Performance of information retrieval models using term co-occurrences

G. Desjardins[1], R. Godin[1] & R. Proulx[2]
[1]*Department of Computer Science, University of Quebec in Montreal, Canada*
[2]*Department of Psychology, University of Quebec in Montreal, Canada*

Abstract

Many advanced models have been developed for information retrieval in recent years. These models are built on various artificial intelligence paradigms to improve the precision of the retrieval. Most of them exploit some form of term co-occurrences to improve retrieval quality. In this paper, we compare the retrieval performance of five of these models: the Extended Boolean model, the Generalized Vector Space model, the Frequent Set model, the Rough Set model and a Genetic-Based model. These models are tested on three sub-collections from TREC (Text REtrieval Conference). We analyze the specificity of the models regarding the form of co-occurrences introduced and report on the retrieval performance and the scalability of each model.
Keywords: text mining, information retrieval, co-occurrences, extended Boolean, generalized vector space, frequent set, rough set, genetic algorithm.

1 Introduction

Term co-occurrences embed major correlation information among the documents of collections. This information can be used to improve the precision at the core level of the retrieval engines. Many models try to capture this information and incorporate it to their output representation in order to increase the effectiveness of the retrieval engine.

For this research, we have selected five retrieval models that exploit term co-occurrences: the Extended Boolean model, the Generalized Vector Space model, the Frequent Set model, the Rough Set model and a Genetic-Based model [1–5]. The next section reviews the principles of each model. Section 3 describes the

WIT Transactions on Information and Communication Technologies, Vol 38, © 2007 WIT Press
www.witpress.com, ISSN 1743-3517 (on-line)
doi:10.2495/DATA070181

environment set up in terms of the collections and the metrics used. We report on the retrieval performance and the scalability of each model in sections 4 and 5, respectively. The last section concludes on the applicability and suggests directions for future research.

2 Review and implementation of the models

This section reviews the principles of each model and analyzse their co-occurrences selection process.

2.1 Extended Boolean Model

The Extended Boolean model was introduced in [1] to overcome the inability of the Boolean model to rank documents. The basic idea is to graduate a conjunction of two terms by the Euclidian distance to the most desirable point where both terms are included in the document. Similarly, a disjunction is graduated by the distance away from the least desirable point where none of the terms is included in the document. Therefore, any logical combination of terms can be evaluated for each document of the collection and ordered in decreasing magnitude of similarities. The similarity between a query q and a document d is calculated using equation (1) for a conjunction and equation (2) for a disjunction.

$$sim(q_{\wedge}, d) = 1 - \left(\frac{\left(1 - w_{t_1}\right)^p + \left(1 - w_{t_2}\right)^p + \ldots + \left(1 - w_{t_n}\right)^p}{n} \right)^{\frac{1}{p}} = 1 - \left(\frac{1}{n} \sum_{i=1}^{n} \left(1 - w_{t_i}\right)^p \right)^{\frac{1}{p}} \quad (1)$$

$$sim(q_{\vee}, d) = \left(\frac{w_{t1}^{\ p} + w_{t2}^{\ p} + \ldots + w_{tn}^{\ p}}{n} \right)^{\frac{1}{p}} = \left(\frac{1}{n} \sum_{i=1}^{n} w^p_{t_i} \right)^{\frac{1}{p}} \quad (2)$$

w_{ti} is the weight of the term t_i within document d;
p is an empirical parameter ($1 \le p < \infty$);
n is the total number of index terms.

Co-occurrence information is accounted for in the Boolean model only from the perspective of the query. A conjunction means to retrieve only documents where the terms are fully correlated. A disjunction means to retrieve all documents containing some of the terms, whether they are correlated or not. With the parameter p, the extended Boolean model introduces flexibility as to how strong the correlations should be. Using $p = 1$ turns the model back to strict Boolean where co-occurring terms are either fully accounted or not accounted. Using a value approaching infinity turns the model toward a pure Vector Space model where co-occurrences are not considered. Using an intermediate value for p reduces the stiffness of the conjunctions and favors the correlated terms for disjunctions. The authors obtained the best retrieval performances with $1 \le p \le 5$.

