providing some suggestions for improving this model. The interested reader that does not have knowledge of the Vector Space Model and of the Free Text Search should first read some relevant book. For example sections 3.1 and 3.2 from the book *Modern*

Information Retrieval Second Edition – shortly MIR2ed – by Baeza-Yates and Ribeiro-Neto, 2010, should be a good starting point.

TF – IDF

Term Frequency and Inverse Document Frequency are discussed in many Information Retrieval text books. Such are the book of Salton (1983), the book of Rijsbergen (1979), the book of Kowalski (1997) and more recently the book of Baeza-Yates and Ribeiro-Neto (2010). In the mentioned books, the interested reader can learn about the Vector Space Model and the other models that are used in Information Retrieval.

Similarity measures

Based on the last suggestion of (MIR2ed) for TF-IDF and the cosine function, we can measure the similarity of a document with a query as follows:

$$S(D_j, Q) = \frac{\sum_{i=1}^{t} q_i w_{ij}}{\sqrt{\sum_{i=1}^{t} q_i^2 \cdot \sum_{i=1}^{t} w_{ij}^2}} = \frac{\sum_{i=1}^{t} q_j w_{ij}}{L_Q \cdot L_{Dj}}$$

$$= \frac{\sum_{i=1}^{t}\left(\left(1+\log f_{i,j}\right)*\log\frac{N}{n_i}\right)*\left(\left(1+\log f_{i,q}\right)*\log\frac{N}{n_i}\right)}{\sqrt{\sum_{i=1}^{t}\left(\left(1+\log f_{i,j}\right)*\log\frac{N}{n_i}\right)^2}*\sqrt{\sum_{i=1}^{t}\left(\left(1+\log f_{i,q}\right)*\log\frac{N}{n_i}\right)^2}} \quad (1)$$

where:

t is the number of index terms

N is the number of documents in the collection

n_i is the number of documents that contain the index term i

$f_{i,j}$ is the number of occurrences of index term i into the document j

$f_{i,q}$ is the number of occurrences of index term i into the query (usually 1)

Another TF-IDF suggestion (Lucarella, 1988) is to use the double normalization for term weight and the logarithmic inverse document frequency. Based on this TF-IDF alternative and the cosine function, we can measure the similarity of a document with a query as follows:

$$S(D_j, Q) = \frac{\sum_{i=1}^{t} q_i w_{ij}}{\sqrt{\sum_{i=1}^{t} q_i^2 \cdot \sum_{i=1}^{t} w_{ij}^2}} = \frac{\sum_{i=1}^{t} q_j w_{ij}}{L_Q \cdot L_{Dj}}$$

$$= \frac{\sum_{i=1}^{t} \left(0.5 + 0.5 \cdot \frac{f_{i,j}}{\max f_j}\right) * \left(\log \frac{N}{n_i}\right)}{\sqrt{\sum_{i=1}^{t} \left(0.5 + 0.5 \cdot \frac{f_{i,j}}{\max f_j}\right)^2} * \sqrt{\sum_{i=1}^{t} \left(\log \frac{N}{n_i}\right)^2}} \quad (2)$$

where:

t is the number of index terms

N is the number of documents in the collection

n_i is the number of documents that contain the index term i

$f_{i,j}$ is the number of occurrences of index term i inside the document j

$\max f_j$ is the maximum term frequency for document j

The double normalization of term weight

The normalization is double because:

- The weight considers the document term frequency in relation with other term frequencies for the same document. This is why $f_{i,j}$ is divided with the maximum term frequency ($\max f_j$) of the examined document (j)
- The weight is in the range of 0.5 – 1.0 if the term exists in the document, otherwise the weight is zero.

