MODELLING DOCUMENT CATEGORIES BY EVOLUTIONARY LEARNING OF TEXT CENTROIDS

J.I. Serrano, M.D. Del Castillo
Instituto de Automática Industrial, CSIC. Ctra. Campo Real, km.0,200. La Poveda. Arganda del Rey. 28500 Madrid. Spain

Abstract: This paper deals with a supervised learning method devoted to producing categorization models of text documents. The goal of the method is to use a suitable numerical measurement of example similarity to find centroids describing different categories of examples. The centroids are neither abstract nor statistical models, but rather consist of bits of examples. The centroid-learning method is based on a genetic algorithm, the GAT. The categorization system using this genetic algorithm infers a model by applying the genetic algorithm to the set of preclassified documents belonging to a category. The models thus obtained are the category centroids that are used to predict the category of a new document. The application of this system is the task of classifying incoming documents.

Key words: similarity function, centroid, genetic learning, text classification

1. INTRODUCTION

There are well-known methods for automating the building of clusters and descriptive models of text documents[6]. Most such methods are included in the machine learning paradigm, where the categorization problem is envisioned as a process of learning supervised by the knowledge of the categories and of the training instances that belong to them. Documents manually classified are the key resource in such a paradigm, and a general inductive process automatically builds a text classification model for every category by extracting the main features from preclassified documents.

Two other example-driven techniques that infer no classification model are the k-nearest neighbour **(K-NN)**[2] and textual case-based reasoning **(TCBR)**[3]. Systems using these techniques start with a set of documents associated by hand with a kind of "solution". According to the terminology employed in these methods, when a new example is entered to be solved, the method compares the new example to the stored examples and retrieves the most similar ones. Then, the "solutions" associated with these similar examples are used to provide the solution to the new example. When examples are text documents, the "solution" associated with each document is the category the document belongs to. These methods employ no learning stage, and the only task that precedes the comparison of documents, very time- and storage space-consuming, is the allocation of preclassified documents.

2. DOCUMENT CLASSIFICATION TASK

This paper deals with text supervised learning where text documents are the only information available. The goal of the method is to find the centroids that describe the different given document categories. Every centroid is neither an abstract nor a statistical model but rather consists of the set of words selected from the category documents that, when used for document categorization, yields the highest effectiveness of the model. The proposed centroid-learning stage is based on a genetic algorithm (GAT).

An important requirement of the supervised centroid-learning is to use a small amount of training examples for building a categorization system taking the right classification decisions in any text domain regardless of the domain characteristics. There are many specific thematic domains in which it is very difficult to obtain significant samples of training documents. Most of current exampled-based systems need exhaustive training sets implying high download and store costs for obtaining a final, reliable classification system[6].

The centroids learned can be used later for organizing tasks like classification and summarization[8]. The application of the GAT-based system focuses on the task of classifying incoming documents in several non-disjoint categories.

The following section describes the text preprocessing step. The details of the newly developed genetic algorithm and the proposed similarity function are discussed in Section 3. Sections 4 and 5 describe the generalization of the genetic algorithm and the classification application, respectively. Section 6 reviews the experimental settings and results. Last sections contain the conclusions.

3. GENETIC ALGORITHMS FOR TEXTS (GAT)

The information contained in text documents is often expressed in a natural language that must be mapped to a representation understandable by the classifier algorithm. The preprocessing step used here is comprised within the *bag of words* approach commonly used in most text applications[8]. The task of the system is to scan the text of documents in every category and to turn that text into lists of words, together with their occurrence frequency. Next, words belonging to a stop list or words without semantic contents are removed, and several stemming procedures are applied[4]. A preprocessed document is a list of pairs, each pair consisting of a word and its occurrence frequency.

Genetic algorithms are an optimization technique that simulates the natural evolution process[1]. Beginning with an initial population of individuals or chromosomes representing tentative solutions to a problem, a new generation is created by combining or modifying the best individuals of the previous generation. The process ends when the best solution is achieved.