2.2 Generalized Vector Space Model

The Generalized Vector Space model was introduced in [2] to account for all combinations of terms contained in the documents. This representation maps the documents from the n-dimensional space to a 2^n-dimensional space where each dimension stands for a specific combination of terms called a *minterm*. The first n minterms represent individual terms and are mutually orthogonal. The remaining minterms introduce the co-occurrence information. They account for all orders of co-occurrences, i.e. combinations of two terms, three terms, …, n terms. A document is represented by the subset of the minterms that covers all combinations of index terms contained in the document.

An individual term can be represented by the normal disjunction of all minterms where the term is active. This representation allows the document to be represented by a vector of index terms where each term accounts for its own and all co-occurrences with it. The term vector is calculated with equation (3).

$$\vec{t}_i = \frac{\sum_r c_{i,r}\vec{m}_r}{\sqrt{\sum_r c_{i,r}^2}} \tag{3}$$

r iterates over all minterms where t_i is active;
$c_{i,r}$ is the correlation factor between the term t_i and all other terms;
$\vec{m}_r$ is the orthogonal representation of the minterm m_r (see Fig. 1).

The correlation factor $c_{i,r}$ for the term t_i associated to the documents represented by the specific combination of terms m_r is express by equation (4).

$$c_{i,r} = \sum_{(j|d_j \leftarrow m_r)} w_{i,j} \tag{4}$$

j iterates over all documents d_j containing the minterm m_r;
$w_{i,j}$ is the weight of the term t_i within document d_j.

Once all correlation factors are calculated, the representation of the documents can be translated from the n-dimensional space to the 2^n-dimensional space, using equations (5) and (6).

$$\vec{d}_j = \bigoplus_{i=1}^{n} w_{ji}\vec{t}_i \tag{5}$$

$$\vec{t}_r \oplus \vec{t}_s = \sum_{m_k \in \{m\}^r \cup \{m\}^s} \max(c_{k,r}, c_{k,s})\vec{m}_k \tag{6}$$

$\vec{d}_j$ is a specific document vector to be translated;
$\vec{t}_i$ is the normalized sum of all active minterms for the specific term t_i;
$\{m\}^r$ is the set of all active minterms for the term t_r.

Thus, the model accounts for all co-occurrences by computing each as the maximum of the correlation factors between all possible combinations of terms.

2.3 Frequent Set model

The Frequent Set model introduced in [3] uses a data mining technique to find the most frequent term sets in a collection of documents. The technique is based on a frequent closed set mining algorithm, which builds sets of n terms from the sets of (n-1) terms. For each order, the process selects the co-occurrence term sets that meet a minimum frequency support expressed as a number of documents. The process ends when no higher order set meets the criterion.

The documents and the queries are then translated into the new space representation spanned by the co-occurrence term sets. A document is indexed by a term set only if it contains all terms of the set. There are two versions for the query representation. In the first version, a query is represented by a term set if all terms of the set appear in the query. In the second version, a query is represented by a term set if at least one of the terms appears in the query.

This model replaces the atomic term representation with a selection of co-occurrences based on their document frequency.

2.4 Rough Set model

The theory of rough sets is applied in information retrieval to build an alternative representation for the documents [4]. First, the terms of the collection must be clustered into meaningful sets of terms called the concepts. Then, the document and the query representations are translated into concept representations. For each, two rough sets are built as approximate upper and lower limit sets of terms (U, L). The lower limit is defined as the subset of concepts for which all the terms appear in the document. The upper limit is defined as the subset of concepts for which at least one term of the concept appears in the document.

