

A genetic algorithm for text mining

G. Desjardins¹, R. Godin¹ & R. Proulx²

¹*Department of Computer Science, University of Quebec in Montreal, Canada*

²*Department of Psychology, University of Quebec in Montreal, Canada*

Abstract

Text workers should find ways of representing huge amounts of text in a more compact form. Textual documents can be represented by concepts. One way to define the concepts is by the terms, keywords extracted from the textual documents and cleaned by several processes like stopwords and stemming. Using the frequencies of the terms, one can quantify the relations between documents or portions of text. These relations can serve many applications, like information retrieval or automatic text classification.

Another way to define the concepts is by the sets of correlated terms rather than by raw terms. Correlated terms usually have a more specific meaning. Finding meaningful concepts within a huge collection of corpuses in a reasonable timeframe is a difficult task to accomplish.

This paper describes a new text mining process to uncover interesting term correlations. The process uses a genetic algorithm to cope with the combinatorial explosion of the term sets. The genetic algorithm identifies combinations of terms that optimize an objective function, which is the cornerstone of the process. We have tested a function designed to optimize the discriminating power of the term sets.

The genetic model was tested on a TREC sub-collection. The parameters were set to discover a thousand combinations of correlated terms. These sets of terms were further added to the basic index and applied to the information retrieval problem. The experiment revealed that the augmented index was unable to improve the effectiveness of the retrieval, when compared with the vector space model.

Keywords: genetic algorithm, co-occurrences, information retrieval, text mining.



1 Background

Applying genetic algorithms for text mining is not new, specifically in the search for better document descriptions. When the final goal is information retrieval, researchers define a GA objective function based on the retrieval performance of past queries [2, 5, 7, 8, 15]. This design gives good results as long as the new queries are within the same domain of knowledge. Our work is an attempt to generalize the document descriptions beyond the specificity of one domain. To accomplish that, one cannot use the results of past queries. Therefore, we designed a genetic algorithm that searches for meaningful co-occurrences of terms within the collection of documents alone. The use of term co-occurrences has been successful for semi-automatic thesaurus building and the like; it has met with mixed results when applied to the retrieval problem [1, 3, 4, 6, 11, 13, 14]. In this paper, we present a new way to discover term co-occurrences with the use of a genetic algorithm. We then apply the results to the information retrieval problem.

Since its first proposal by Holland [10], genetic algorithms have been used by many researchers in a variety of domain applications as a mean of optimizing solutions for non-trivial problems. Genetic algorithms borrow their process from the Darwin natural process of survival. The genetic process changes the individuals over generations. The environment selects the most fitted individuals to survive and allow them to reproduce in order to perpetuate the strong genetic codes. Recombining the genes of the individuals makes the changes to the overall population. New generations either augment the initial population or replace individuals.

When adapting the genetic theory to the text categorization problem, the documents represented by a vector of terms become the chromosomes of the population. Each term into a vector becomes a gene. The categorization problem turns into finding the best set of terms to represent each document of the collection, with respect to a specific goal, which might be, for example, maximizing the distances between the categories. The goal is modeled as an objective function to optimize, which is termed as the fitness function in the genetic domain. The fitness function plays the role of the natural selection. New individuals are generated by exchanging the genes at random between the most fitted sets of terms according to the fitness function. This guided-random process continues until the fitness of the population stops increasing, Goldberg [9].

The following section describes how the genetic model is adapted to the co-occurrences finding problem. Section 3 describes the general retrieval problem. Section 4 reports the results on mining the texts with the genetic algorithm. Section 5 reports on the use of the genetic co-occurrences to improve the effectiveness of the information retrieval process.

2 The genetic model

In text analysis, documents are represented by a set of index terms. These terms are words extracted from the documents and cleaned by several processes. Two



of the most used processes are the stopwords and the stemming. The stopwords process eliminates the insignificant words like 'the' and 'a'. The stemming process extracts the root of the words in order to account for a single term different words bearing the same morpheme. For example, the words 'ski', 'skies' and 'skiing' would be counted as three occurrences of the same stem 'ski'. This process greatly influences the term co-occurrences in a collection of documents.

