Sci. J. Fac. Sci. Minufia Univ., 2009, Vol.XXIII(2).43 - 63

A HYBRID GENETIC AND C4.5 ALGORITHM FOR TEXTUAL DOCUMENT CLASSIFICATION

MONA GAMAL[1], AHMED ABO EL-FATOH[2], ELSAYED RADWAN[3], AZIZA ASEM[4]

Mansoura University, Faculty of Computer and Information Sciences
Information System Department[1,2,4]
Computer Science Department [3]
P.O.Box: 35516

ABSTRACT

The popularity of the Internet and World Wide Web increases the need for information management of electronic texts. Textual document are the easier way in saving information in all aspects on the computer in spit of the difficulties in making use of these information. This paper attempt to find a way of mining information from textual document, using hybrid of Genetic Algorithms (GA's) and the machine learning algorithm C4.5. GA depends on structuring conventions due to reducing the feature vector size without affecting the classification accuracy. The most informative features (synopses) are extracted from the document. Then, a succinct feature vector is prepared to represent the document. In progress for more classification, a machine learning algorithm is needed. The C4.5 is used to classify the document based on the succinct feature vector. The experimental results show that the proposed technique gives more accurate classification than previous methods.

Keywords: Text Classification, Feature Extraction, Genetic Algorithms, The Machine Learning Algorithm C4.5.

INTRODUCTION

Text classification aims to automatically categorize text documents into pre-defined classes or types based on their contents [2, 13, 14]. The field of information extraction, on the other hand, tackles the problem of extracting relevant information from this textual data. However, we believe that information extraction can aid text classification by identifying a small set of features in each document that provide very effective discrimination for classification.

The standard document representation used in text classification is the vector space model. In this model, each document is represented by a vector of (feature, value) pairs. Features are textual units such as words or phrases, also called terms. Values can be the presence, the frequencies, or the weights of terms. A collection of documents can then be represented as a set of document vectors or, alternatively, as a matrix $D = \{V_{ij}\}$ where V_{ij} is the corresponding feature value of term t_i in document d_j.

Since the total number of words in the document collection is large and each individual word may not appear in every document, the matrix D is often sparse. Also, the computation of the classification process can be extremely cost. Moreover, this model does not take advantage of informative terms. A common way to rank the importance of a term is based on its frequency, so that frequent terms may be given higher weights and/or less frequent terms may be discarded. This approach is subjective since terms with high frequency are not necessarily important. Applying the second cutoff can reduce the size of feature matrices but may miss out important terms as its consequence. Another problem in many systems using vector space model is that if the training data is insufficient, the classification tends to become unreliable, i.e. the accuracy will be low. Therefore, an improvement can possibly be achieved by not only reducing the size of original feature sets but also increasing the quality of features.

A large number of useful online documents are comprised largely of natural language but with some structuring conventions. These are denoted as PSLNL (Partially-Structured, Largely-Natural Documents) [8]. For example, most job advertisements are laid out as a sequence of informative regions including job description, requirements, salary, deadline, etc. In other words, they often contain similar semantic and orthographic structures. Making use of these particularities, we propose a new method to reduce the size of the feature set for PSLNL documents with out compromising the classification accuracy. First document synopses are extracted, which contain the most informative data; then succinct feature vectors are constructed based on these synopses. Finally, a decision tree machine learning algorithm uses these feature vectors to classify the documents. The method in this paper is described in the context of PSLNL documents. In fact, the requirement here is the ability to partition documents into regions. In PSLNL documents, this is achieved via machine learning on a set of basic orthographic features. We postulate that the method would apply equally to semi structured documents, such as Web pages, where markup features are available to assist in the regioning process.

In this paper a hybrid model of Evolutionary Computing search algorithm, Genetic Algorithms, GA [1] and the machine learning algorithm C4.5 [3,4,5] has been used to classify textual documents. GA shows high ability to find an admissible solution in a large search space and to find the global minima, even in a noisy and discontinuous search space without using differentiable information about the cost function. Also, the machine learning algorithm C4.5 shows high ability to produce an accurate classification. A new fashion of GA and C4.5 is used to classify documents via feature extraction process and then generating a succinct feature vector to give the classification process. According to the previous results of the machine learning in textual classification, a comparison with the new hybrid model should be appear.

The rest of this paper is organized as follows: Section 2 represents the preliminaries such as the declaration of the textual document classification as well as a previous work in the feature vector reduction process , Genetic Algorithms GA and the machine learning algorithm C4.5. Section 3 gives an over view on the whole system and its modules. It goes inside the system to explain in detail the feature extraction process , finding the fitness function for the GA and the textual classification process . Experimental results and conclusion will appear in sections 4and 5 respectively.

PRELIMINARIES

Recently, a growing number of statistical and machine learning methods have been applied to solve text classification problems. The major difficulty in doing this is the high-dimensionality of conventional document feature spaces. Words, or, more rarely, phrases, are typically used as features. Even with moderated-size document collections, the size of the feature space can reach hundreds of thousands. Most existing machine learning algorithms are not designed to deal with such large feature spaces, and applying them naively is computationally infeasible. In addition, the use of words as features can cause other problems: the classification might be misled by general words and discriminating words might not be used effectively. It is therefore highly desirable to reduce the size of the feature space.

Text Classification

The simplest feature selection technique, document frequency thresholding, discards words whose document frequency is not within some predetermined range. This technique is based on the assumption that neither rare words nor common words are useful as classification discriminators. Applying this cutoff can reduce the size of feature matrices but, as a consequence, may miss out important terms. Information gain can be used to rank and select the most categories predictive words [11]. This heuristic, which was first introduced in [5], is based on the probability of a document containing a given term and belonging to a given category. This method has the drawback of being strongly affected by the marginal probabilities of terms, i.e. rare terms will be given a higher score [16]. Another method measures the goodness of a term as a function of its χ^2 statistic, which used co-occurrence to extract important words from a document. In this research, the bias of the probabilistic distributions between the co-occurrence and the appearance of the most frequent words in the document were measured. Such frequent words are referred to as keywords. They have evaluated the bias by chi-squared measure and selected the most important words from [17]. An other method is the Word strength which measures how informative a word is in identifying documents. The strength of a word is the probability of finding it in the document [15].

