ENHANCING HIGH PRECISION BY COMBINING OKAPI BM25 WITH STRUCTURAL SIMILARITY IN AN INFORMATION RETRIEVAL SYSTEM

Yaël Champclaux, Taoufiq Dkaki and Josiane Mothe
IRIT, équipe SIG, Université de Toulouse, 118 Route de Narbonne Toulouse, France

Keywords: Information Retrieval, Structural similarity, Graph comparison, Term weighting.

Abstract: In this paper, we present a new similarity measure in the context of Information Retrieval (IR). The main objective of IR systems is to select relevant documents, related to a user's information need, from a collection of documents. Traditional approaches for document/query comparison use surface similarity, i.e. the comparison engine uses surface attributes (indexing terms). We propose a new method which combines the use of both surface and structural similarities with the aim of enhancing precision of top retrieved documents. In a previous work, we showed that the use of structural similarity in combination with cosine improves bare cosine ranking. In this paper, we compare our method to Okapi based on BM25 on the Cranfield collection. We show that structural similarities improve average precision and precision at top 10 retrieved documents about 50%. Experiments also address the term weighting influences on system performances.

1 INTRODUCTION

Many applications require the use of similarity measures. This is the case for information retrieval (IR), where determining whether or not a given piece of information corresponds to a user needs is mainly based on uncovering similarities between documents and queries -queries are the expressions of the user needs whereas documents are the information sources handled by information retrieval systems (IRS). This central issue in IR involves a complex task that aims at determining if a document is sufficiently similar to a user's query to be retrieved.

However, similarity is a difficult concept to define and to use in the context of IR, as it is in many areas related to cognition as in pattern recognition, clustering and categorization (Jones, 1993)(Medin, 1990), case-based reasoning and generalization. Similarity is the cornerstone of the understanding of these areas and of the implementation of related-applications. In IR, similarity is a component that shapes IR models. An IRS compares documents against a query to select those documents that may be useful to the user. The comparison is usually performed on some document and query representations rather than on the primary documents and queries. Representation space and comparison method define the IR model.

The graph-based model we present in this paper is an algebraic model closely related to the vector space model (Salton, 1975). A vector space model considers each document as a vector in the space defined by the set of terms that the system collects during the indexing phase. Each vector coordinate is a value representing the importance of the term in the document or in the query that the vector represents. Many similarity measures such as the cosine measure, the Jaccard measure, the Dice coefficient… are used to determine how well a document corresponds to a query. Such measures determine local similarities between a document and a query on the basis of the terms they have in common. Our goal is to exploit another type of similarities called structural similarities. These similarities identify resemblances between two elements on the basis of their relationships to the remaining elements (Halford, 1988). The relational structure that we use originates from the fact that documents contain words and that words are contained in documents. The idea is to compare these documents through the similarities between the

Champclaux Y., Dkaki T. and Mothe J. (2009).
ENHANCING HIGH PRECISION BY COMBINING OKAPI BM25 WITH STRUCTURAL SIMILARITY IN AN INFORMATION RETRIEVAL SYSTEM.
In *Proceedings of the 11th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 279-285
DOI: 10.5220/0002017202790285

words they contain while similarities between words are themselves dependent on similarities between the documents they are contained in. In our first paper (Champclaux, 2007), we have shown that the only use of the structural similarity we proposed was not sufficient to improve the performance of an IRS. Then in a later paper (Champclaux, 2008), we presented a different model that combines the use of both structural and surface similarities, and we showed that our SimRank measure combined with the cosine can improve the high precision. In this paper, we experiment our SimRank measure in combination with a more efficient measure namely Okapi.

The remainder of this paper is structured as follows: Section 2 presents related works in graph theory when used in IR and management fields. In section 3, we describe our approach. Section 4 deals with the evaluation of our method, section 5 comments and discusses the results we obtain. This paper will be concluded by giving some perspectives to our work.

2 RELATED WORKS

The earliest paper on graph theory is said to be the one by Leonhard Euler in (Euler 1736) where he discusses whether or not it is possible to stroll around the town of Konigsberg crossing each of its bridges across the river exactly once. Euler gave the necessary conditions to do so. Two century later, Claude Berge lays the groundwork of this field in his book (Berge, 1958).From the sixties to the present, graphs have been used to model real world problems, especially those related to networks: electric circuits, biological network, social network, transport network, computer network, World Wide Web.