As we already mentioned, our goal is not to replace the basic term representations of the documents by better representations but rather to enrich the actual representations with the introduction of co-occurrent terms. Our genetic model is specifically designed to discover the best sets of co-occurrent terms. In this model, a chromosome stands for a specific combination of terms. Each gene represents a term of the combination. The population of chromosomes aims to become, through the genetic cycles, the best sets of co-occurrent terms across the entire collection of documents. This goal is accomplished through the optimization of an objective function that measures the fitness of the chromosomes. The overall fitness of the population is the sum of the individual fitness of the chromosomes. The genetic cycle is as follows. (Figure 1)

1. An initial set of solutions is established either at random or by other means from some of the co-occurrences into the documents. Other means include the selection of the most frequent sets of terms, which represent a good starting solution.
2. Then the fitness of the current population of solutions is evaluated using the objective function. The stopping criteria are tested. As a general criterion, the genetic process is stopped when the overall fitness does not increase over a few iterations.

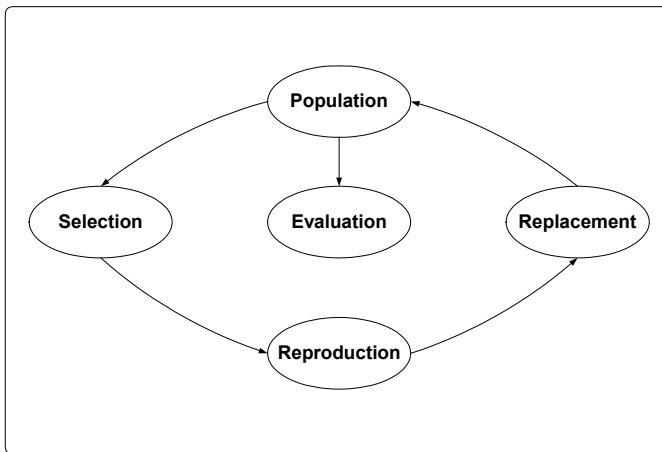


Figure 1: Genetic cycle.

3. Two of the highest fit individuals are selected at random for reproduction. This process generates two new individuals by modifying the parent's genetic codes through the crossover and the mutation operators. (Figure 2)
4. The two new individuals replace two of the lowest fit individuals and the iterative process buckles up from step 2.

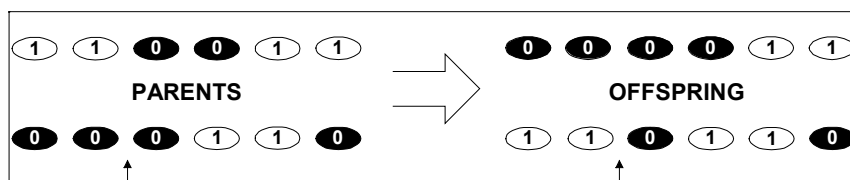


Figure 2: One-point genetic crossover.

The genetic algorithm generates new solutions by recombining the genes of the current best solutions. This is accomplished through the crossover and the mutation operators. The crossover operator exchanges part of the genetic codes between the parents. On a one-point crossover, the crossing point is selected at random and the genes from one side of the chromosomes are exchanged. Then a mutation is operated on one gene of one of the two new chromosomes. The mutation is usually only operated at a low frequency (with probability $\leq 0.1\%$). The mutation operation is justified by the need to explore the space of solutions.

In our model, the chromosomes are defined with a maximum length of 20 genes, some of which could be empty. With this definition, the number of co-occurrent terms in a solution can vary from 2 to 20. The positions of the genes within the chromosomes are selected at random. The mutation operator will either empty a position occupied by a term or generate a new term on an empty position.

Because the space of solutions is so vast ($2,5 \times 1026$ sets of six terms or less in a corpus of 75 000 terms) and because only a small portion of all combinations exists into the collection, we introduced a hyper mutation rate into the model. The mutation rate will be fixed between 50% and 70%; at least one chromosome will undergo a mutation each generation.

The objective function is the cornerstone of the genetic process. We designed the following fitness function to explore the space of solutions:

$$F(P) = \sum_c F(c) = \sum_c \sum_d w_{c,d} = \sum_c \sum_d sf_{c,d} \times ids_c = \sum_c \sum_i sf_{c,d} \times \log \frac{N}{ds_c}, \text{ where}$$

$F(c)$ is the fitness of chromosome c ;

$w_{c,d}$ is the normalized information unit of chromosome c within document d ;

$sf_{c,d}$ is the frequency of the term set represented by the chromosome c within document d ;

ids_c is the inverse frequency of the term set represented by the chromosome c ;

ds_c is the number of documents containing the specific combination of chromosome c ;

N is the total number of documents in the collection.

