Multistrategy Hybrid Text Categorization

Mª Dolores del Castillo[1], J. Ignacio Serrano[1], Mª Paz Sesmero[1]

[1]Instituto de Automática Industrial. CSIC. Arganda del Rey,
28500, Madrid, Spain
{lola, nachosm, mpaz, info}@iai.csic.es
http://www.iai.csic.es

Abstract. The goal of the research described here is to develop a multistrategy classifier system that can be used for documents categorization. The system automatically discovers classification patterns by applying different empirical learning methods on different representations for the documents. These methods are embodied in agents that can solve a problem or different parts of a problem in a parallel manner. Every agent carries out a feature selection based on a genetic search and learns a classification model. In the classification of documents, the system combines the predictions made by all the methods in a novel and effective way. The system relies on an architecture modular and flexible, making no assumptions about the design of agents or the number of agents available. The system flexibility warrants the independence of the type of text and of the application domain.

1 Introduction

Assigning categories to documents is essential to the efficient management and retrieval of information and knowledge. There are a large number of text classifiers that take a single approach based on supervised learning and attack a concrete application domain [2], [3]. The algorithm used is always a key point at design time. Certain learning algorithms are more suitable for some types of text and parts of a problem than for others. Besides, the performance of the selected algorithm depends on the application domain and the features or attributes chosen to represent the documents [4], [5], [6]. So, many experiments are needed to decide the final algorithm and the suited feature set. Moreover, once the algorithm and features are set, the achieved solution could lack of relevant information lost in the mapping from the full document to the feature set.

Choosing the right feature set is critical to successful induction of classification models. Conventional approaches use a general method, based on statistical measurements and stemming procedures, which is independent of the learning algorithm and the application domain for creating the vocabulary of the problem [6], [7]. The approach presented in this paper takes into account the relevance of the vocabulary for every learning algorithm.

The growth of electronically stored text has led to the development of machine learning methods prepared to exploit ungrammatical text. However, most of them are

based on a single strategy and work on a single particular domain. The richness and the redundancy of information present in hypertext documents suits well for a multistrategy learning approach. There are scarce references about multistrategy systems for text classification. Current multistrategy systems [8], [9] combine statistical and symbolic algorithms in a predefined manner by using a common feature extraction stage and so a shared feature set. These systems solve the problem by different ways and take the most confidential one.
The main goal of the proposed system is to obtain the best feasible classification performance for any text categorization task by considering all the current information contained in documents regardless of the document domain. Hypertext documents contain information about document formatting, meta-information, plain text, and hyperlinks [10]. This system relies on a multistrategy hybrid architecture. Next section surveys in detail the capabilities of the architecture. The subsequent section summarizes the critical aspects of this approach. The last section presents the future work in this subject.

2 Basic Multistrategy Hybrid Architecture

The system developed relies on an modular and flexible architecture. Figure 1 shows the modules of the architecture and the information flow. The system is first trained to obtain different classification models by giving a labeled sample of documents which are divided in two groups: training sample and test sample. The learned models are then evaluated over the test sample and predictions made are combined in order to achieve the best classification [8]. The next subsections explain the modules and procedures of the architecture instantiated to hypertext documents.

2.1 Preprocessing Step

The preprocessing step is common to all the agents. The task here is to scan the text of all the training documents and to produce a list of the words or vocabulary contained in them. The preprocessing step begins by removing those words found in the documents that belong to a stop list [4], [6], consisting of words without semantic contents, and applying a stemming procedure [11]. After that, the frequency of occurrence of every valid word is calculated. Next, the value of the frequency is increased depending on the word format: a frequency ten times higher for a word found in the “TITLE” tag, nine to one in the “H1” tag and so on. Words recurring only a few times are not reliable indicators and are removed.
When the system receives a set of hypertext documents, whose first line is the *url* (*u*niform *r*esource *l*ocator) of the document, three vocabularies are generated from every document: one containing the words from meta-text, a second one containing all the words from the plain text together with the *url* words, and the third one containing the words from the hyperlinks. Next, for each vocabulary, several information measurements are calculated: information gain, mutual information, document frequency, chi square [5], and crossover entropy [12]. The words of all of

the new vocabularies are sorted by each measurement and the k best words are retained. The