The model defines a variety of retrieval strategies based on different combinations of test operators (equals, includes, overlaps) between the limit sets. The similarities are evaluated with equations (7), (8) and (9).

$$\mathrm{sim}(q, d) = \mathrm{simL}(q, d) + \mathrm{simU}(q, d) \tag{7}$$

$$\mathrm{simL}(q, d) = |L(q) \cap L(d)| \ / \ |L(q) \cup L(d)| \tag{8}$$

$$\mathrm{simU}(q, d) = |U(q) \cap U(d)| \ / \ |U(q) \cup U(d)| \tag{9}$$

The clustering of the terms into concepts could be accomplished by any manual or automatic process. We have opted for the implementation of the frequent closed set algorithm in order to compare the results with the Frequent Set model. Within this implementation, the upper and the lower limit sets can be viewed as a second selection process that further specializes the representations.

2.5 Genetic-Based model

A genetic algorithm is developed in [6] to optimize the description of the documents in relation to the query terms. The objective function is based on the Jaccard score, which uses the user relevancies to quantify the fitness of the

reformulated descriptions. The whole process aims at tuning the weights of the terms within each document in order to agree with past user judgments. Other models using genetic algorithms have been developed since then [7, 8].

Desjardins *et al* built upon this work to develop a Genetic-Based model where the genetic algorithm is used to find a number of optimal co-occurrences of terms within the documents [5]. In that model, the objective function (equation (10)) is based on the similarity function. The fitness of a correlated terms set is quantified by the distance between pairs of documents instead of using past queries. The representation of documents is similar to the one adopted in the Frequent Set model. Each document is represented by a vector of term sets, including the original atomic terms as single term sets and the sets of correlated terms discovered by the genetic algorithm.

$$F(P) = \sum_{c} F(c) = \sum_{c} \sum_{d} sf_{c,d} \times ids_{c} \tag{10}$$

$F(c)$ is the fitness of chromosome c
$w_{c,d}$ is the weight of chromosome c in document d
$sf_{c,d}$ is the frequency of the term set c (chromosome c) in document d
ids_c is the inverse document frequency of the term set c

This process selects co-occurrences based on their fitness according to the objective function. In comparison, the Frequent Set model selects co-occurrences based only on their document frequencies.

3 Environment set-up

Two different tests were conducted on the retrieval models. The first test evaluated the retrieval performances of the models whereas the second test was built specifically to evaluate the scalability of the models.

For the performance test, the precisions are evaluated at the eleven recall levels (0%, 10%, …, 100%) using the standard TREC procedure. The retrieval performance of each model is compared to the results of a basic vector space model [9], for which no co-occurrence information is considered. Thus, the results highlight the contribution of the co-occurrences selection process introduced in each model. Three collections have been extracted from TREC ('Text REtrieval Conference') to assess the retrieval performances (see Table 1).

For the scalability test, five subsets of 2 000 documents each have been extracted from the FT943 collection. These sets were cumulated to build five progressive volume collections (see Table 2).

Table 1: Collection statistics for the performance test.

Collection	CR93H	FT943	ZF109
number of documents	**12 320**	**17 109**	**22 709**
number of terms	56 892	71 011	72 983
number of relevant documents	665	273	790
number of queries	21	15	19
% of relevant documents	5,40%	1,60%	3,48%

Table 2: Collection statistics for the scalability test.

	Cumulative collections				
number of documents	**2 000**	**4 000**	**6 000**	**8 000**	**10 000**
number of terms	25 838	36 363	44 039	49 013	55 447
number of relevant documents	20	45	64	89	112
number of queries	7	7	7	7	7
% of relevant documents	1,00%	1,13%	1,07%	1,11%	1,12%

4 Retrieval performance

The following diagrams show the precision-recall curves for the five models (Fig. 2.) identified as GV (Generalized Vector space), XB (eXtended Boolean), FS (Frequent Set), RS (Rough Set) and GA (Genetic Algorithm), as compared to the VS (Vector Space model).

The precisions are generally converging as the level of recall approaches 100%. The differences in precision are more significant at the first few levels of recall (< 30%).