The problem proposed is to obtain a centroid document representing the documents of a class. The central notion is a measure of similarity among documents, so documents in a class show a high intra-class similarity and a low inter-class similarity. One possible solution to this problem could be generated by taking a random set of words from the documents in a class and measuring the similarity between that random set and every document. Due to the huge search space and the lack of good heuristics based on the semantics of the words, there are many potential initial sets of words.

The initial population designed here to lead the search for centroids consists of the preprocessed documents of every class, since every document is the most similar to itself and one possible centroid with regard to the other documents in its class. After applying the GAT method, the centroid of every category is a document composed of different portions of every labelled document belonging to the category.

3.1 Text Representation and Genetic Operators

Every chromosome symbolizes a possible centroid document. The chromosomes of the initial population are the documents obtained after preprocessing. The genes of a chromosome are pairs consisting of a word and its occurrence frequency. Since documents are of a variable size, the length of chromosomes is also variable.

The genetic algorithm for texts uses three operators: copy, crossover and mutation.

Copy Operator. This operator selects some of the best chromosomes of a population and duplicates them in the next generation. Since the evolution of a population over time can produce worse chromosomes than the original set, this operator provides a mechanism for remembering chromosomes that were previously useful.

Two-Point Crossover Operator. Typically, a simple crossover operator generates a new offspring from selected parent chromosomes by swapping all the genes between a randomly selected position and the length of the chromosome less one. The version used by this GAT selects at random two positions in each parent chromosome.

There are two reasons for using a random multiple-point crossover operator. The first crossover point must belong to each parent because of the different length of chromosomes. The second random crossover point in each parent allows the number of exchanging genes to be smaller than the number of exchanging genes with a simple crossover operator. The resulting offspring may therefore turn out to be modified to a lesser degree.

Mutation Operator. This operator selects one gene of a chromosome and modifies its value. In the proposed algorithm, there are two ways of modifying a gene. One way lies in replacing the selected gene by another randomly selected from the full current set of words present in the documents of a category. This option enables all the words to contribute fairly. The other way to mutate a gene lies in increasing or decreasing the value of the occurrence frequency of the gene. This latter kind of mutation is justified by the design of the fitness function, as discussed in the following subsection.

3.2 Population Fitness Function

The objective of the fitness function is to compute some measurement of the profit or goodness a chromosome would have as a centroid document of a class. According to the assumptions, every chromosome of a concrete population is a centroid. The closer to every preprocessed document the centroid is, the better it will be.

Obviously, the chromosome taking the highest fitness value will be the best centroid. The main point of the fitness function is to find the measurement of similarity or, inversely, the measurement of distance among documents. The more similar a document is to another, the less distance will exist between them.

There are many studies about how to characterize the similarity between any two texts; some are statistics-based, and others are word semantics-based[5]. In this paper, similarity is calculated by a statistical function that

takes into consideration the number of times words occur within the compared texts. Eq. 1 reflects the similarity function.

$$\text{Similarity}(X,Y) = \Sigma nx_i \cdot ny_j \quad (\forall I,j \,/\, x_i \in X \,\&\, y_j \in Y \,\&\, xi = y_j) \qquad (1)$$

where nx_i is the number of appearances of word x_i in document *X*, and ny_j is the number of appearances of word y_j in document *Y*. This function calculates the similarity between document X and document Y. The degree of similarity between two documents is obtained by multiplying the number of occurrences of the words that are common to both documents. Thus, if a centroid contains many relevant words that are present in many documents, the centroid will take a high average similarity value with every document and therefore a low average distance value.

The fitness function value of a chromosome is the average similarity between all documents and that chromosome (see Eq. 2).

$$\text{Fitness}(Chr) = \frac{\Sigma_i^N \text{Similarity}(i, Chr)}{N} \qquad (2)$$

where *i* is document *i* from the initial set, *N* is the number of initial documents, and *Chr* is the chromosome evaluated.

In order for a genetic algorithm to be applied, certain other parameters must be determined as well, such as the maximum number of generations, the stop fitness value, the probability of application of operators and the operator workspace. The sizes of the workspace for the Copy, Two-Point Crossover and Mutation operators have been set to 30%, 65% and 80%, respectively. The optimal values of the application probabilities of the Copy, Two-Point Crossover and Mutation operators were 0.4, 0.65 and 0.8, respectively. All these values have been determined experimentally.