Implementation issues

In case of equation 1 we need:

- an inverted file for index terms where each index term (i) has a list of frequencies of term for the documents (j) where the term exists ($f_{i,j}$)
- a term file where for each term (i) it contains the number of documents (n_i) that contain the index term

In case of equation 2 we need:

- an inverted file for index terms where each index term (i) has a list of frequencies of term for the documents (j) where the term exists ($f_{i,j}$)
- a term file where for each term (i) it contains the number of documents (n_i) that contain the index term
- a document file where for each document (j) it contains the maximum term frequency for the document ($\max f_j$)

Document Length

$$L_{D_j} = \sqrt{\sum_{i=1}^{t} w_{ij}^2}$$

$$= \sqrt{\sum_{i=1}^{t} \left(0.5 + 0.5 \cdot \frac{f_{i,j}}{\max f_j} \right)^2} \quad (3)$$

Our experimentation gave us an indication that equation (2) presents a “preference” to short documents against longer ones. To tackle the problem of "preference" of short documents against longer ones we shall decrease L_D for longer documents. Our suggestion (Karanikolas and Skourlas 2000; Karanikolas 2007) is to replace equation (3) with equation (4)

$$L_{D_j} = \ln\left(\sum_{i=1}^{t} w_{ij}^2 + e - 1 \right) \quad (4)$$

Phrases

So far we have not considered the use of phrases for the calculation of Document versus Query similarity. For the exploitation of phrases we suggest (Karanikolas 2009) the following contribution of each phrase in the nominator of equation (2):

$$c \cdot q_{\{A..\Delta\}} \cdot \left(0.5 + 0.5 \cdot \frac{f_{\{A..\Delta\}j}}{\max f_j} \right) + B \cdot \sum_{x \in \{A..\Delta\}} q_x \cdot w_{xj} \quad (5)$$

This is the calculation of a phrase's contribution and replaces $q_j w_{ij}$ (the contribution used otherwise – for simple query words). The following are the new terms introduced by (5):

$$w_{xj} = 0.5 + 0.5 \cdot \frac{f_{xj}}{\max f_j}$$

(The weight of phrase for document)

$$q_x = \log_2 \left(\frac{N}{n_x} \right) \text{ where } x \in \{\text{A..}\Delta\}$$

(The weight of some word participating in a query's phrase)

$$q_{\{\text{A..}\Delta\}} = \max_{x \in \{\text{A..}\Delta\}} (q_x)$$

(The weight of a query's phrase. Phrase discriminates as match as the most discriminating constituent word.)

$$B = b / |\{\text{A..}\Delta\}|$$

(For example b=0.5, $|\{\text{A..}\Delta\}|$=4, B=0.125)

$\max f_j$ is the maximum frequency of simple term (not phrase) appearance in document j

$f_{\{\text{A..}\Delta\}j}$ is the frequency of the phrase $\{\text{A..}\Delta\}$ in document j

c and b are constants determining what is the contribution of the phrase and what is the contribution of the phrase constituents (words).

For the denominator of equation (2), the calculation of LDj remains unchanged. However, for the calculation of LQ, its calculation considers simple terms (q_i) and phrases ($q_{\{\text{A..}\Delta\}}$) and none of the phrase constituents (q_x where $x \in \{\text{A..}\Delta\}$). In other words, every phrase (compound term) of the query is taken as a simple term but with increased weight. The standard weight is given by:

$$c \cdot q_{\{\text{A..}\Delta\}} \cdot \left(0.5 + 0.5 \cdot \frac{f_{\{\text{A..}\Delta\}j}}{\max f_j} \right)$$

and the increased weight result by the add up of:

$$B \cdot \sum_{x \in \{\text{A..}\Delta\}} q_x \cdot w_{xj} .$$

Our motivation for using increased weightiness is double:

- The documents that contain the query phrase (compound term) gain a heavy bounty.
- The documents that do not contain the query phrase but contain some of the phrase constituents (words) gain some (less heavy) bounty. (In this way the documents that include only phrase constituents appear at the end of the results list.)