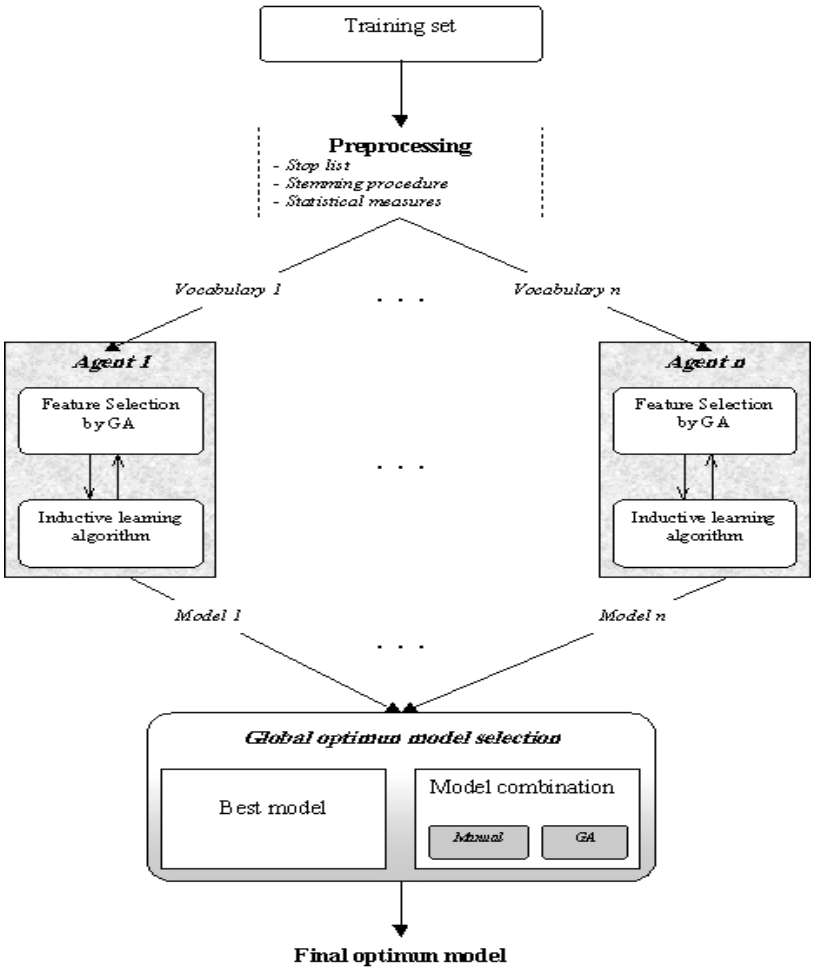

Fig. 1. System architecture

value of k is determined empirically. The words ranked by all these measurements will be the initial feature subsets of every agent.

Although some information is lost in any one feature set, the multiple views of every initial vocabulary will allow better overall performance.

2.2 Structure and Function of the Agents

Since hypertext documents contain different kinds of information, the multistrategy approach suits that each agent to solve a part of the problem with a different input information from the same sample.
The size of vocabularies and the nature of their features suggest different strategies to handle them. Thus, each agent can learn and classify the documents with regard to different views of the vocabularies obtained from the preprocessing step. At the current stage of development of the multistrategy system, the decision about which type of information is processed by each agent is made in a non autonomous way. The designer applies his/her experience about the relation between type of information and biases embodied into learning methods.
Every agent receives a vocabulary and carries out the following tasks:
Feature selection. The agent takes a tentative feature set sorted by different statistical measurements and obtains the final feature set after applying a genetic algorithm.
Empirical learning. Given the feature set, the agent works on the training documents, represented according to the feature set, in order to induce the classification model.
Testing. The agent applies the model inferred over the test set and calculates the classification accuracy.

2.2.1 Feature Selection Using Genetic Algorithms. The goal of this step is to reduce the feature space size with the lowest loss of classification accuracy. There are many ranking techniques that assign a score to the features based on a certain criterion and then select the first k. The performance of these techniques are very sensitive to the score criterion. In order to avoid this situation, the agents apply genetic algorithms to achieve an optimal feature set in a large and criterion independent search space. The experiments carried out show that a significant departure from approaches taking into account an universal feature selection yields better results [13].
Genetic algorithms intend to simulate the natural evolution process. Coming from an initial population of individuals or chromosomes, a new generation is created by combining or modifying the best individuals of a previous generation. The process ends when the best solution is achieved or after a number of fixed generations. The task of combining and modifying the best chromosomes is performed by two basic operators: crossover and mutation. Basic crossover operator splits two chromosomes in two parts. Then, the second parts of both chromosomes are exchanged producing two new chromosomes. The basic mutation operator changes one gene value of a chromosome. The proportion of chromosomes involved in crossover and mutation operations is determined by crossover and mutation probabilities, respectively.
The application of genetic algorithms to feature selection implies to establish the chromosome representation, the definition of crossover and mutation operators fitted for the chromosome representation, and the fitness function used to determine the best chromosomes of a population. These aspects have been defined as follows:

Chromosome Representation. Each feature subset obtained in preprocessing step is a chromosome. Each gene represents the position of a feature in the original vocabulary. For example, if the input vocabulary is *{bye, see_you, hello,*

good_morning, good_afternoon} and the feature subset found by applying chi-square and $k = 3$ is *{see_you, bye, good_afternoon},* the chromosome representation would be *'215'*. Chromosome length is fixed to the k best values of the different statistical measurements. Population size matches the number of views of a vocabulary.

Operators. Crossover and mutation operators are defined as follows:

- Crossover. This operator chooses a point randomly in the two selected parents chromosomes and exchanges the right segments of the parents to create a new offspring. For example, if the parents '2137' and '5264' are selected, and the crossover point is the middle point of the chromosome, the resulting offspring will be '21**64**' and '52**37**'. This operator allows the best features remain constant and only the worst features can be changed.

- Mutation. This operator modifies one randomly selected gene value of a chromosome by switching its value to one of the possible different values. If the vocabulary is *{bye, see_you, hello, good_morning, good_afternoon}* and the second gene of the chromosome '214' has been selected to mutate, the next possible values of this gene would be 3 or 5.

Crossover and mutation probabilities are set empirically.

Fitness Function. For each chromosome, the agent proposes a model which is applied to the test documents represented relative to the chromosome being evaluated. The predictive value of the model, i.e., the fraction of test documents correctly classified, on the test set is the fitness function value for the chromosome.

2.2.2 Learning Methods. Once every agent finds its own optimized feature set, and the training sample is represented relative to it, the final classification model can be learned. Since there are three kinds of redundant information in hypertext documents, the architecture is currently composed of three agents: meta-information agent, link information agent, and plain information agent. The selected learning methods are:

- Naïve Bayes [2], [3], [15] for plain text agent, since the vocabulary size is the largest and the noisiest.

- Decision trees [2] for meta-information agent. The vocabulary size is small and the statistical measurement scores are high. The meta-information is very accurate and with very little noise.

- Rule discovery [1], [9], [14] for link agent. The information contained in links is very rich due to they describe the manner in which documents are connected and web net is formed. The feature set size is the smallest and the rules discovered will be small too. Moreover, the rules can express all the richness of the information in an understandable manner.

Once every classification model has been learned, the models are applied to the test sample and several predictive measurements are calculated: predictive value and number of failing negatives and failing positives. Choosing the most confident classification model is based on the importance given to these three measurements. There are some contexts in which a high predictive value and the lowest number of failing negatives are the most suited option -xenophobia, pornography- whereas in other contexts it is more convenient to achieve a low number of failing positives.

2.3 Results Combination

When the system receives unlabeled documents to be classified in some of the previously learned classes, the different kinds of texts are represented according to the final vocabularies. Although meta-information and links give more accurate information about the category of a document, the system could assign the highest confidence to the prediction of the plain text agent since there are many hypertext documents without links and meta-information. The concrete confidence of the predictions can be established basing on several trial and error experiments, or automatically by heuristics. So, given the predictions made by individual models from a particular unlabeled document, there exist two possibilities:

- To take the model with best results on classification as the optimal final solution.
- To take a combination of the models as the final solution. The combination can be determined as an average of the predictions or as a weighted sum, where the weights can be set manually or by a genetic algorithm.

The system developed has been successfully applied to classify web documents from three categories: interactive gambling, games and music. The number of processed documents in the training set has been: 5000 for gambling documents, 2000 for music documents, and 2000 for game ones. This training set is very noisy and many documents are error pages downloaded from servers when the page request fails. The precision results are showed in Table 1.

Table 1. Precision results. The table shows the precision (proportion of examples correctly classified) by the system for several test sets

Test Set	Precision
1500 gambling documents 1500 games documents	82%
1500 gambling documents 1500 music documents	96%
2000 gambling documents	97%
2000 games documents	79%
2000 music documents	97%
1500 games-gambl.-music	89%

At present, the procedure used to integrate several classification models is to take the model with the highest confidence value. This model is the one obtained by the plain text classifier. Meta-information and hyperlinks models take a lower confidence value because many processed documents do not have these kinds of information. The precision decreases when games documents are present in the test sets because many documents are very similar to gambling documents and the system classify them in gambling category. The system performs quite well for the remaining categories. The noise reduction could improve the results reached.

