International Journal of Soft Computing 3 (1): 58-62, 2008
ISSN: 1816-9503
© Medwell Journals, 2008

Similarity-Based Techniques for Text Document Classification

S. Senthamarai Kannan and N. Ramaraj
Department of Information Technology, Thiagarajar College of Engineering,
Madurai-15, Tamilnadu, India

Abstract: With large scale text classification labeling a large number of documents for training poses a considerable burden on human experts who need to read each document and assign it to appropriate categories. With this problem in mind, our goal was to develop a text categorization system that uses fewer labeled examples for training to achieve a given level of performance using a similarity-based learning algorithm and thresholding strategies. Experimental results show that the proposed model is quite useful to build document categorization systems. This has been designed for a small level implementation considering the size of the corpus being used. This can be enhanced for a larger data set and the efficiency can be proved against the performance of the presently available methods like SVM, naïve bayes etc. This approach on the whole concentrates on categorizing small level documents and does the assigned task with completeness.

Key words: Similarity-based classifier, text classification, feature selection, knowledge-based strategies

INTRODUCTION

Text Mining techniques are applied to Large collections of documents from various sources such as news articles, research papers, books, digital libraries, e-mail messages and Web pages, library database, etc. The Data stored in database is usually semi-structured. Traditional information retrieval techniques (Damais *et al*., 1998) become inadequate for the increasingly vast amounts of text data in database. Text categorization is the process of automatically assigning predefined category labels to new text documents. With the increasing availability of documents in digital form, automatic text categorization faces a bigger challenge. Text categorization is now being applied in many fields, including document filtering, topic search engine, web resources classification, automatic indexing in information retrieval system and many applications requiring document organization.

Supervised learning is a machine learning technique for creating a function from training data. The training data consist of pairs of input objects and desired outputs. The output of the function can be a continuous value, or can predict a class label of the input object. The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples. To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way. The parallel task in human and animal psychology is often referred to as concept learning. Supervised learning can generate models of two types. Most commonly, supervised learning generates a global model that maps input objects to desired outputs. In some cases, however, the map is implemented as a set of local models.

Lee *et al.* (2002) describe the investigation and development of supervised and semi supervised learning approaches to similarity-based text categorization systems. It uses a small number of manually labeled examples for training and still maintains effectiveness with good categorization performance. Sebastiani *et al.* (2004) views Text categorization as the task of automatically sorting a set of documents into categories from a predefined set. This task has several applications, including automated indexing of scientific articles according to predefined thesauri of technical terms, filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogues of Web resources, spam filtering, identification of document genre, authorship attribution, survey coding and even automated essay grading. Automated text classification is attractive because it frees organizations from the need of manually organizing document bases, which can be too expensive, or simply not feasible given the time

Corresponding Author: S. Senthamarai Kannan, Department of Information Technology, Thiagarajar College of Engineering, Madurai-15, Tamilnadu, India

constraints of the application or the number of documents involved. The accuracy of modern text classification systems rivals that of trained human professionals, thanks to a combination of Information Retrieval (IR) technology and Machine Learning (ML) technology. Basili *et al.* (2000) presents a statistical inference technique for profile-based classification. Extensive testing of models with respect to weighting policies and inference algorithms has been carried out. They proposed relatively simple method and compared against other more complex models (i.e. Logistic Regression) and reported better performances in large scale and dynamic classification scenarios.

PROBLEM STATEMENT

Supervised approaches to text categorization usually require a large number of training examples to achieve a high level of effectiveness. Training examples used in supervised learning approaches require human involvement in the labeling of each example. So, we applied these uncertainty selective sampling methods to the proposed classifiers. However, the sampling methods are quite general and could be used for any machine learning approaches. The proposed approach uses the frequent term frequencies for each document and weighted feature vector calculated by averaging the frequencies over all training documents. The document is given as input in trainer for learning and then categorizes the document using the custom-developed Runner module.

Text categorization algorithms assign texts to predefined categories. The study of such algorithms has a rich history dating back at least 40 years. In the last decade or so, the statistical approach has dominated the literature. The essential idea is to infer a classifier from a set of labeled documents.

Standard statistical classification tools such as Naive Bayes, logistic regression and decision trees are immediately relevant and have been used with some success. Researchers in statistical text categorization face two particular challenges. First, the scale of text categorization applications causes problems for many standard learning algorithms (Wittem and Frank, 2005).

Documents are represented by vectors of numeric values, with one value for each word that appears in any training document. Document feature vectors therefore are typically of dimension 105-106 or more. More recently, increased computing power and a better theoretical understanding of classifier complexity have enabled algorithms (Lewis *et al.*, 1996; Lee *et al.*, 2002) to learn less restricted and thus more accurate classifiers while simultaneously avoiding overfitting and maintaining sufficient speed. The challenge of integrating knowledge with learning has attracted much less attention in text categorization. This has motivated our interest in similarity-based learning algorithms. Similarity-based learning algorithms allow the user to specify possible parameter values of the learned classifier. This not only provides one solution to the over fitting problem, but the prior also provides a mathematically well-justified way to allow domain knowledge to influence the parameter values that result from learning. Our focus here is on developing a similarity-based learning algorithm which avoids over fitting, is computationally efficient and gives state-of-the-art effectiveness. As such, we focus on experimental comparisons of competing models.

PROPOSED APPROACH

Categorization and learning: We now describe the process of text categorization where there is user feedback that initiates the learning process. Once initial-learning has been completed, the resulting knowledge base can be used to predict the categories of a new document. The results of prediction are presented to human expert and then, if there is any feedback on this prediction, learning is initiated to update the current knowledge base.