The results for the XB model are outstanding in the first collection but not in the two others. The FS and RS models are outstanding in the first and the last collections, but only a little above VS results in the FT943 collection. The GV model outperforms all models in the second collection and obtained above VS results in the last collection, but below VS results in the CR93H collection. The GA model shows the same curve as the VS model in all collections.

These results confirm the brittleness of the retrieval processes across different collections, a general conclusion reported in the literature.

The GV model takes into account all possible combinations of terms. This costly process does not guarantee a better retrieval performance, as observed with the results in the last collection.

The XB model focuses on the co-occurrences from the query vector. This strategy seems to give significant results, at least in the first collection. As opposed to the GV model, the strategy avoids selecting co-occurrences that would reduce the quality of the retrieval. A similar strategy was recently adopted for the FS model where only the first order of co-occurrences is selected from the collection [10]. Higher orders of co-occurrences are selected from the query terms at run time. This strategy could apply to many models.

The FS model exhibits more stable improvements over all collections. This model selects co-occurrences based solely on the inverse document frequencies. The results of the RS model fall a little under the results of the FS model. Both models use the same co-occurrences selection process. The RS model further constrains the co-occurrences in the vector representation, which influences the similarity calculations. This strategy does not seem to improve the retrieval results, as this model did not outperform the FS model.

The more specific co-occurrences selection implemented by the GA model did not produce a better retrieval performance. During the experiment, the Genetic-Based model focused on a few co-occurrences that seemed to be

connected to only one major theme of the collection. The poor coverage of the collection could be responsible for the low precisions observed. This finding suggests considering a recursive approach where each iteration discovers the significant co-occurrences for a specific subject. The collection would then be better covered by many subclasses of co-occurrences. Such a local fit process could also be adapted to other models as well.

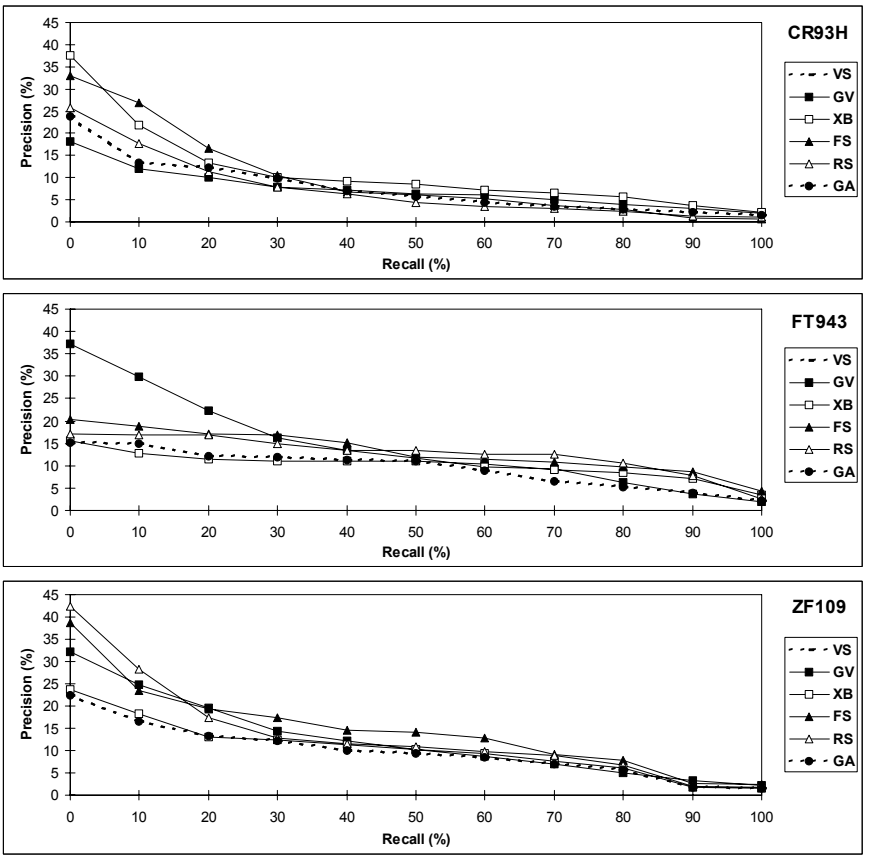

Figure 1: Precision-Recall curves for CR93H, FT943 and ZF109 collections.