Modeling problems with graphs have paved the way to new approaches to solve them. We use a sub-field of graph theory –namely graph comparison- to provide new solutions for IR.

In (Blondel, 2004), Vincent Blondel laid the ground for graph comparison in the context of information management. Blondel's method compares each node of a graph to every node of another graph. The approach in (Blondel, 2004) is presented as a generalization of Kleinberg's method (Kleinberg, 1999) which associates authority and hub scores to web pages to enhance web search accuracy. Blondel's comparison is based on a similarity measure that takes into account the neighboring nodes of the compared nodes in the graph they belong to. This makes it possible to determine which node of a graph being analyzed behaves like a given node of a graph considered as a model. This method has been successfully applied to web searching and synonym extraction (Blondel, 2004).

More generally, graph comparison has been used in many fields such as biological networks comparison using phylogenetic trees from metabolic pathway data (Heymans 2003); Social network mapping and small world related phenomenon (Milgram, 1967)(Watts, 1999); Chemical structure matching and similar structures uncovering from a chemical database (Hattori, 2003). Our method could be related to Latent Semantic Indexing (Deerwester, 1990), or neural network (NN) (Belew, 1989). Indeed, both methods try to capture the added-value of documents-terms interrelationship. The LSI method decomposes the document-term matrix in a combination of three matrices which represent the information of the original matrix in a different space where similar documents and similar terms are closer as a direct consequence of the underlying space reduction. Our method creates links between pairs of objects when each element of the pair is related to an element of a previously linked pair. As in the LSI, we can build similarity measures between documents, between terms and between documents and terms. The aim of our method is not to reduce the representation space, but, rather, to find all indirect similarities that exist with the queries. Regarding NN, the retrieval mechanism is based on the neural activation that is propagated from query nodes to document nodes through the network synapses. In our approach, terms nodes and documents nodes are directly related to each other under indexing considerations whereas, in IR based on NN, they are related to each other following an a priori heuristic that may involve a hidden neuron layer. In our approach there is a back and forth calculation of documents and terms similarities. At the initial step of our method, we consider that the similarity between any pair of separate documents (resp. terms) is nil; then we evaluate the similarity between terms on the basis of the similarity between documents they index (calculated similarity). After this, we evaluate the document to document similarity on the basis of the previously calculated term similarities, and then repeat those steps until convergence is reached. This is a back and forth automatic similarity refining. Sophisticated neural approaches (Mothe, 1994) does propagate and retro propagate neural activation just once. Hyperspace Analog to Language (Burges,

1998), Latent Semantic Analysis (Landauer, 1998) and the Correlated Occurrence Analog to Lexical Semantics (Rhode, 2004) are models based on the assumption that words are similar if they co-occur in similar contexts. The models tabulate co-occurrence in a matrix, in which each row vector codes the co-occurrence frequency of one word with every other word in a 4-word window (COALS), in a 10-word window (HAL) or within a single document (LSA). Those models exploit lexical co-occurrence information to represent a certain type of similarity between words: the contextual co occurrence. The focus of these models is on word-similarity. Our method can also be used as a word similarity measure but structural rather than contextual. The common concept of these applications is similarity which defines in what regard objects or items are alike.

3 OUR METHOD

We use graphs to define structural similarities for IR purposes. Our method uses graph structure to capture structural information of documents and queries. As structural similarities are known to enhance retrieval precision (Forbus, 1995), we expect that our method which is based on graph comparison will have lead to good high precision information retrieval system.

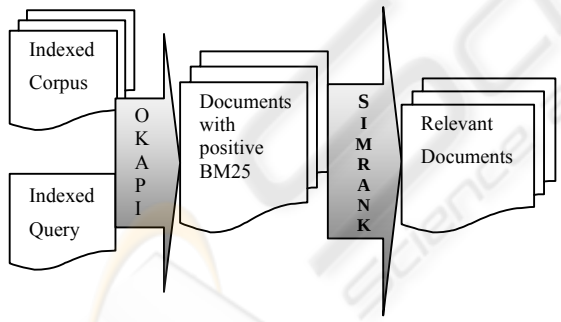

Figure 1: A two-phase ranking process.

The method we propose consists of two stages: first, documents are filtered using the Okapi BM25 ranking function (Robertson, 1994) and a prefixed threshold[1]; then the selected documents and the query are stored in the corpus graph then sorted using a SimRank-based score. We call this 2-stages method OkaSim.

[1] In practice, we chose zero as a threshold.