4. APPLYING GAT TO DOCUMENT CLASSIFICATION

The application of the GAT to text web pages or hypertext implies taking into account certain additional issues regarding the presence of often ungrammatical text in web pages. First, in the preprocessing step, the frequency of word occurrences is increased for some word formats in an experimental way.

Four different types of information can be found in web pages explicitly: URL, META keywords, hyperlinks and plain text. Words can assume a

different semantic power depending on their placement. Based on this division of documents, the preprocessing step will generate four lists of word/occurrence frequency pairs, one each for the four types of information.

The GAT can consider a web page as a whole text or as four parts of text. When the web page is to be processed as a whole, the GAT will apply the crossover operator only to a part chosen at random for every two parent chromosomes selected to be crossed. If the web page is considered as consisting of four parts, then the chromosomes handled by the GAT are divided into four parts and the genetic operators will be applied to the four parts in a parallel manner. Therefore, the fittest chromosomes (and the text categories) can be modelled by as many centroids as different types information exist in the web pages belonging to the categories. Thus, the application of the GAT can be generalized to include grammatical texts and hypertexts, because any kind of document can be mapped onto the described web representation; and therefore use can be made of the information that web page authors give when they place a word in some special position and/or format.

The classification process begins when the system receives an unlabeled document. First, the similarity between the document and every learned centroid according to Equation 1 is calculated. Next, the document is classified as belonging to the category or categories whose centroid or centroids are closest to the document. In each category a threshold value of the similarity is set so that any document with a similarity less than the threshold is not classified into the category. An interesting advantage of the similarity measurement is that the values it takes for a document with respect to each centroid can be seen as degrees of membership in each category.

When the categorization system classifies web pages, it takes into account the existence of centroids composed of four centroids. Therefore, when a new web page is being classified, the similarity between every type of information on the page and the corresponding centroid is calculated. The final similarity between a web page and a category is given by the average of at most four similarity measurements.

5. EMPIRICAL TESTS

The system described has been evaluated on two text collections. The first experiment allowed to set up the influence of the number of generations and the number of centroids per category to the system performance. Once these parameters were determined, the second experiment intended to evaluate two issues involved in the genetic centroid-based approach: 1) differences in classification performance, checked by considering the web

documents as a whole or as four separate types of information (URL, META, TEXT, LINKS); 2) the results obtained with a two-point crossover operator instead of a simple crossover operator.

The first experiment was carried out on the Reuters-21578 collection. A subset of this collection was selected consisting of 1,987 documents belonging to eight categories. These documents were divided into two groups, a training set of 240 examples and a test set of 1,747 examples. The GAT method was applied to the training set, and the centroids thus obtained were used to classify the test set. Table 1 shows the classification performance for each category in the rows and the results for different values of the number of generations in the columns.

Table 1. Average results from five runs of the genetic algorithm on Reuters test set for classification with different maximum numbers of generations

	20 Generations			50 Generations			100 Generations			175 Generations		
	Pr	*Rc*	*F*	*Pr*	*Rc*	*F*	*Pr*	*Rc*	*F*	*Pr*	*Rc*	*F*
ACQ	69.91	64.6	67.15	93.88	61.4	74.24	84.68	**79.6**	82.06	**99.5**	70	**82.15**
COF	49.18	93.75	64.51	**56.86**	90.62	**69.87**	50.81	**96.87**	66.66	37.03	93.75	53.09
EAR	96.33	72.37	82.65	97.82	78.62	87.17	**99.85**	**87.87**	**93.48**	98.40	87.5	92.59
GOL	90.9	80	85.1	**99.99**	**88**	**93.61**	95.65	**88**	91.66	79.16	76	77.55
NAT	88.23	44.11	58.82	78.26	**52.94**	63.15	74.99	44.11	55.55	**99.99**	**52.94**	**69.23**
MON	**99.99**	**77.77**	**87.49**	80.64	55.55	65.78	**99.99**	71.11	83.11	**99.99**	**77.77**	**87.49**
SUG	**99.99**	70.73	82.85	**99.99**	73.17	84.5	**99.99**	**75.6**	**86.11**	**99.99**	73.17	84.5
TRA	**99.99**	49.6	66.31	**99.99**	57.6	73.09	**99.99**	**64**	**78.04**	98.79	59.2	74
Avg.	86.81	69.11	76.95	96.93	59.5	73.73	92.33	**71.8**	**80.78**	**99.14**	64.6	78.22