The information unit could be either the binary information (1 if the term set is included in the document; 0 otherwise), the frequencies of the term sets or the weights of the term sets. The fitness function has been specifically designed for use with the weights of the term sets. It could also be used with the other information units. This formula aims at maximizing the global weight of the solutions. It follows from the standard discriminating formula used by Salton in the vector space model [12].

3 The retrieval problem

Information retrieval is concerned with the classification processes and the selective recovery of information for the benefit of an information seeker. For the text type of information, the typical scenario consists of indexing a collection of documents with keywords and then matching the index terms with the terms of a user query. A perfect match would fire all relevant documents of the collection and none of the others. These are the recall and the precision principles of the retrieval. When assessing the effectiveness of a retrieval process, the recall is measured by the number of relevant documents retrieved over the total number of relevant documents and the precision is measured by the number of relevant documents retrieved over the total number of documents retrieved. Once the index terms are determined, the matching process is straightforward. The terms vector of the query is compared to the terms vector of the documents using a similarity function. All documents that compares with a predetermined threshold value are retrieved. The most commonly used similarity function is the well-known cosine measure:

$$\text{sim}(q, d_j) = \frac{\sum_{i=1}^n w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^n w_{i,j}^2} \times \sqrt{\sum_{i=1}^n w_{i,q}^2}}, \text{ where}$$

$w_{i,j}$ is the unit information associated with term i in the document d_j ;

$w_{i,q}$ is the unit information associated with term i in the query q ;

n is the number of terms in the query q .

The effectiveness of the retrieval depends on both the quality of the query and the quality of the index terms. For the collection corpus, a good quality index term is a term that has a great discriminating power among the documents. Such a term should index as few documents as possible in order to be discriminating. It should also be a highly frequent term within the documents in order to be significant for the queries. The information unit ‘term frequency $\times$ inverse document frequency’ ($tf \times idf$) introduced by Salton [12] became popular in information retrieval precisely because it follows the quality specifications just stated.

$$w_{i,j} = tf_{i,j} \times idf_i = \frac{freq_{i,j}}{\max_k freq_{k,j}} \times \log \frac{N}{n_i}, \text{ where}$$

$freq_{i,j}$ is the frequency of term i within document j ;



n_i is the number of documents containing term i ;

N is the total number of documents in the collection.

A good query is a set of terms that expresses accurately the information need while being usable within the collection corpus. The last part of this specification is critical for the matching process to be efficient. That is why most research efforts are actually put toward the query improvement. It is also possible to improve the index terms to express more discriminating power. To do so, one would have to explore other unit information formulas or alternate representations for the documents. We chose to go with the later.

The term co-occurrences schema developed within our genetic model can be used to improve the discriminating power of the index terms. Next is the application of the genetic model to the retrieval problem and the resulting performances.

4 Mining the texts

The test collection is a sub-collection of the TREC-6 ad hoc track ('Text REtrieval Conference'). The sub-collection ZF109 contains 22 709 documents taken from the Computer Select disks and has been indexed with 72 983 terms after running the stopwords and the stemming processes. The terms indexing a hundred documents and more have been discarded because of their high document frequency, which make them poor discriminating terms. The remaining terms index an average of 6 documents each. The documents are indexed by 1 to 914 terms each, with an average of 20 terms per document.

The fitness function yielded term co-occurrences spread over 375 documents, which represents about 1.7 % of the collection. If we take a close look at the sets of terms generated (table 1), we can definitely identify many meaningful relationships among the correlated terms. Although, we can't interpret these relations as semantic relations because they are solely constructed from statistical occurrences.

If we look at the first five most fitted chromosomes, we can see that the chromosomes 1, 2 and 4 are the two by two genes decompositions of chromosome 5. We should expect a three correlated terms set to bear more discriminating power than any of its two-terms sub-sets. This is probably the case when considering the inverse document frequency alone. A three-terms set certainly indexes less documents than any of its two-terms sub-sets. But the fitness function uses the weights of the sets, which takes into account the within documents frequencies, in addition to the inverse document frequencies. In the case of chromosome 5, the reduction in document frequencies outbalanced the reduction in the inverse document frequency, resulting in a lower fitness than any of its two-terms component (246 < 250, 297, 351).

There also seems to be noisy relations. For example, the terms 'agha', 'att', 'dept', 'mcc' and 'rand' appeared in many relations without apparent signification. As another example, 'orlean' appeared in many sets of terms. It also co-appeared with 'pittsburgh' in many relations and with 'portland' in many others, but never the three of them nor 'pittsburgh' and 'portland' together.