3.1 Phase 1: Okapi Sorting

First, we rank documents using the Okapi BM25 similarity measure which is one of the most popular functions used in IR. Okapi BM25 not only considers the frequency of the query terms, but also the average length of the whole collection and the length of the document under evaluation.

$$\sum_{t_i \in q} \left[\log \frac{N}{n}\right] \cdot \frac{(k_1+1)tf_{id}}{k_1\left((1-b)+b\times\left(\frac{dl}{avdl}\right)\right)+tf_{id}} \cdot \frac{(k_3+1)tf_{iq}}{(k_3+tf_{iq})} \quad (1)$$

Where Q is a query containing term t

N is the number of documents in the collection
n is the number of documents containing the term
tf is the frequency of occurrence of term t within a specific document
tf_{iq} is the frequency of the term i within the query.
dl and $avdl$ are the length of document d and the average document length for the whole collection

The variable $k1$ is a positive tuning parameter that calibrates the document term frequency scaling. A k_1 value of 0 corresponds to a binary model (no term frequency), and a large value corresponds to using raw term frequency.

b is another tuning parameter ($0 <= b < 1$) which determines the scaling by document length: $b=1$ corresponds to fully scaling the term weight by the document length, while $b=0$ corresponds to no length normalization.

k_3 being another positive tuning parameter that this time calibrates term frequency scaling of the query.

We choose $k_1=1.2$; $b=0.75$; $k_3=7.0$. Further information about Okapi measure is available in (Spark Jones, 2000).

We use the Okapi ranking function to filter the documents: only the documents for which the Okapi similarity value is superior to a threshold ($RSV(d,q) > \tau$) are retained. We then apply phase 2 in order to sort those documents using the method SimRank (Champclaux, 2007).

3.2 Phase 2: SimRank Sorting

The aim of this step is to find documents that are structurally relied to the query. We consider the bipartite graph where documents and terms are nodes and where each edge between a document and a term reflects the fact that this term indexes that document. As shown in figure 2, the query is viewed as a document-type node of the bipartite graph.

To compute the structural similarity between a query and the documents, we iterate an algorithm which initialize a document to document similarity

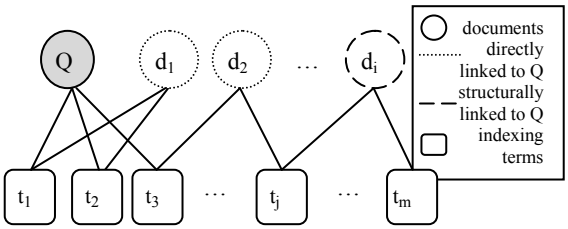

Figure 2: Corpus as a bipartite graph.

matrix, then calculate the term to term similarity matrix using the built graph. We follow the idea that a term is similar to another one if they appear in similar documents and a document is similar to another document if they are related to similar terms.

The similarity function between two documents (*Sd*) which updates the similarity values in the document to document matrix and the similarity function between two terms (*St*) which updates de similarity values in the term to term matrix are expressed as follows:

$$S_d(d_1,d_2)=\frac{C_1}{|T_{d_1}||T_{d_2}|}\sum_{i\in T_{d_1}}\sum_{i\in T_{d_2}}S_t\left(t_i(d_1),t_j(d_2)\right) \quad (2)$$

$$S_t(t_1,t_2)=\frac{C_2}{|D_{t_1}||D_{t_2}|}\sum_{i\in D_{t_1}}\sum_{i\in D_{t_2}}S_d\left(d_i(t_1),d_j(t_2)\right) \quad (3)$$

Where d_i is the i^{th} document of the document collection,

t_i is the i^{th} indexing term,

$|T_{di}|$ (resp. $|T_{dj}|$) is the number of terms in document i (resp. j),

$|D_{ti}|$ (resp. $|D_{tj}|$) is the number of documents in which t_i (resp. t_j) appears,

C_1 and C_2 are two propagation coefficients, which values are between 0 and 1. C_1 (resp. C_2) is used to moderate the similarity between two terms (resp. documents) which is calculated as an average similarity between documents (resp. terms) containing (resp. contained in) the compared terms (resp. documents). In (Champclaux, 2007), we choose to use $C_1 = C_2 = 0{,}95$ to not have too small values.