5 Scalability

In the following figure, the progressions of the processing costs are reported on a logarithmic scale to better visualize the tendencies as the size of the collection increases.

The curves indicate a near linear progression for the XB model and the GA model. The GV model exhibits an exponential progression in the magnitude of 2. The two set theoretic models, FS and RS, show exponential progressions in magnitudes higher than 2. This cost progression is incured by the frequent set algorithm. These figures agree with the scalability analysis pictured in Table 3.

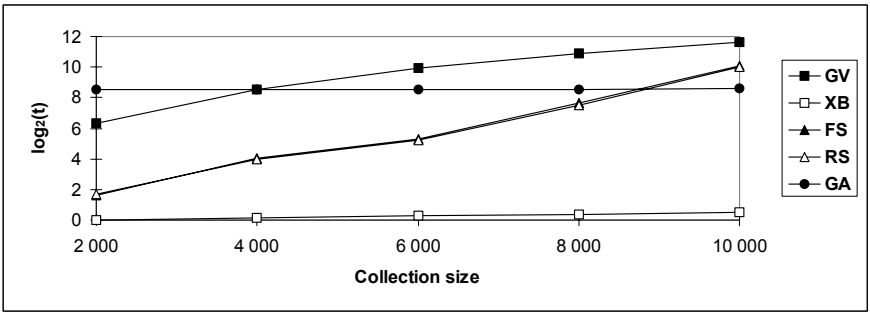

Figure 2: Progression of the processing cost (*scale: log_2(cost in minutes)*).

Table 3: Scalability summary.

Model	Compression ratio	Scalability
(GV) Generalized Vector	4,1 X	$O(n + mk^2)$ (1) (2)
(XB) Extended Boolean	none	$O(m + n)$ (1)
(FS) Frequent Set	10,2 X	$O(m \cdot 2^k)$ (1) (3)
(RS) Rough Set	10,2 X	$O(m \cdot 2^k)$ (1) (3)
(GA) Genetic-Based	0,985 X	$O(m + n + k)$ (1) (4)
(1) m = number of documents; n = number of terms;		
(2) k = number of minterms; $k \leq m$; (11 491 < k < 22 576)		
(3) k = number of co-occurrence orders; $k << n$; ($k \leq 12$ for a support < ¼ %)		
(4) k = number of chromosomes; $k << n$; (100 < k < 1000)		

The curse of dimensionality in information retrieval emphasises the importance of reducing the number of dimensions in the output representation. Compressing the space of representation is also appealing because it speeds up any further retrieval, assuming a reasonably low cost for the compression.

Table 3 outlines the compression ratio and the order of progression of the processing cost for each model. The GV model compresses the dimensions by a factor of 4,1. This seems surprising since the model translates the representation from n dimensions to 2^n dimensions. Noting that only a small portion of all possible minterms effectively appears in the collections brings the number of dimensions to a much more tractable size. As an example, this number varied from 11 491 to 22 576 in the collections used here. The XB model does no compression at all since it uses the basic vector space representation. With a support of 20 to 40 documents, the frequent set algorithm produced from 5 604 to 8 672 co-occurrences sets, resulting in a compression factor of 10,2.

The GA model added 1 000 co-occurrences sets to the basic vector space representation, which has no significant effect on the number of dimensions.

From the analysis of the algorithms implemented, we can derive the progression order of the processing cost (see the scalability column in Table 3). The number of iterations varies linearly with the number of documents and the number of terms for the XB and the GA models. It varies with the square of the

number of documents for the GV model, which makes it more difficult to extend to very large collections. The two set theoretic models were implemented with a basic close frequent set algorithm that scales up with difficulty. However, faster close frequent set algorithms are under development [11, 12].