Again, care should be taken not to consider any set of terms as semantically related. The genetic algorithm, like many other artificial intelligence paradigms, is a mean to uncover only statistical relations. This is why some term sets may appear as unrelated terms. Nevertheless, there exists a strong statistical relation among them. This is analogue to discovering a rule like 'red hair women by sport cars'. There is no relation between the colour of the hair and the buying behaviour, other than a pure statistical relation.

Table 1: Term co-occurrences sample.

Chrom. Id.	Fitness	Chromosome
1	351	inheritance superclass
2	297	inheritance subclass
3	274	bitmap rectangle
4	250	subclass superclass
5	246	inheritance subclass superclass
8	218	queuing synchronization
32	119	inheritance iterative
40	103	declaration identifier inheritance
84	71	interprocess queuing
114	69	granularity occurring
144	63	constrained magnitude
255	58	exponential magnitude
423	53	chinese coordinator gannon orlean portland
440	53	chinese gannon mcc orlean pittsburgh
740	37	citizen nippon
956	20	conditional disjoint implementor induce presley

5 Application to the information retrieval

Introducing the sets of term co-occurrences into the documents representation necessitates a modification to the representation. A document is no longer represented by the vector of its indexing terms but rather by the vector of its indexing sets of terms. In order to enrich the existing representation, the single indexing terms are translated to the new representation into a set of a single term each. The new indexing sets of correlated terms are then added to the documents representation. For example, a document represented by the vector {inheritance, superclass, subclass, bitmap, rectangle} is translated to the following vector:

{ {inheritance}, {superclass}, {subclass}, {bitmap}, {rectangle},
 {inheritance, superclass},
 {inheritance, subclass},
 {subclass, superclass},
 {inheritance, subclass, superclass},
 {bitmap, rectangle} }

The first line is the translation of the original representation. The following lines are the sets of correlated terms generated by the genetic algorithm that are contained within the document. The document representations were all revised and the ' $tf \times idf$ ' factors were recalculated including the sets of multiple terms. The query representations were revised as well. Then the matching between the queries and the documents has been reprocessed using the enriched representations and the usual cosine formula to calculate the similarities. Instead of using a threshold value for firing the documents, all documents were ordered by decreasing value of similarity. This follows the TREC official procedure for evaluating the retrieval effectiveness. The precisions were then interpolated for each query at the standard levels of recall (0%, 10%, ..., 100%) and averaged over all queries of the run.

The graph in figure 3 shows the resulting precisions for the run using the genetic model, along with the results of the classic vector space model. A third curve shows the potential gain one can make by adding the appropriate term co-occurrences. This dotted curve has been obtained by running the retrieval process with the use of the query term co-occurrences that exist within the documents.

It is clear from the graph that the two first curves are the same, meaning that the term co-occurrences found by the genetic process did not improve the retrieval effectiveness. The third curve suggests that some of the term co-occurrences could improve the retrieval, especially at the levels of recall from 20% to 60%. The genetic algorithm did not find these sets of terms. It found co-occurrences from only 375 documents. The relevant documents to the queries under test fell outside these few documents.

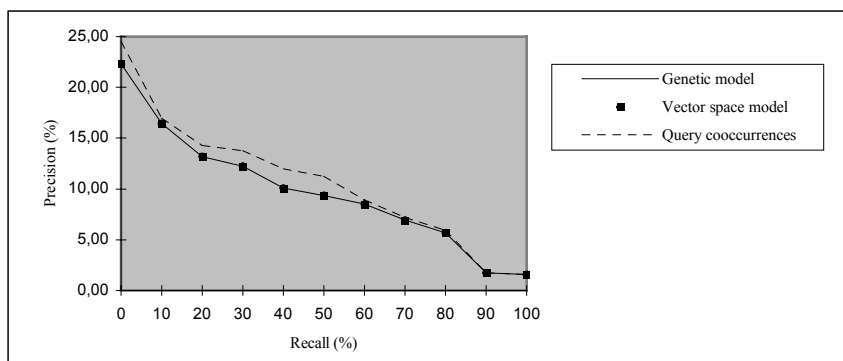


Figure 3: Precision-recall curves.