The re-parameterization branch of automatic feature extraction aims to reduce the size of feature vectors by constructing new features as combinations or transformations of lower level features. Latent Semantic Analysis (LSA) is one such method [10].It assumes there is an underlying ("latent") structure in word usage. This structure is estimated using singular-value decomposition technique. Classification is then performed using the database of these singular values.

All existing feature extraction methods first construct a feature space as the set of all non-trivial terms that appear in the documents. After that different heuristics are used to refine the original feature set and reduce its size. Our approach differs from these in attempting to extract small, yet effectively discriminating vectors, in the first instance. To do this, it exploits the fact that documents in particular categories tend to exhibit characteristic stylistic structures.

Genetic Algorithms (GAs)

John Holland (1975) proposed an attractive class of computational models, called Genetic Algorithms (GA) [1], that mimic the biological evolution process for solving problems in a wide domain. GA has three major applications, namely, intelligent search, optimization and machine learning. Currently, GA is used along with neural nets and fuzzy logic for solving more complex problems. Because of their joint usage in many problems, these together are often referred generic name: "soft-computing". A GA operates through a simple cycle of stages:

i) Creation of a "population" of strings,

ii) Evaluation of each string,

iii) Selection of best strings and IV) Genetic manipulation to create new population of strings. The cycle of a GA is presented below in Figure 1.

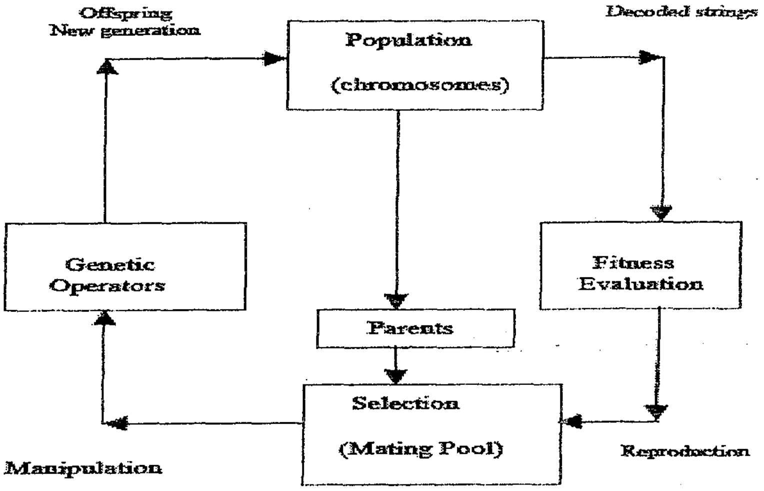


Figure 1: The cycle of genetic algorithms.

Each cycle in GA produces a new generation of possible solutions for a given problem. In the first phase, an initial population, describing representatives of the potential solution, is created to initiate the search process. The elements of the population are encoded into bit-strings, called chromosomes. The performance of the strings, often called fitness, is then evaluated with the help of some functions, representing the constraints of the problem. Depending on the fitness of the chromosomes, they are selected for a subsequent genetic manipulation process. It should be noted that the selection process is mainly responsible for assuring survival of the best-fit individuals. After selection of the population strings is over, the genetic manipulation process consisting of two steps is carried out. In the first step, the crossover operation that recombines the bits (genes) of each two selected strings (chromosomes) is executed. Various types of crossover operators are found in the literature. The single point and two points crossover operations are illustrated in figures 2 respectively. The crossover points of any two chromosomes are selected randomly. The second step in the genetic manipulation process is termed mutation, where the bits at one or more randomly selected positions of the chromosomes are altered (figure 3). The mutation process helps to overcome trapping at local maxima. The offsprings produced by the genetic manipulation process are the next population to be evaluated.

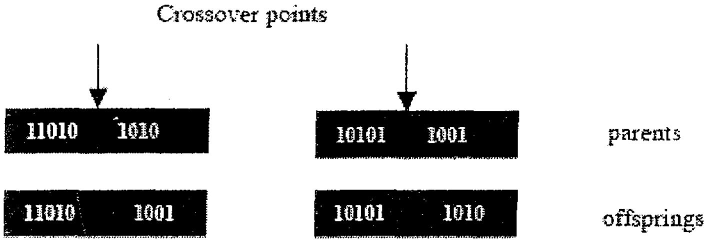


Figure 2: A single point crossover after the 3-rd bit position from the L.S.B.

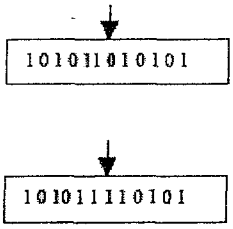


Figure 3: Mutation of a chromosome at the 6th bit position.

The Machine Learning Algorithm C4.5

The Machine Learning Algorithm C4.5 [3] [4] [5] is a program that creates a decision tree based on a set of labeled input data. This decision tree can then be tested against unseen labeled test data to quantify how well it generalizes. C4.5 is a software extension of the basic ID3 algorithm.

C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of Information Entropy. Entropy (p) can be thought of as a measure of how random the class distribution is in p.

If we are given a probability distribution $P = (p_1, p_2 ... p_n)$ then the Information conveyed by this distribution, also called the Entropy of *P*, is:

$$\text{Info}(P) = I(P) = -(p_1 * \log(p_1) + p_2 * \log(p_2) + \ldots + p_n * \log(p_n)) \qquad (1)$$

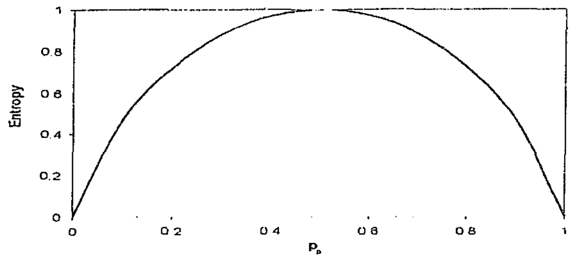


Figure 4: the Entropy Function

Notice in figure 4 that describes the entropy function that the entropy is 0 if all members of P belong to the same class. For example, if all members are positive ($p_p = 1$), then p_n is 0, and *Entropy (P)* $= -1* \log_2 1 - 0* \log_2 0 = -1* 0 - 0* \log_2 0 = 0$. Note the entropy is 1 (at its maximum!) when the collection contains an equal number of positive and negative examples. If the collection contains unequal numbers of positive and negative examples, the entropy is between 0 and 1. Entropy is minimized when all values of the target attribute are the same.Entropy is maximized when there is an equal chance of all values for the target attribute (i.e. the result is random).