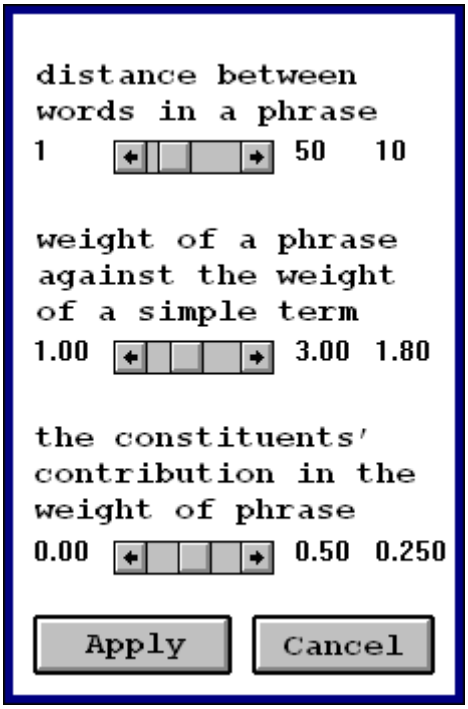
It remains to explain which is the role of the parameters c and B and how this parameters can be configured. The parameter c determines the contribution of the phrase (as an unbreakable whole). The parameter B determines which is the contribution of each of the phrase constituents (words). Although parameter B is a necessary part of (5), it is not of interest to the user. The user should be able to determine what is the contribution of all phrase constituents (words). This contribution is determined by the parameter b. The parameter B results as a division with nominator the parameter b and denominator the number of words that constitute the phrase.

The user's intervention can be based on two handlers. First handler should define what is "the weight of a phrase against the weight of a simple term". This handler determines *the sum of parameters* c *and* b. The second handler should define what is "the constituents' contribution in the weight of phrase". This second handler determines *the value of* $b/(c+b)$. For example if the user defines that "the weight of a phrase against the weight of a simple term" is 1.32 and "the constituents' contribution in the weight of phrase" is 0.25, then (c+b)=1.32 and (b/(c+b))=0.25. Consequently, b=0.33 and c=0.99.

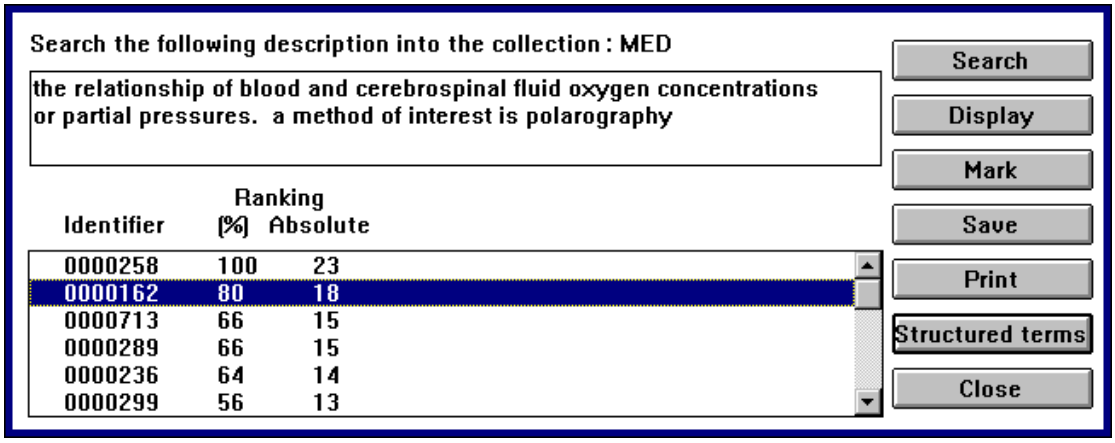
The range of values of the first handler ("the weight of a phrase against the weight of a simple term"), in our opinion, should be from 1.0 to 3.0. The range of values of the second handler ("the constituents' contribution in the weight of phrase"), in our opinion, should be from 0.0 to 0.5. These ranges are not definite but they can be used as a first approach.