Classification performance is based on calculating three different measurements: precision or percentage of correct predictions, recall or percentage of documents that have been correctly classified, and F-measure as a combination of the precision and recall measurements, *F-measure=(2*precision*recall)/(precision+ recall).*

The greater the number of generations, the better the results are. Although the best results depend on the category, it seems that the best macroaveraged value of the maximum number of generations is 100.

A similar analysis was carried out to determine the influence of the number of centroids to the classification performance. The results showed the more centroids there are, the worse the results. Using more than two centroids is not a good option, because although the precision value is kept the remaining performance values are decreased.

The second experiment was carried out using a collection of web pages called BankSearch. This data set is jointly provided by BankSearch

Information Consultancy Ltd. and the Computer Science Department at the University of Reading. The collection consists of 10,000 web documents classified into ten categories of equal size, each containing 1,000 web pages[7]. A subset of this collection was selected, consisting of 4,625 examples equally distributed into five categories (Commercial Banks, Java, Astronomy, Soccer, Sport). All the categories were divided into four disjoint sets: one training set with 50 examples to learn the category centroids and three test sets with 250, 125 and 500 examples, respectively, to validate them. In this experimental setting, GAT was run for 100 generations and only one centroid per category was selected from the last generation.

The issue of considering web pages as a whole or as four separate types of information (URL, META, TEXT, LINKS) was explored using the first test set. The different performance between a two-point crossover operator and a simple crossover operator was explored using the second test set.

Table 2 and Table 3 show the classification performance for Test Set I and Test Set II, respectively.

Table 2. Average results from five runs of the genetic algorithm on Test Set I

Categories TEST SET I	**Full Document Two-Point Crossover**			**Four Information Types Simple Crossover**			**Four Information Types Two-Point Crossover**		
	Pr	***Rc***	***F***	***Pr***	***Rc***	***F***	***Pr***	***Rc***	***F***
COMMERCIAL	**95.45**	8.4	15.441	89.28	60	71.770	90.53	**95.6**	**92.996**
JAVA	**80.83**	54	64.748	79.23	**99.2**	**88.099**	77.18	98.8	86.666
ASTRONOMY	98.03	40	56.818	93.13	**76**	**83.700**	**99.41**	68.4	81.042
SOCCER	53.57	**98.8**	69.47	89.64	90	89.820	**91.86**	90.4	**91.129**
SPORT	**100**	5.6	10.606	**100**	67.6	80.668	**100**	**72**	**83.720**
Average	85.57	41.36	55.765	90.26	78.56	84.004	**91.79**	**85.04**	**88.286**

Table 3. Average results from five runs of the genetic algorithm on Test Set II

Categories TEST SET II	**Full Document Two-Point Crossover**			**Four Information Types Simple Crossover**			**Four Information Types Two-Point Crossover**		
	Pr	***Rc***	***F***	***Pr***	***Rc***	***F***	***Pr***	***Rc***	***F***
COMMERCIAL	93.33	11.2	20	**95.23**	64	76.555	91.79	**98.4**	**94.980**
JAVA	**82.60**	60.8	70.046	77.63	**100**	**87.412**	78.70	97.6	87.142
ASTRONOMY	98.33	47.2	63.783	94.73	72	81.818	**100**	**76.8**	**86.877**
SOCCER	53.44	**99.2**	69.467	**95.72**	89.6	**92.560**	92.06	92.8	92.430
SPORT	**100**	5.6	10.606	**100**	77.6	87.387	**100**	**80**	**88.888**
Average	85.54	44.8	58.803	**92.62**	80.64	86.215	92.51	**89.12**	**90.783**

The columns in Table 2 and Table 3 show the values of these measurements with three different GAT configurations: web pages as a whole text and web pages consisting of four types of information with

simple and two-point crossover operators. All the numerical values given in the tables are the average result of five runs of the genetic algorithm.