The training data is a set $P = p_1, p_2, \ldots.$ of already classified samples. Each sample $p_i = x_1, x_2, \ldots.$ is a vector where $x_1, x_2, \ldots$ represent attributes or features of the sample. The training data is augmented with a vector $C = c_1, c_2, \ldots$ where $c_1, c_2, \ldots$ represent the class that each sample belongs to.

C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets and examines the normalized information gain (difference in entropy) that result from choosing an attribute for splitting the data. It measures the information that is gained by partitioning P in accordance to the test X.

$$Gain(X) = \inf o\ (P) - \inf o_x\ (P) \qquad (2)$$

Where

$$Info_x\ (P) = \sum(|P_i|/|P|) * \inf o\ (P_i) \qquad (3)$$

This is biased to tests with many outcomes.

The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sub lists.

This algorithm has few base cases. The most common base case is when all the samples in your list belong to the same class. Once this happens, you simply create a leaf node for your decision tree telling you to choose that class. It might also happen that none of the features give you any information gain; in this case C4.5 creates a decision node higher up the tree using the expected value of the class. It also might happen that you've never seen any instances of a class; again, C4.5 creates a decision node higher up the tree using expected value [3,4].

GENETIC ALGORITHMS AND C4.5 IN DOCUMENT CLASSIFICATION

Our approach tends to extract the most informative terms of the document basing on partitioning the document into heading, region titles and regions. Usually heading and region titles contain the most trivial terms but regions contain just an explanation of the titles so it is just a repeating. Hence our succinct feature vector is composed of the terms that appear on the heading and title lines.

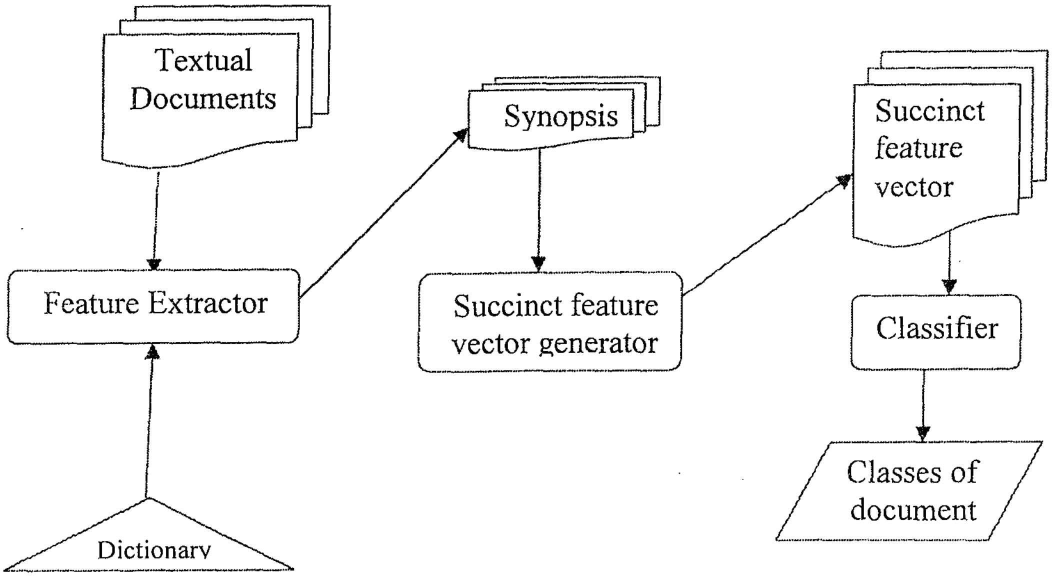


Figure 5: System Architecture

The hybrid system that outlines the main modules is represented in figure 5. The set of textual docxuments goes insisde the system through the feature extracxtor process which make the dictionary keywords.That dictionary is then farther processsed by removing the set of common words to produce the set of synopsis. These synopsis make up the succinct feature vector which the classifier C4.5 takes to build the classification tree.The main components are:-

Feature Extractor:

In that module we use GA to reduce the feature vector by extracting the only significant keywords that exist on the heading and region title. Figure 7 shows the flow chart of that module .

Succinct Feature Vector Generator(SFV):

In that module we use the set of keywords produced from the feature reduction process to create the succinct feature vector. The SFV will be a binary string each bit represents the appearance or absence of a keyword (1 for appearance and 0 for the absence) from the dictionary.

Classifier:

In that last module we use the C4.5 learning algorithm which is supplied by the SFV to produce a set of classes each class contains a set of documents that talk about the same topic or at least similar topics.

FEATÜRE EXTRACTION PROCESS

Our system depends on reducing the feature vector by using the Genetic Algorithm. Here, we use the usual structure in writing a document. According to the previous research[7], the document heading contains the most significant information such as the name of the topic the document talks about, the date when it was published, the author name and any other information such as the conference where it was published in, the region title contains the subtopics from the original topic at the heading & the regions that contain an explanation about the region title but these regions always contain so many words that are too far from being a feature. The lines could be classified as heading or region titles by a set of orthographic, relative and semantic features. So we use only the heading and the region title lines to extract the feature keywords that represent the document. These lines are used to have some structure in the document. Some of these feature structure are documented in table 1.

Table 1: feature vector

Name of feature	Description of feature	Values
Orthographic Features		
iUpperCaseLine	All alphabetic are upper case	0,1
iFirstUperCaseLine	Each word starts with uppercase	0,1
dateline	Contains date information	0,1
webLine	Contains a URL	0,1
endsWithColon	Ends with colon	0,1
startsWithDigit	Starts with a number	0,1
startsWithCharacter	Starts with non-alphanumeric	0,1
unPunctuatedLine	Not normal punctuated	0,1
Relative features		
preLineIsBlank	After a blank line	0,1
nextLineIsBlank	Before a blank line	0,1
nearBreak	Neighbor is a break line	0,1
isShortLine	Shorter than (max line length)/2	0,1
sameAsPreLine	Same format as previous line	0,1
sameAsNextLine	Same format as next line	0,1
Semantic features		
keyWordFeatured	Contains a keyword	0,1

The Features Structure:

We currently produce one feature vector for each line. The kinds of features that we consider are: orthographic, relative and semantic. The choice of specific orthographic and relative features was based on observation of typical conventions in PSLNL documents, and refined by experimentation. The features in Table 1 are suitable for many classes of PSLNL documents. For other kinds of documents, different kinds of features would need to be chosen.