6 Conclusion and future work

In this research, we have compared five models that use different approaches to exploit the term co-occurrences. The retrieval performance of the models is evaluated on three significant collections extracted from TREC and compared to the results of a basic vector space model used as the baseline. Their scalability is evaluated using a progressive volume collection.

The results highlight the difficulty to build a unique classification algorithm that would grasp the essential information from co-occurrences for accurate retrieval on general collections. Accounting for all co-occurrences does not always improve the retrieval effectiveness. Therefore, a selection process is needed. Among the models experimented that incorporate a selection process, only the Frequent Set model exhibits improvement on all three collections.

The Generalized Vector space model offers a compression ratio of 4,1 and the two set theoretic models offer a compression ratio of 10,2. Despite this appealing compression, these three models are difficult to scale up to very large collections. However, faster algorithms are continually under development.

Future research should consider processing the selection of co-occurrences at query time, at least partly. This would decrease the costs by avoiding indexing too many combinations from the collection of documents. As a second benefit, it should improve the retrieval by focusing on the query terms. The experiment with the Genetic-Based model suggests adopting an iterative approach to classify different portions of the collection using an incremental procedure. Such a locally fit representation could be implemented in many models.

References

[1] Salton, G., Fox, E.A. and Wu, H., Extended Boolean Information Retrieval. *Communications of the ACM*, Vol. 26, No. 11, pp. 1022-1036, 1983

[2] Wong, S.K.M., Ziarko, W. and Wong, P.C.N., Generalized Vector Space Model in Information Retrieval. *Proceedings of the 8th annual international ACM SIGIR Conference on Research and Development in Information Retrieval,* pp. 18-25, 1985.

[3] Pôssas, B., Ziviani, N., Meira, W. and Ribeiro-Neto, B., Set-Based Model: A New Approach for Information Retrieval. *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval,* pp. 230-237, 2002.

[4] Das-Gupta, P., Rough Sets and Information Retrieval. *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 567-581, 1988.

[5] Desjardins, G., Godin, R. and Proulx, R., A Genetic Algorithm for Text Mining. *Proceedings of the 6th International Conference on Data Mining, Text Mining and their Business Applications*, Vol. 35, pp. 133-142, 2005.

[6] Gordon, M., Probabilistic and Genetic Algorithms for Document Retrieval. *Communications of the ACM*, Vol. 31, No.10, pp. 1208-1218, 1988.

[7] Desjardins, G. and Godin, R., Combining Relevance Feedback and Genetic Algorithm in an Internet Information Filtering Engine. *Proceedings of the 6th RIAO Content-Based Multimedia Information Access*, Vol. 2, pp. 1676-1685, 2000.

[8] Martin-Bautista, M.J., Vila, M-A. and Larsen, H.L., Fuzzy Genes: Improving the Effectiveness of Information Retrieval, 2000, Department of Computer Science and Artificial Intelligence, Granada University, Computer Science Department, Roskilde University.

[9] Salton, G., The SMART Retrieval System – Expirements in Automatic Document Processing, 1971, Prentice Hall inc.

[10] Pôssas, B., Ziviani, N, Meira, W and Ribeiro-Neto, B., Set-Based Vector Model: An Efficient Approach for Correlation-Based Ranking. *Proceedings of the ACM Transactions on Information Systems*, Vol. 23, No. 4, pp. 397-429, 2005.

[11] Nehmé, K., Valtchev, P., Rouane, M. H. and Godin, R., On Computing the Minimal Generator Family for Concept Lattices and Icebergs. *Proceedings of the International Conference on Formal Concept Analysis*, Vol. 3403, pp. 192-207, 2005.

[12] Lucchese, C., Orlando, S. and Perego, R., Fast and Memory Efficient Mining of Frequent Closed Itemsets. *IEEE Transactions on Knowledge and Data Engineering,* Vol. 18, No. 1, pp. 21-36, 2006.