As the results show, when the GAT is configured to consider four different kinds of information in web documents, it gives a better average performance than when it processes documents as a whole. The two-point crossover yields a slightly better performance than the simple crossover in both test sets. The best configuration for the algorithm therefore seems to be the configuration that considers the types of information in each document separately and employs the two-point crossover operator.

Table 4. Average results from five runs of the genetic algorithm on Test Set III

Categories TEST SET III	**Four Information Types**		
	Pr	***Rc***	***F***
COMMERCIAL	71.69	98.8	83.095
JAVA	78.55	76.2	77.360
ASTRONOMY	96.15	75	84.269
SOCCER	89.76	91.2	90.476
SPORT	100	71.6	83.449
Average	87.23	82.56	84.830

This configuration was used to test classification performance in Test Set III, the test set with the largest number of web documents. Table 4 shows that the system performed very well in some categories, and, on average, the values of the performance measurements are quite high in all three test sets with only 50 training examples per category.

Table 5. Performance results of the GAT model and the Naïve Bayes classifier

Categories TEST SET III	**Four Information Types Two-Point Crossover**			**Naïve Bayes**		
	Pr	***Rc***	***F***	***Pr***	***Rc***	***F***
COMMERCIAL	71.69	**98.8**	83.095	**87.4**	98	**92.39**
JAVA	78.55	76.2	77.360	**95.6**	**77.2**	**85.02**
ASTRONOMY	96.15	**75**	**84.269**	**97.7**	71.4	83.21
SOCCER	**89.76**	**91.2**	**90.476**	75.44	84.8	79.84
SPORT	**100**	**71.6**	**83.449**	69.52	84.4	76.24
Average	**87.23**	82.56	**84.830**	85.13	**83.16**	84.13

Table 5 shows a comparison between the GAT model and a Naïve Bayes classifier on the BankSearch collection. The training set was composed of SO examples and the test set of 250 examples (Training Set and Test Set I). Due to the high dimensionality of the word space, a dimensionality reduction

technique selecting 20% of the best ranked words was used before applying the Naïve Bayes classifier. Scoring words was carried out by the gain information statistical measurement. The results obtained strengthen the successful classification performance of the genetic-based model working on few training documents.

6. CONCLUSIONS

A genetic algorithm for texts has been proposed for obtaining centroid documents that describe text categories by learning those centroids and using them to classify documents. This technique consumes little time in the classification stage. The system requires no computations to find the similarity between new documents and the documents stored in the repository or the case base, but only to find the similarity with learned centroids. The classification results have shown that the technique works quite well using very few training documents and at most two centroids per category. The classification results can even be improved by fine-tuning the algorithm parameters and perhaps by selecting more representative training examples.

REFERENCES

1. D. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning* (Addison-Wesley Publishing Company, Inc, 1989)
2. S. Han Eui-Hong, G. Karypis and V. Kumar, Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification, PAKDD'2001 (2001).
3. M. Lenz, A. Hubner and M. Kunze, Textual CBR, in *Case-Based Reasoning Technology,* edited by M. Lenz, B. Bartsch, H.D. Burkhard and S. Wess (Springer. LNAI 1400, 1998).
4. M.F. Porter, An algorithm for suffix stripping, *Program,* **14**(3), 130-137 (1980).
5. M.M. Ritcher, The Knowledge Contained in Similarity Measures. Invited Talk at ICCBR-95 (1995).
6. F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys,* **34**(1), 1-47 (2002).
7. M.P. Sinka and D.W. Corne, A Large Benchmark Dataset for Web Document Clustering, in *Soft Computing Systems: Design, Management and Applications,* edited by A. Abraham, J. Ruiz-del-Solar, and M. Koeppen (Volume 87 of Frontiers in Artificial Intelligence and Applications, 2002), pp. 881-890.
8. K. Zechner, A Literature Survey on Text Summarization. Paper for Directed Reading (Fall 1996), *Computational Linguistics* (1997).