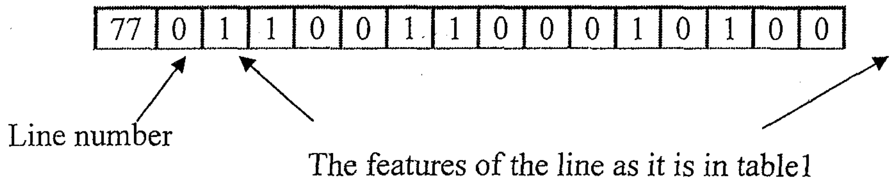
Orthographic features are style characteristics that are used to convey the role of words or sentences. Examples include using all-uppercase letters for headings, first-uppercase letters for proper names, starting text in a particular column, a star (bullet) at the beginning of a line, etc. The primary assumption used in feature generation is that authors want readers to notice the most important information easily, and so they typically mark the information by a distinguishing layout. Also, for the purpose of coherence, authors generally use the same layout to express related pieces of information.

Relative features indicate the position of a line within the document and its relationship to neighboring lines. These kinds of features are important because the significance of orthographic features is often affected by their context. For example, a line that is preceded by a blank line and followed by a line of hyphens is significant (most likely a heading), almost regardless of its own orthographic features. Similarly lines that appear near the top of the document typically have added importance. Another example might be two successive lines that have precisely the same format (e.g. both start with an asterisk) indicating a list.

Semantic features are domain-specific keywords that help to identify particular semantic contexts within the document. These contexts are used to assist in identifying what kind of information is expected in a given region .Note that orthographic and relative features are dependent on the document format (e.g. plain text documents have different features to HTML document) but are largely independent of any domain. Semantic features, on the other hand, are domain specific and need to be reconsidered for each new application of our method. It also worth noting that preliminary experiments established that these different kinds of features need to be considered in combination in order to achieve satisfactory classification results [7].

The Genetic Algorithm Process:

Since GA which is an evolutionary computing algorithm, shows high ability to find an admissible solution in a large search space. The new hybrid algorithm tries to use this property in solving the high dimensionality problem. So the Genetic Algorithm's chromosome is defined on the heading and the region title lines to extract the feature keywords that represent the document. The individual of the GA will be the combination of these features 1 for the presence of the feature and 0 for the absence. The individual in the GA, illustrated in Figure 6, will be a binary string contains the binary representation of the line number and fifteen other bits for the features mentioned on Table 1.



77	0	1	1	0	0	1	1	0	0	0	1	0	1	0	0

Figure 6: the GA individual

Using the lines of a document as the population and measuring the fitness of each individual. The GA generation has been evolved to select the fittest individuals (lines) by applying the crossover process, which yields two different offspring from two different chromosomes and the mutation process, which is a change in a single gene on a random chromosome.

The Feature Extraction process uses GA to reduce the feature vector by extracting the only significant keywords that exist on the heading and region title. The flow chart of that process is presented in figure 7.The process Picks up a document from the document set , Parses the document into lines , Generates a random string for each line.The Genetic Algorithm Calculates fitness for each string (individual- or chromosome), Constructs the pole mate based on a random process to apply the genetic operations, Applies the selection process, which selected the best individual in the current generation to be in the next generation , Applies the crossover process, which yields two offspring in the next generation form two different individuals in the current generation, Applies the mutation process, which yields a new individual in the next generation by simply change a single gene in a random chosen chromosome (individual) and Tests if it is the last generation. If yes stop and get the last generation lines else repeat the genetic algoriyhm again.The module then Parses lines of the last generation into keywords and remove all common words, Adds keywords to dictionary and tests if it is the last document in the training set. If yes finish module else repeat the whole module again.

Finding the fitness function:

It is very important to determine the basis on which a line is picked up as a heading or a region title.These lines words are features of some importance. So the following is an explanation of the fitness of an individual.

Heading Recognition:-The heading of the document contains its title and some other information about the conference where it was published in, website and date. This information has special orthographic features. For example, a conference CFPs contains at least the conference title, possibly with additional information such as URLs, date and location, etc. The fitness function considers primarily orthographic and relative features to identify the lines in the heading region. That means the positions of orthographic and relative features in the individual should be one. The main set of features to be a heading could be isUpperCaseLine, dateline, webLine and unPunctuatedLine. The amount of 1's at these positions on the individual judge its fitness value to be a heading line.

Other information might also appear near the top of the document, such as reminders and notices which indicate the type and purpose of the document. This information has no relevance to the heading section and is not part of the information that we wish to extract from the document.

Region Title Recognition:-The document heading contains the main topic but the most detailed information to be extracted from the document really exists in the rest of the document. However, parsing this section is also difficult because of its loose structure,t we just proceed in the same way finding the set of features that identify the region titles and counting the 1's on them to find the amount of fitness to be a title region line. The set of features could be isUppercaseLine, isFirstUppercaseLine, startsWithDigit, startsWithChar, unPunctuatedLine, preLineIsBlank, nextLineIsBlank, nearBreak, isShortLine.

The fitness of a chromosome is measured by the following formula:

$$F_i = (\sum 1\text{'s } \textit{in the chromosome}) / N \qquad (4)$$

N is the number of features that determine heading and region titles.

A basic observation made by Prof. Holland is that "An individual with an above average fitness tends to increase at an exponential rate until it becomes a significant portion of the population [6]. Then the individual selected for the next generation should have the next condition true.