6 Concluding remarks and future work

In this experiment, we have designed a genetic model to find useful term co-occurrences within a collection of documents. We have defined an objective function to target the discriminating power of the index terms. This function

served as a fitness function, which is the cornerstone of the genetic algorithm. When defining this function, we attempted to target the effectiveness of the information retrieval process. The co-occurrences found by the genetic process did not improve the effectiveness of the retrieval. A number of explanations arose from the analysis of the results. Firstly, the thousand sets of co-occurrent terms indexed only about a few hundreds documents of the collection. Each set certainly have a good inverse document frequency, but some sets are definitely almost redundant, at least regarding the documents they index. Eliminating the redundant sets would better spread the chromosomes over the collection, which would provide better odds for improving the retrieval. Secondly, the discriminating power of the index terms might not be the only key factor toward better retrieval performance. The most useful subsets to improve the retrieval might not be the most discriminating ones, as defined by the ' $tf \times idf$ ' type of information. Thirdly, a poor query formulation already has a significant impact on the retrieval effectiveness. The use of co-occurrences makes it even worse. When testing, this problem could have hidden any potential improvement.

The application of the genetic model to the retrieval problem left some open issues. 1. We must alter the genetic algorithm in order to increase the coverage of the chromosomes over the space of solutions. 2. We must find ways to automatically identify and eliminate the apparent redundancies. 3. A related issue is to decrease the noise caused by apparent insignificant terms. 4. Finally, we have to set up a testing environment with queries that include correlated terms of the collection. Future work will be oriented toward these goals. Also, an in depth study of the cognitive factors involved in judging the relevancy of documents to queries could certainly reveals other key factors to take into account when designing a fitness function.

References

- [1] Byrd, R.J. and Ravin, Y. "*Identifying and Extracting Relations from Text*", in NLDB'99 - 4th International conference on applications of natural language to information systems, Austria, pp. 149-154, 1999.
- [2] Chen, H. "*Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms*", MIS Department, College of Business and Public Administration, University of Arizona, 1994.
- [3] Chen, H. Yim, T., Fye, D. and Schatz, B. "*Automatic Thesaurus Generation for an Electronic Community System*", Journal of the American Society for the Information Science, vol. 46, no. 3, pp. 175-193, 1995.
- [4] Chen, H. Martinez, J., Kirchhoff, A., Ng, T.G. and Schatz, B.R. "*Alleviating Search Uncertainty through Concept Associations*", Journal of the American Society for the Information Science, Special Issue on Management of Imprecision and Uncertainty in Information Retrieval and Database Management Systems, vol. 49, no. 3, pp. 206-216, 1998.
- [5] Desjardins, G. et Godin, R. "*Combining Relevance Feedback and Genetic Algorithm in an Internet Information Filtering Engine*", in 6th



- Proceedings of the RIAO Content-Based Multimedia Information Access, vol. 2, pp. 1676-1685, 2000.
- [6] Ding, Y., Engels, R. "*IR and AI: Using Co-occurrence Theory to Generate Lightweight Ontologies*", 12th International Conference on Database and Expert Systems Applications, vol. 2, pp. 1676-1685, 2001.
 - [7] Ferguson, S. "*BEAGLE: A Genetic Algorithm for Information Filter Profile Creation*", University of Alabama, 1995.
 - [8] Gordon, M. "*Probabilistic and Genetic Algorithms for Document Retrieval*", Communications of the ACM, Vol. 31, No.10, pp. 1208-1218, 1988.
 - [9] Goldberg, D.E. "*Genetic Algorithms in Search, Optimization & Machine Learning*", Addison-Wesley Publishing, ISBN 0-201-15767-5, 1989.
 - [10] Holland, J.H. "*Adaptation in Natural and Artificial Systems*", University of Michigan Press, ISBN 0-472-08460-7, 1975.
 - [11] Peat, H.J. and Willett, P. "*The Limitation of Term Co-occurrence Data for Query Expansion in Document Retrieval Systems*", Journal of the American Society for the Information Science, vol. 42, no. 5, pp. 378-383, 1991.
 - [12] Salton, G. "*The SMART Retrieval System – Experiments in Automatic Document Processing*", Prentice Hall, 1971.
 - [13] Schütze, H., and Pedersen, J.O. "*A Co-occurrence-based Thesaurus and Two Applications to Information Retrieval*", in 4th Proceedings of the RIAO Intelligent Multimedia Information Retrieval Systems and Management, vol. 1, pp. 266-274, 1994.
 - [14] Sparck Jones, K. "*Automatic Keyword Classification for Information Retrieval*", Butterworths, London, 1971.
 - [15] Yang, J-J. & Korfhage, R.R. "*Effects of Query Term Weights Modification in Document Retrieval - A Study Based on a Genetic Algorithm*", University of Pittsburgh, Second Annual Symposium on Document Analysis and Information Retrieval, IEEE, pp. 271-285, 1993.