(2) defines the similarity between two documents as a function of similarity measures between the indexing terms they are related to. Symmetrically, (3) defines the similarity between terms as a function of the similarity measures between the documents they are related to. Thus, for a given query, we compute the similarity to each document in the collection, on the basis of the similarities between the indexing terms; then we compute the similarity between the terms on the basis of the previously computed similarities between the documents. This represents a first iteration; which is repeated a given number of times. At each iteration similarities are updated taking into account the similarities computed during the previous iteration. This process is stopped when computed similarities do not change anymore (or when the changes are below a given threshold). Then, the system retrieves all documents sorted according to their similarity to the query. The (2) and (3) formulas could be written in form of matrix multiplication way as follow ($n>1$):

$$WT_{n-1}{}^{T}W^{T}/\sum_{l=1..n_t}W(i,l).\sum_{l=1..n_t}W(j,l) \quad (4)$$

$$W^{T}D_{n-1}{}^{T}W/\sum_{l=1..n_d}W(l,i).\sum_{l=1..n_d}W(l,j) \quad (5)$$

(4) reflects the similarity between two documents at n^{th} iteration $S_{d_n}(d_i,d_j)$.

(5) reflects the similarity between two terms at n^{th} iteration $S_{t_n}(t_i,t_j)$.

W is the document-term matrix representing corpus and query, W_{ij} is the weight of term j in document i.

T_n is the term to term square matrix at nth iteration, T^T is the transposed T.

D_n is the document to document square matrix at nth iteration.

n_t is the number of terms

n_d is the number of documents

4 EVALUATION

4.1 Test Collection and Experiments

To evaluate our method, we used the Cranfield[2] corpus. This corpus consists of 1400 documents and 225 queries. A document from that corpus has an average number of 53 terms and a query an average number of 9 terms. 69% of all documents are composed of terms having a term frequency equal to one, 95% of terms' query have a term frequency equal to 1. 19 queries do not provide an answer (no documents retrieved) after Okapi ranking; we choose not to take those queries into account[3].

Document indexing for phase 1 is processed with Lemur Index Builder. Document indexing for phase 2 is based on IR principles: terms are extracted from each document, stop words are removed using the

[2] http://www.dcs.gla.ac.uk/idom/ir_resources/test_collections/cran

[3] removed queries : 15 48 68 71 90 97 109 140 141 142 143 153 192 198 200 202 203 204 211

SMART stop list of 571 English stop words[4], and the remaining terms are stemmed in order to limit the variations in syntax. This is performed through the Snowball algorithm [Porter, 1980]. We filter again stems and suppress stems for which collection frequency and term frequency are equal, this because they do not link two documents, so are not useful for our method. It also permits to reduce the data size. Finally, terms are weighted according to four different weighting schemes from [Salton, 1988]. We ran four different experiments to rerank the Okapi ranking with four different term weighting methods:

- *OkaSim bxx-bxx (OSb)*: terms from the documents (resp. queries) are weighted with 1 if term is present in the document (resp. query) and 0 otherwise.

- *OkaSim txx-txx* (*OStf*): terms from the documents (resp. queries) are weighted with their term frequency i.e. the number of times the term occurs in the document (resp. query).

- *OkaSim tfx-txx (OStf.idf)*: terms from documents are weighted using tf.idf (multiply original tf factor by an inverse collection frequency factor) and terms from queries are weighted by their term frequency.

- *OkaSim tfc-nfx (OStfc-nfx)*: Namely the "Best fully weighted system" in which terms are weighted using tf.idf and a cosine normalization. Terms from queries are weighted using augmented normalized term frequency (tf factor normalized by maximum tf, and further normalized to lie between 0.5 and 1.0).

4.2 Evaluation Criteria

To evaluate our method we consider both MAP and precision at top n retrieved document.

The mean average precision (MAP) is used for assessing the accuracy of retrieval engines. It measures the average of precision computed after each relevant document is retrieved. MAP is defined as follows:

$$\text{MAP} = \frac{1}{n}\sum_{i=1}^{N} p(i) * R(i)$$

Where *N* is the collection size: the total number of the documents

n is the number of relevant documents retrieved

R(i) = 1 if document *i* is relevant and *R(i)* = 0 if document *i* isn't relevant

[4] Smart's English stoplist: ftp://ftp.cs.cornell.edu/pub/smart/english.stop

$p(i) = \frac{|Pert_i|}{i}$ with $Pert_i$ as the set of relevant documents after the i^{th} document is retrieved.

Recall is the proportion of relevant document retrieved in regard of all relevant documents existing in the given collection.