$$F_i > F_{avg}(\text{pop}) \qquad (5)$$

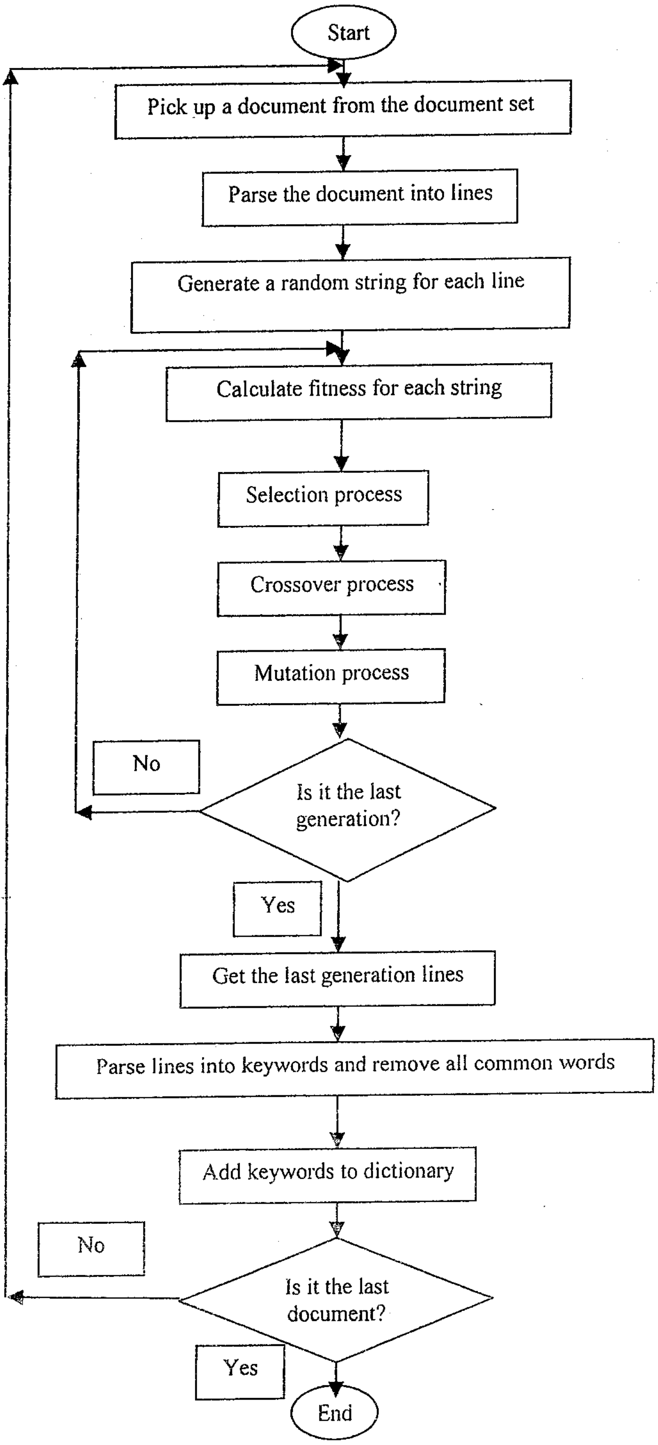

Start
Pick up a document from the document set
Parse the document into lines
Generate a random string for each line
Calculate fitness for each string
Selection process
Crossover process
Mutation process
No
Is it the last generation?
Yes
Get the last generation lines
Parse lines into keywords and remove all common words
Add keywords to dictionary
No
Is it the last document?
Yes
End

Figure 7. Flow chart of the feature extraction process

The following features have been used to enforce a more efficient representation phase:

- The best individual from the previous generation substitute the worst in the current generation if no improvement is made.
- Crossover oprator is chosen to be linear combination of two values, arithmetic crossover
- Mutation operator is chosen to be uniform mutation.

Textual document classification process:

This process comes in two stages 1) taking all titles (T) and headings (H) and performs some further processing on them and makes the succinct feature vector (SFV) 2) input the SFV to the C4.5 to output the document classes.

The Succinct Feature Vector:

This stage of the classification process takes the T and H values produced by the region extractor and combine them to produce a synopsis of the document. This is further processed into a keyword-based feature vector that represents the document and is used as the basis for classification (since these feature vectors are relatively small compared to the ones typically used in document classification, we denote them succinct feature vectors). The classifier itself is produced via machine learning on the succinct feature vectors of a pre-classified training set.

The synopsis is produced by forming a set of all words from T and H. This set is then refined via the standard text retrieval stemming and stopword removal operations. Finally, various classes of words that do not aid in classifying the type of the document are replaced by their class name. For example, the location of a conference might be useful in finding the program for a specific conference, but it is not a useful feature in determining whether a given document is a conference program or not. We thus replace proper names by the class to which they belong (e.g. country, person, and organization). Similarly, all dates times and URLs are replaced by their class names (e.g. www.acm.org and www.cse.unsw.edu.au would both be replaced by the name URL). The effect of all of these transformations is that we have a considerably smaller set of words to deal with, but without losing substantial discriminatory power for the task of classification .We also perform one transformation that actually increases the size of the words: we distinguish between the occurrence of a term in H or T. The rationale for doing this is that terms appearing in H typically have a different function to the same terms appearing in T. Note that this transformation actually takes place before the transformations described above[7].

Given a training set, we form a feature space by forming a union of all of the individual synopses. The individual terms form the features (dimensions of this space). A succinct feature vector can be generated for a given synopsis by assigning 1 for each feature that appears in that synopsis, and 0 for any term that does not appear. This will generally lead to quite "sparse" feature vectors.

Classification by C4.5:

The classification process is to build a model based on a set of properities or features. The information required for the classification is obtained either by 1)expert in the field 2)induction (i.e) generalizing specific examples through records.The artificial intelligence tries to use induction of machine learning algorithms to help in the classification preoess. C4.5 machine learning algorithm is a program that creates a decision tree based on a set of labeled input data.A decision tree is composed of leafs that indicate classes ansd decision nodses that specifies some tests to be carried out on a single attribute value with one branch and subtree for each possible outcome of the test. A decision tree can be used to classify a case by starting at the root of the tree and moving through it untill a leaf is encountered. This decision tree can then be tested against unseen labeled test data to quantify how well it generalizes. C4.5 is a software extension of the basic ID3 algorithm. Pruning decision trees is one of the advantages of using C4.5.The idea is to remove parts of the tree that do not contribute to the classification accuracy on unseen cases , producing a tree less complex and thus more comprehensible called a pruned tree.An other advantage of using C4.5 is the windowing process .Windowing is a solution for memory limitations in old times when the training set is very large for a large set of attribute.It is to take a subset of the training set randomly and then generate the decision tree then test it on the remaining cases if there are no exception so it is the final decision tree .If not the exceptions are added to the winsdow and regenrate the decision tree .this is repeated again and again untill the correct decision tree is found.The succinct feature vector is the set of variable that C4.5 takes to build the decision tree . That decision tree is the classifier for this training set. Basing on the pruning and windowing features C4.5 produces a more comprehensive tree that is more accurate .That decision tree is then usesd to classify un seen cases. The result is a classifier that can take new documents and accurately map them to an appropriate class, where each incoming document needs to be processed as described above to produce a succinct feature vector.