3 Discussion

The architecture presented is a combination of a variable number of agents. Depending on the specific task certain agents could be added or removed. The modularity makes the system to be adaptable to any particular context. Moreover, the combination of the predictions done by the agents allows the system to consider all the available information for classification in an optimal way.

The computational cost needed to achieve the final model can seem very expensive in terms of consumed time. However, if all the agents work in a parallel way, the cost is the one of the slowest agent. On the other hand, the classification time is similar to the pure systems.

Learning methods embodied in the agents are applied twice: first, for calculating fitness function values in feature selection stage, and second, for learning the classification model from learned feature sets. Since the population size and the chromosome length are both small values, the computational cost of the double operation is worthless if it is compared to the advantages of having a relevant feature set.

The system can process hypertext documents and grammatical documents. For grammatical documents, all the agents will receive the same input information and the system will solve the same problem by several learning methods.

Another way to process hypertext documents, different from the one shown in this paper, is to consider every kind of information as grammatical documents and to apply all the agents to it in an iterative manner.

The models obtained from decision trees and rule discovery allow that the classification results can be expressed in an intelligible manner. Parametric learning methods like Naïve Bayes prevents understanding the classification model. In order to make easier the comprehension of the classification results of this method, the system gives the set of the most relevant features of every category.

4 Future Work

The future work will be mainly focused on the discovery of two heuristics. The first one, based on the training set size, initial feature set and number of categories, will be used to lead the system to autonomously select the inductive learner for each kind of

information in hypertext samples, or to select the number and types of agents in grammatical samples.
The second heuristic, based on the predictive classification of agents and the relevance of the information processed by them, will be used to lead the system to find automatically the best way to combine several predictions.
Another future goal is to use the system not only for general document categorization but also for personalized filtering of free text. This task will allow to investigate and develop new procedures that will contribute to a more complete and flexible system.

Acknowledgments

This work is being supported by Spanish Ministry of Science and Technology with the project FIT-070000-2001-193 and the enterprise Red Educativa S.A.

References

1. Castillo, Mª. D. del, Sesmero, P., "Perception and Representation in a Multistrategy Learning Process", *Learning'00. ISBN 84-89315-19-1*. (2000)
2. Dumais, S. T., Platt, J., Heckerman, D, and Sahami, M. "Inductive Learning Algorithms and Representation for Text Categorization". In *CIKM-98: Proceedings of the Seventh International Conference on Information and Knowledge Management.* (1998)
3. Lewis, David D., Ringuette, Mark. "A Comparison of Two Learning Algorithms for Text Categorization", *Symposium on Document Analysis and IR, ISRI, Las Vegas.* (1994)
4. David D. Lewis. "Feature selection and feature extraction for text categorization". In *Proceedings of Speech and Natural Language Workshop*, pages 212-217. Defense Advanced Research Projects Agency, Morgan Kaufmann, February (1992)
5. Yang, Y., Pedersen J.P. "A Comparative Study on Feature Selection in Text Categorization". *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pp412-420, (1997)
6. Grobelnik, M. & Mladeni'c, D. "Efficient Text Categorization". *Proceedings of the ECML-98 Text Mining workshop.* (1998)
7. Dunja Mladenic. "Feature Subset Selection in Text-Learning". *European Conference on Machine Learning* (1998)
8. Freitag, Dayne. "Multistrategy Learning for Information Extraction". *Proceedings of the 15th International Conference on Machine Learning*, (1998)
9. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam & S. Slattery "Learning to Knowledge Bases from the World Wide Web". To appear in *Artificial Intelligence*, (1999)
10. Esposito, Floriana, Malerba, Donato, Di Peace, Luigi, Leo, Pietro. "A Learning Intermediary for the Automated Classification Web Pages".
11. Porter, M.F., "An algorithm for suffix stripping", *Program*, 14(3) :130-137, (1980)
12. Mladenic, D., Grobelnik, M. "Feature selection for classification based on text hierarchy". *Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery CONALD-98.* (1998)
13. J. Yang and Vasant Honavar. "Feature subset selection using a genetic algorithm". *IEEE Intelligent Systems and their Applciations. 13(2). 44-49*, (1998)

14. William W. Cohen. “Text categorization and relational learning”. In *Proceedings of the Twelfth International Conference on Machine Learning*, Lake Tahoe, California, (1995)
15. Kamal Nigam, Andrew McCallum, Sebastian Thrun and Tom Mitchell. “Learning to Classify Text from Labeled and Unlabeled Documents”. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pp. 792-799. (1998)