We'll use Precision-Recall curves to show how behave precision for different recall points (i.e. when 0, 10%, 20% … 100% of documents are retrieved).

We use the rank of relevant documents to analyze their behavior in our system. In order to compare OkaSim runs to Okapi, we represent the average rank's evolution of relevant documents over all documents.

5 RESULTS

Table 1: Average MAP and average precision when 10 documents returned for the 5 methods.

Tests	MAP	Gain %	p@10	Gain %
Okapi	0,1156		0,0956	
OSb	0,1868	61	0,1582	65
OStf	0,2343	102	0,1951	104
OStf.idf	0,2608	125	0,2165	126
OStfc-nfx	0,2627	127	0,2131	122

We can see at first glance that our method clearly enhances the original results of Okapi method. That is a case for all different term weighting. The second positive result is that our method is sensitive to term weighting. The *bxx-bxx* weighting already overcome the Okapi method. This suggests that pure structural information can be useful for IRS. Indeed, the only use of a term's presence to relate documents can already bring relevant documents closer to the top of the ranking, as we can see with the precision when 10 documents are returned. The *txx-txx* results show that the use of term frequency instead of simple term's presence improves the Okapi's results by 100%, and *bxx-bxx* results by 25%. A better knowledge on the term's importance seems useful to our method, it participates to the quality of links between documents and terms, and thus, to the quality of the analysis we can process on the corpus structure. The MAP result presents the *tfc-nfx* method as the best weighting for our system accuracy. If we take into account the precision when 10 documents are retrieved, the tf.idf weighting seems a better compromise. For precision when 10 documents are retrieved, *OkaSim bxx-bxx* do better

or equal for 174 queries over the 206, *OkaSim tfx-txx* do better or equal for 194 queries over the 206. Queries n° 34, 52, 122, 186, 191, 199, 202 and 204 are never improved with our methods.

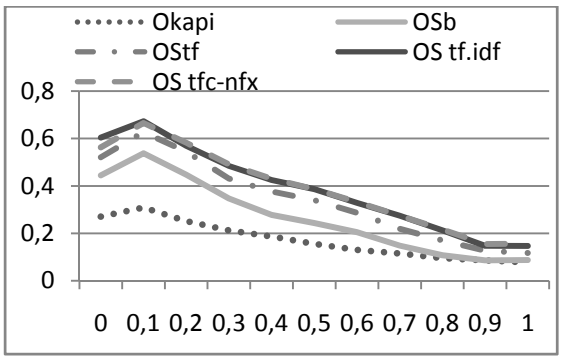

Figure 3: Precision/Recall curves for the 5 methods.

The figure above show precision at different recall points, it permits us to see how evolves precision when recall varies from 0,1 to 1.0. If we look at the precision/recall curve we can see that all curves have the same shape, methods become closer when recall is near to 1. We can notice the same method ranking as for MAP and precision; the tf.idf weighting seems to be the best weighting when recall is under 0, 1. We also can see that *the tfc-nfx* and *tfx-txx* (tf.idf) curves are quasi identical.

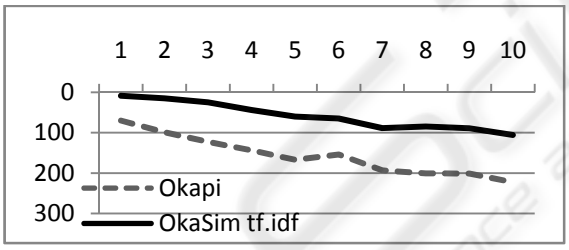

Figure 4: Average rank of first 10 documents returned for Okapi and OkaSim using tf.idf.

On Cranfield corpus, the 206 chosen queries have an average number of relevant documents equal to 8 (between 2 and 29). Okapi return an average of 486 documents per queries in phase 1. Our aim when reranking is to ensure the rank of relevant documents. On Figure3 we draw the average rank (y=axis) for the 10 first relevant documents returned (x=axis) for all queries to compare original okapi and OkaSim using *tf.idf*. We show that our method has clearly ensured the relevant document ranking. The three first returned documents ensure their ranking about 75 places, and the 8th, 9th, 10th returned document about 200 places.

6 CONCLUSIONS

The re-ranking method we propose works well with the Cranfield corpus and Okapi BM25 measure. The power of our method is to retrieve documents indirectly related to the query; this is performed through the use of the structural similarity that acknowledges the relationship between documents, between documents and words as well as between words.