EXPERIMENTAL RESULTS

We developed a system to implement our hybrid model that is composed of two main sub models. The first sub model involves the Feature Extraction (FE) process that is an implementation of the GAs in building the dictionary or the succinct feature vector as showed in the flow chart in Figure 8. The second sub model is implemented by using C4.5 to classify the output of the first sub model to produce the categories or predefined classes we introduced in our experiment shown in table 2. The system was based on the region extractor implemented in some previous work [7] [8] [9] on content extraction, extended in a straightforward way to generate succinct feature vectors. The previous work was devoted to implement FE process using the C4.5 machine learning algorithm but we used the GA as a replacement to show that its optimality proprieties produce better results than using C4.5 that depends on making pruned decision trees. Those pruned decision trees has some disadvantages as it eliminate sub trees to make the tree size smaller but this simply affects its accuracy. GA gets the optimal solution from some population according to some fitness function with no ignorance or pruning of any set of the population.

In our experiment we tried to find the best parameters(number of generations ,crossover rate and mutation rate) that help us in reducing the classification process error rate. After a number of trails we succeeded to reach the best parameters. Using the document lines as the population size for the GA in the feature reduction process depending on 500 generation to find the best generation with 70% cross over rate and 0.01 a mutation rate we could reach better classification results than C4.5 used before in feature reduction. Figure 8 shows the training set size and the root mean squared error results from our experiment compared to some results of using C4.5 in the same process. It is obvious that GA proved its optimality in reducing the error in much smaller training set sizes than C4.5 . Because of the random search that GA depends on there are some results that have different error results for the same training set size and the same number of iteration .

The results indicate an average accuracy of around 90%, with accuracies above 85% even for quite small training sets. This compares favorably with previous systems for classifying documents, whose accuracy is typically in the range 80%-90% (e.g. [2], [13], [14], etc.).

Table 2: Document Categories
E-commerce
PNN
Text Classification
Feature Extraction
RNN
Tutorial
IR
GA
Optimization
Feature Extracting & GP
LSA
Search Arabic Text
RS
NN
GNN
Writing Research

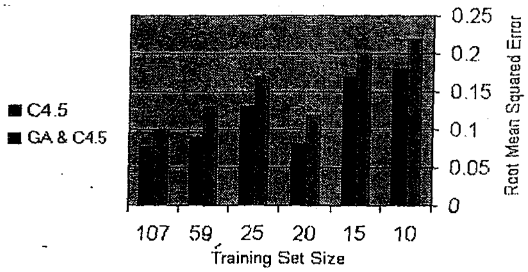


Figure 8: Feature Reduction Results By GA & C4.5

The result appendix shows some figures of each sub model input/output files.

The input of the FE process is the training set and the test set (pdfs each one of them talks about a topic). These topics will be the predefined classes also these pdfs are called the training set which is used to build the dictionary & train C4.5 for the classification model. The set of the documents (pdfs) used to prepare the test set based on the dictionary produced before to be used in calculating the accuracy of the classification model produced by C4.5 in the classification process.

Figure 9 shows the output files of the FE process which is written in a format suitable for the C4.5 to take as an input. Then C4.5 uses its input file to build the classification model and make the categories mentioned in table 2 and the decision tree for the model as shown in figure 10 and figure 11. Then we use the test set file to test the classification model.

Figure 12 show the categories of the documents in the test set produced by the C4.5 decision tree known that the test file contains the feature vector of the documents in the test set with no predefined classes and thus we performed the classification process based on the plain text of the documents and simply reached our target.

CONCLUSION

Text classification aims to automatically categorize text documents into pre-defined classes or types based on their content. Two main problems appear at the classification process. First the feature extraction process that aims to minimize the size of the feature vector by selecting the only significant features. Second the classification algorithm that reduces the error rate and increases the accuracy and performance.

This paper showed that evolutionary algorithms and robust search such as the GA can participate in the feature extraction process that helps in facilitating the text classification process by reducing the high dimensionality of the feature vector also increases performance and accuracy. The feature reduction can be performed by selecting the features (words) that belong to the head or title region from the document because only these words refer to the document category as the previous research mentioned. These words will form the succinct feature vector (the reduced feature vector) that helps in facilitating the text classification process by working only on the significant words in the document.

For the classification problem we used C4.5, machine learning algorithm, to build decision trees from a set of training data using the concept of information entropy. The decision trees classify unseen cases (documents) for their predefined classes as shown in the experimental results.

In contrast with the previous research , a comparison between the hybrid system of GA and C4.5 and C4.5 alone in the text classification process ,the new hybrid system provides its ability to achieve more accurate classifications for the same training set size and hence increases the classification performance.

REFERENCES

[1] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," Reliability Engineering & System Safety, vol. 91, no. 9, pp. 992-1007, September 2006.

[2] F. Wu, R. Hoffmann, and D. S. Weld, "Information extraction from wikipedia: moving down the long tail," in KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY, USA: ACM, pp. 731-739, 2008.

[3] J. R. Quinlan, *C4.5:* Programs for Machine Learning. Morgan Kaufmann, January 1993.

[4] J. R. Quinlan, "Improved use of continuous attributes in c4.5," Journal of Artificial Intelligence Research, vol. 4, pp. 77-90, 1996.

[5] J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, no. 1, pp. 81-106, March 1986.

[6] Konar, A.," Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain," CRC Press, Inc, Boca Raton, FL, USA, 2000.

[7] Ma, L., Shepherd, J., and Nguyen, A., "Document classification via structure synopses," In Proceedings of the 14th Australasian Database Conference - Volume 17 (Adelaide, Australia). K. Schewe and X. Zhou, Eds. ACM International Conference Proceeding Series, vol. 143.Australian Computer Society, Darlinghurst, Australia, 59-65, 2003.

[8] Ma, L., Shepherd, J. & Zhang, Y. ," Extracting information from semi structured data," inAdvances in Web-Age Information Management Third International Conference, WAIM , Beijing, China, Proceedings Series: Lecture Notes in Computer Science , Vol. 2419 Meng, Xiaofeng; Su, Jianwen; Wang, Yujun (Eds.), XV, 446 p. August 11-13, 2002.

[9] Martín-Bautista, M.J., Larsen, H.L. and Vila, M.A., "A Genetic Fuzzy Classifier to Adaptive User Interest Profiles with Feature Selection," Proc. of the European Society for Fuzzy Logic and Technology (Eusflat-Estylf) Joint Conference 327-330. Palma, Spain, 1999b.