User Interface

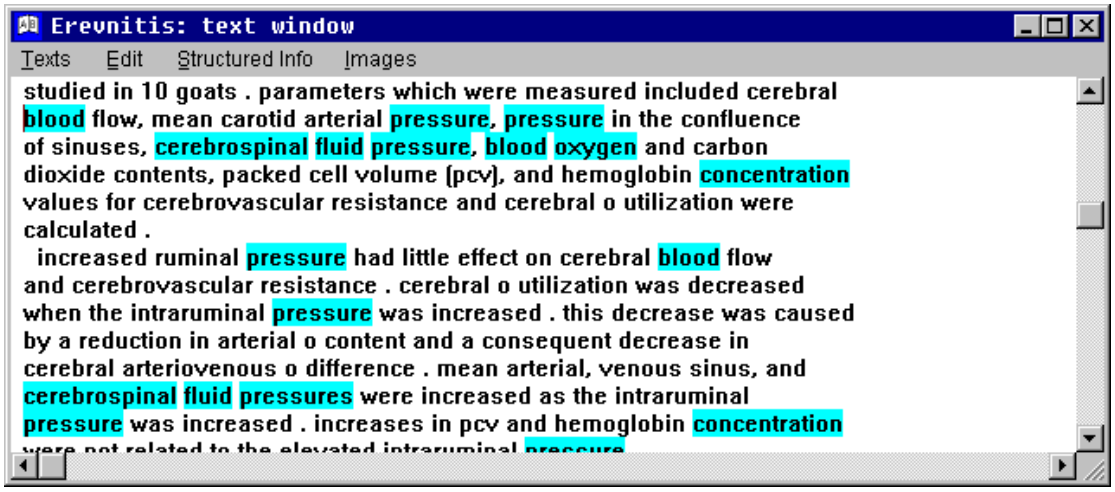
Erevnitis is an IR system developed in 1994. *Erevnitis* is implemented according the suggestions given in this document. More details about *Erevnitis* can be found in Moumouris (1995) and Paparhanasiou (1996). The following figures are screenshots from *Erevnitis*. The only thing that is not explained by the next figures is how the user defines a query's phrase. This is based on the use of tilde grapheme (character). For example, the user could have entered "cerebrospinal~fluid" instead of "cerebrospinal fluid" in order to define a phrase.



User configured parameters for phrase matching.



Query and relevant documents in descending order according to Ranking.



The presentation of some relevant document.

References

Nikitas N. Karanikolas, Search Culture. PCI'2011: 15th Panhellenic Conference on Informatics, 30 September – 2 October 2011, Kastoria, Greece. IEEE CPS. ISBN: 978-0-7695-4389-5.

Nikitas N. Karanikolas and Christos Skourlas, Exploiting the Search Culture modulated by the Documentation Retrieval applications. IC-ININFO: International Conference on Integrated Information, September 29 – October 3, 2011, island of Kos, Greece.

Salton, G., 1983, Introduction to Modern Information Retrieval, 1st edn (Printed in the USA;McGraw-Hill), ISBN 0-07-054484-0.

Rijsbergen, C. J. van, 1979, Information Retrieval, 2nd edn (London; Butterworths).

Kowalski G., 1997, Information Retrieval Systems: Theory and Implementation, 1st edn (Printed in the USA; Kluwer Academic Publishers), ISBN 0-7923-9899-8.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto, 2010, Modern Information Retrieval Second Edition.

N. Karanikolas and C. Skourlas. Computed Assisted Information Resources Navigation. Medical Informatics and the Internet in Medicine, volume 25, No 2, 2000.

Nikitas N. Karanikolas. The measurement of similarity in stock data documents collections. eRA-2. 2nd Conference for the contribution of Information Technology to Science, Economy, Society and Education, September 22-23, 2007, Athens, Greece.

Nikitas N. Karanikolas. The role of phrases in Information Retrieval and related domains. eRA-4. 4th Conference for the contribution of Information Technology to Science, Economy, Society and Education, September 25-26, 2009, Spetses, Greece.

D. Lucarella, A document retrieval system based on nearest neighbour searching, *Journal of Information Science* 14(1) (1988) 25-33.

Moumouris, N.,1995, The document retrieval system Erevnitis. *CHIP (Greek edition)*, no 11, March, 60-61.

Papathanasiou, S., 1996, Text researcher. *Techniki Eklogi, (in Greek)*, no 359, November, 100-101.