The main limitation of our method is its high computational complexity: $\theta = \max(d,t)^3$, t number of terms and d number of documents. As a consequence it cannot be used in a real-time retrieval process, and the use of such method on bigger corpora will certainly requires optimization techniques.

In the meantime, it certainly will be interesting to conduct more testing and assessment that involve different other corpus. It will also be interesting to use SimRank in combination with other ranking measures different from Okapi. We intend to do this in the context of TREC7 ad-hoc task[5]. We plan to use runs submitted to different past TREC tracks to see if our method will enhance the inputted rankings.

REFERENCES

R. K. Belew, 1989. Adaptive information retrieval. In *Proceedings of the Twelfth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 11-20, Cambridge, MA, June 25-28.

Berge, Claude, 1958. *Théorie des graphes et ses applications* -Paris : Dunod.

Burgess, C., Livesay, K., & Lund, K. 1998. *Explorations in context space: Words, sentences, discourse.* Discourse Processes, 25, 211-257.

Blondel, V.D, et al., 2004. Measure of Similarity between Graph vertices: Application to synonym extraction and web searching, *SIAM* Rev. 46(4):647-666.

Champclaux Y., Dkaki T., Mothe J. 2007. Utilisation des similarités structurelles pour l'évaluation de la pertinence en Recherche d'information. Dans: *Colloque VSST 2007, Marrakech* (Maroc)

Champclaux Y., Dkaki T., Mothe J. 2008. Enhancing high precision using structural similarities. Dans : *IADIS International Conference WWW/Internet*, *Freiburg, Germany*, *13-OCT-08-15-OCT-08*, IADIS, p. 494-498.

S. Deerwester, et al., 1990. Indexing by Latent Semantic Analysis, in *Journal of the Society for Information Science*, 41, p. 391-407.

[5] http://trec.nist.gov

Euler, 1736, *Solutio problematis ad geometriam situs pertinentis, Commetarii Academiae Scientiarum Imperialis Petropolitanae* 8(1736), 128-140.

Forbus K D Gentner, D &Law, K. 1995. MAC/FAC A model of similarity-based retrieval. *Cognitive Science*, 19, 141-205.

Halford, G. S., 1992. Analogical reasoning and conceptual complexity in cognitive development. *Human Development*, 35, 193-217.

Hattori, et al., 2003. Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways, J. Am. *Chem. Soc*., 125(39):11853–11865.

Heymans M., Singh A. K., 2003. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. *Bioinformatics* 19, Suppl. 1, 138--146. 1.

Jones, S. S. and Smith, L. B., 1993. The place of perception in children's concepts. *Cognitive Development*, 8, 113-139.

Kleinberg, J.M., 1999. Authoritative Sources in a hyperlinked environnement, *Journal of the ACM*, 46(5):604-632.

Landauer, T. K., Foltz, P. W., & Laham, D. 1998. An introduction to latent semantic analysis. *Discourse Processes*, 25, 259-284.

D.L. Medin, R.L. Goldstone, and D. Gentner, 1990 'Similarity involving attributes and relations: Judgments of similarity and difference are not inverses', *Psychological Science*, 1(1): 64-69.

Milgram, S., 1967. The Small world problem. *Psychology today* 1(61).

Mothe, J. 1994. Search mechanisms using a neural network-Comparison with the vector space model. *4th RIAO Intelligent Multimedia Information Retrieval Systems and Management*, Vol.1, pages 275-294, New York.

Porter, M.F., 1980. An algorithm for suffix stripping, *Program,* vol. 14, no 3, p.130-137.

Robertson S. E., Walker S., Jones S., Hancock-Beaulieu M., and Gatford M.. 1994. Okapi at TREC-3. In *Proceedings of the Third Text REtrieval Conference* Gaithersburg, USA.

Rohde, D. L. T., Gonnerman, L. M., & Plaut, D. C. 2004. An improved model of semantic similarity based on lexical co-occurrence. *Cognitive Science. (submitted).*

Salton G., Wong A.; Yang, 1975. Vector-Space model for automatic indexing. *Communication of the ACM* 18(11):613-620.

Salton, Chris Buckley; 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manage. 24*(5): 513-523.

SpärckJones, Karen, S.Walker,and Stephen E.Robertson 2000. A probabilistic model of information retrieval: Development and comparative experiments. *Information Processing and Management* pp.779–808, 809–840.

Watts, D.J., 1999. *Small Worlds*, Princetown University Press.