[10] Nakov, P.," Latent semantic analysis of textual data," In Proceedings of the Conference on Computer Systems and Technologies (Sofia, Bulgaria), B. Rachev and A. Smrikarov, Eds. CompSysTech '00, ACM, New York, NY, 5031-5035. 2000.

[11] Padraig Cunningham," Dimension Reduction," In Technical Report UCD-CSI, 2007.

[12] POMIKÁLEK, Jan. ŘEHŮŘEK, Radim. ," The Influence of Preprocessing Parameters on Text Categorization," International Journal of Applied Science, Engineering and Technology, Thailand. ISSN 1307-4318, vol. 4, no. 1, pp. 430-434, 2007.

[13] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive learning algorithms and representations for text categorization," in *CIKM '98*: Proceedings of the seventh international conference on Information and knowledge management. New York, NY, USA: ACM Press, pp. 148-155, 1998.

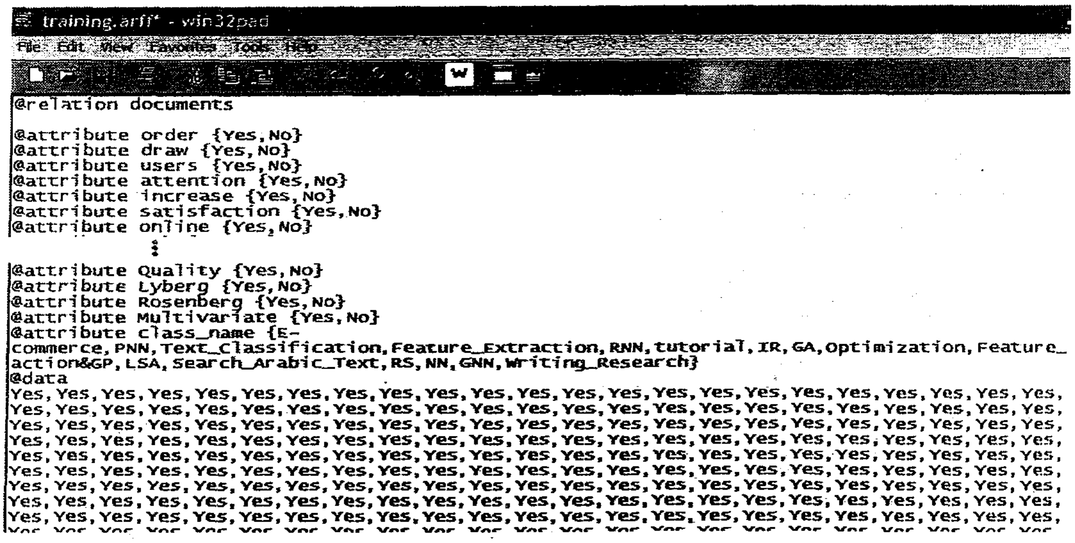
[14] Wenqian Shang, Youli Qu, Haibin Zhu, Houkuan Huang, Yongmin Lin, Hongbin Dong, "An Adaptive Fuzzy kNN Text Classifier Based on Gini Index Weight," iscc,pp.448-453, 11th IEEE Symposium on Computers and Communications (ISCC'06), 2006.

[15] Y. Yang," Noise reduction in a statistical approach to text categorization," In Proceedings of the 18th Annual international ACM SIGIR Conference on Research and Development in information Retrieval (Seattle, Washington, United States, July 09 - 13, 1995), E. A. Fox, P. Ingwersen, and R. Fidel, Eds. SIGIR '95. ACM, New York, NY, 256-263, 1995.

[16] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 412-420, 1997.

[17] Y. Matsuo and M. Ishizuka, "Keyword extraction from a single document using word co-occurrence statistical information," *International Journal on Artificial Intelligence Tools*, vol. 13, no. 1, pp. 157-169, 2004.

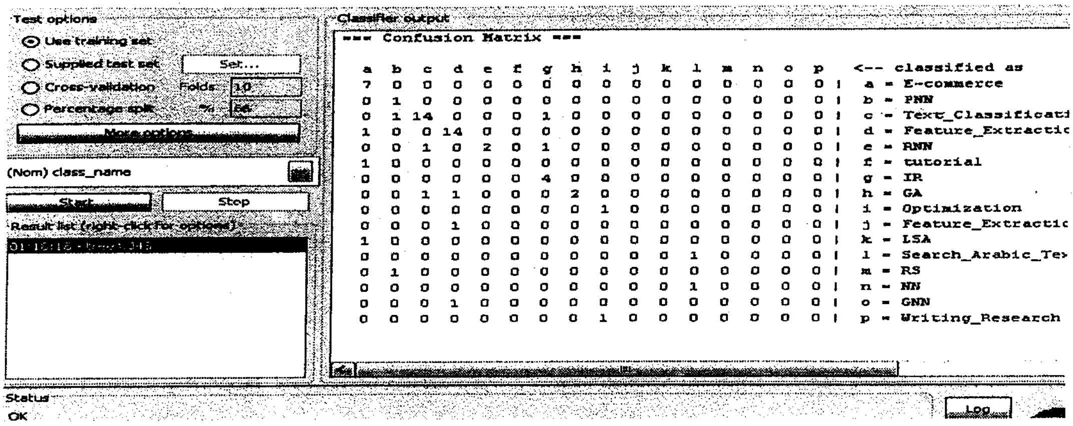
RESULT APPENDIX


training.arff* - win32pad
File Edit View Favorites Tools Help

```
@relation documents

@attribute order {Yes,No}
@attribute draw {Yes,No}
@attribute users {Yes,No}
@attribute attention {Yes,No}
@attribute increase {Yes,No}
@attribute satisfaction {Yes,No}
@attribute online {Yes,No}
                ⋮
@attribute Quality {Yes,No}
@attribute Lyberg {Yes,No}
@attribute Rosenberg {Yes,No}
@attribute Multivariate {Yes,No}
@attribute class_name {E-
commerce,PNN,Text_Classification,Feature_Extraction,RNN,tutorial,IR,GA,Optimization,Feature_
action&GP,LSA,Search_Arabic_Text,RS,NN,GNN,Writing_Research}
@data
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,Yes,
```

Figure 9: the FE output file



Test options

● Use training set
○ Supplied test set
○ Cross-validation Folds 10
○ Percentage split % 66

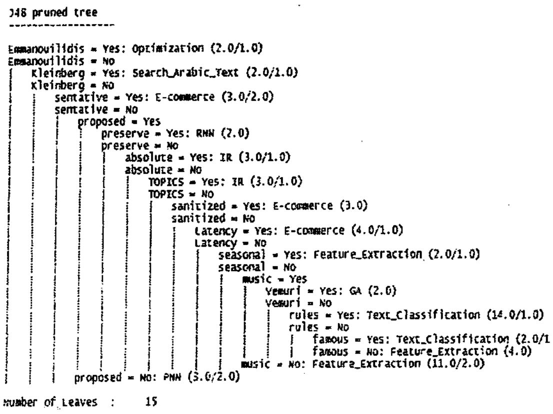
(Nom) class_name

Status
OK

```
=== Confusion Matrix ===

  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p   <-- classified as
  7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  a = E-commerce
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  b = PNN
  0  1 14  0  0  0  1  0  0  0  0  0  0  0  0  0 |  c = Text_Classificati
  1  0  0 14  0  0  0  0  0  0  0  0  0  0  0  0 |  d = Feature_Extractic
  0  0  1  0  2  0  1  0  0  0  0  0  0  0  0  0 |  e = RNN
  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  f = tutorial
  0  0  0  0  0  0  4  0  0  0  0  0  0  0  0  0 |  g = IR
  0  0  1  1  0  0  0  2  0  0  0  0  0  0  0  0 |  h = GA
  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 |  i = Optimization
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 |  j = Feature_Extractic
  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  k = LSA
  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 |  l = Search_Arabic_Te>
  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0 |  m = RS
  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0 |  n = NN
  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0 |  o = GNN
  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0 |  p = Writing_Research
```

Figure 10: the confusion matrix of the classification process

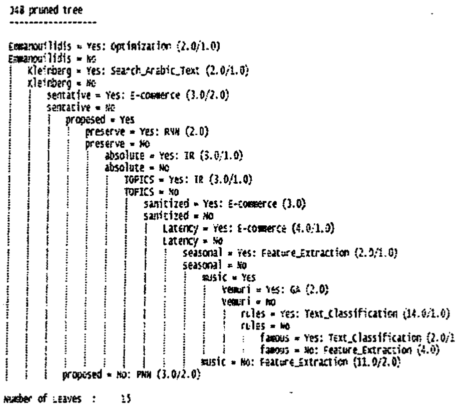


```
J48 pruned tree
------------------

Emmanouilidis = Yes: Optimization (2.0/1.0)
Emmanouilidis = No
|   Kleinberg = Yes: Search_Arabic_Text (2.0/1.0)
|   Kleinberg = No
|   |   sentative = Yes: E-commerce (3.0/2.0)
|   |   sentative = No
|   |   |   proposed = Yes
|   |   |   |   preserve = Yes: RNN (2.0)
|   |   |   |   preserve = No
|   |   |   |   |   absolute = Yes: IR (3.0/1.0)
|   |   |   |   |   absolute = No
|   |   |   |   |   |   TOPICS = Yes: IR (3.0/1.0)
|   |   |   |   |   |   TOPICS = No
|   |   |   |   |   |   |   sanitized = Yes: E-commerce (3.0)
|   |   |   |   |   |   |   sanitized = No
|   |   |   |   |   |   |   |   Latency = Yes: E-commerce (4.0/1.0)
|   |   |   |   |   |   |   |   Latency = No
|   |   |   |   |   |   |   |   |   seasonal = Yes: Feature_Extraction (2.0/1.0)
|   |   |   |   |   |   |   |   |   seasonal = No
|   |   |   |   |   |   |   |   |   |   music = Yes
|   |   |   |   |   |   |   |   |   |   |   Vemuri = Yes: GA (2.0)
|   |   |   |   |   |   |   |   |   |   |   Vemuri = No
|   |   |   |   |   |   |   |   |   |   |   |   rules = Yes: Text_Classification (14.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   rules = No
|   |   |   |   |   |   |   |   |   |   |   |   |   famous = Yes: Text_Classification (2.0/1
|   |   |   |   |   |   |   |   |   |   |   |   |   famous = No: Feature_Extraction (4.0)
|   |   |   |   |   |   |   |   |   |   music = No: Feature_Extraction (11.0/2.0)
|   |   |   proposed = No: PNN (3.0/2.0)

Number of Leaves  :     15
```

Figure 11: the decision tree of the classification process of training set



```
J48 pruned tree
------------------

Emmanouilidis = Yes: Optimization (2.0/1.0)
Emmanouilidis = No
|   Kleinberg = Yes: Search_Arabic_Text (2.0/1.0)
|   Kleinberg = No
|   |   sentative = Yes: E-commerce (3.0/2.0)
|   |   sentative = No
|   |   |   proposed = Yes
|   |   |   |   preserve = Yes: RNN (2.0)
|   |   |   |   preserve = No
|   |   |   |   |   absolute = Yes: IR (3.0/1.0)
|   |   |   |   |   absolute = No
|   |   |   |   |   |   TOPICS = Yes: IR (3.0/1.0)
|   |   |   |   |   |   TOPICS = No
|   |   |   |   |   |   |   sanitized = Yes: E-commerce (3.0)
|   |   |   |   |   |   |   sanitized = No
|   |   |   |   |   |   |   |   Latency = Yes: E-commerce (4.0/1.0)
|   |   |   |   |   |   |   |   Latency = No
|   |   |   |   |   |   |   |   |   seasonal = Yes: Feature_Extraction (2.0/1.0)
|   |   |   |   |   |   |   |   |   seasonal = No
|   |   |   |   |   |   |   |   |   |   music = Yes
|   |   |   |   |   |   |   |   |   |   |   Vemuri = Yes: GA (2.0)
|   |   |   |   |   |   |   |   |   |   |   Vemuri = No
|   |   |   |   |   |   |   |   |   |   |   |   rules = Yes: Text_Classification (14.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   rules = No
|   |   |   |   |   |   |   |   |   |   |   |   |   famous = Yes: Text_Classification (2.0/1
|   |   |   |   |   |   |   |   |   |   |   |   |   famous = No: Feature_Extraction (4.0)
|   |   |   |   |   |   |   |   |   |   music = No: Feature_Extraction (11.0/2.0)
|   |   |   proposed = No: PNN (3.0/2.0)

Number of Leaves  :     15
```

Figure 12: the decision tree of the classification process of test set