Like the raw documents in the initial learning model of Fig. 1, any real-world document first needs to be transformed to a representation in the preprocessor. To make the binary decision between a document and each category, the predictor computes similarity scores for every document-category pair, based on the document representation and classifiers. The decision on the categories for the documents is made by threshold-ing these similarity scores. A wide range of measures can be

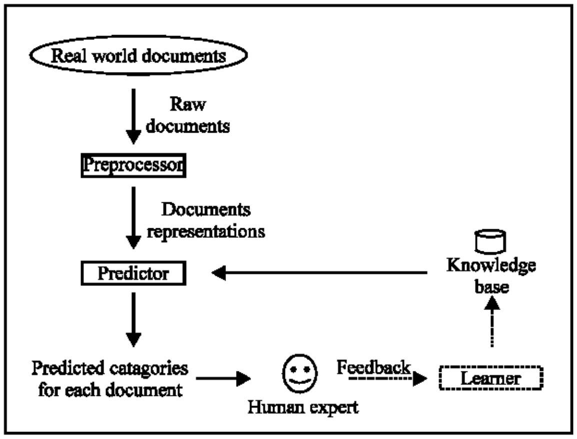

Fig. 1: Similarity-based categorization framework

used for the similarity computation in the predictor. One of the most widely used similarity measures is the cosine of the angle between two vectors, computed as the inner product of two normalized vectors.

We choose this measure since it performs well in most text categorization literature and exploration of this aspect is not the core of our research. Any feedback from human experts should be incorporated quickly into the current knowledge base to make it more accurate. As well as the simple category information on new documents, the expert may modify the predefined categories: Adding new categories or deleting some predefined categories. Some approaches allow human experts to annotate documents indicating or writing some important terms in a given document. The creation and manipulation of such annotated documents is an important area in the field of text categorization.

Feature selection: Feature selection is an essential part of text classification. Document collections have 10,000 to 100,000 or more unique words. Many words are not useful for classification. Restricting the set of words that are used for classification makes classification more efficient and can improve generalization error. We describe how the application of Information Gain to feature selection for multi class text classification is fundamentally awed and compare it to a statistics-based algorithm which exhibits similar difficulties.

A major problem in the text categorization task is the high dimensionality of the feature space. Even for a moderately sized corpus, the text categorization task may have to deal with thousands of features and tens of categories. Most sophisticated machine learning algorithms applied to text categorization cannot scale to this huge number of features. The learning process on such a high feature space may require unacceptably long processing time and need a large number of training documents since all features in the representation are not necessarily relevant and beneficial for learning classifiers. As a result, it is highly desirable to reduce the feature space without removing potentially useful features for the target concepts of categories. The stop-word removal and word stemming methods are dimensionality reduction methods. However, the main advantage of applying these methods is the reduction in the size of each document, not on the size of the full feature set. As a result, high dimensionality of the feature space may still exist even after applying them. They still leave the dimensionality prohibitively large for machine learning algorithms (especially for some learning algorithms that are trying to use the inter-relationships among features). This means that text categorization needs more aggressive methods to reduce the size of overall feature set.

A difficulty with most sophisticated feature selection methods (Yang and Pederson, 1997) is that they are very time-consuming and so it is not practical or possible to perform the feature selection process whenever new training examples are available. This high time complexity is a critical problem for our text categorization system in which one of main characteristics is to quickly incorporate any new information into the current kowledge base. A text feature selection algorithm should select features that are likely to be drawn from a distribution which is distant from a class-neutral distribution. Neither of the two algorithms do this. We describe a framework for feature selection that encapsulates this notion and exposes the free parameters which are inherent in text feature selection. Information Gain (IG) is a commonly used score for selecting words for text classification For each word, IG measures the entropy difference between the unconditioned class variable and the Class variable conditioned on the presence or absence of the word,

$$IG = H(C) - H(C \mid W_k) = \sum_{c \varepsilon C_{wk}} \sum_{\in \{0,1\}} p(c, w_k) \log \frac{p(c \mid w_k)}{p(c)}$$

This score is equivalent to the mutual information between the class and word variables, IG = I(C;Wk). Hence, this score is sometimes called mutual information. The probabilities correspond to individual word occurrences. wk = 1 corresponds to the occurrence of word wk. wk = 0 corresponds to the occurrence of some other word.

Initial-learning: The initial-knowledge-base can be used for predicting categories of new documents and for uncertainty selective sampling of informative examples for future training. In this model, a small number of unlabeled raw documents should be randomly selected. These are presented to human experts for actual labels. Then, in the

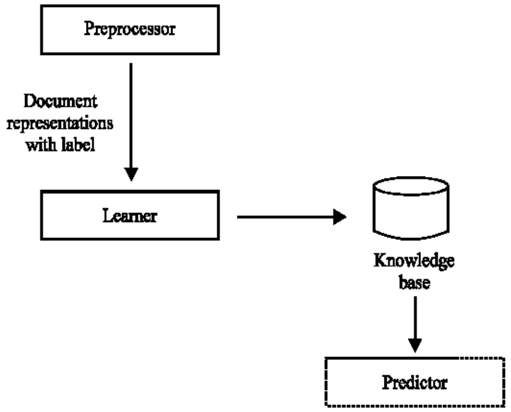

Fig. 2: learning model for text categorization

preprocessor, these labeled documents are transformed to a representation that is readable and suitable for a machine learning algorithm, in the 'learner'. The common representation adopted by most machine learning algorithms is the vector space model. In this representation method, each document is represented by a list of extracted features (words, terms, tokens, or attributes). Their associated weights are computed from either the existence or the frequency of each feature in a given document. Various techniques from natural language processing should or could be applied to extract informative features (Fig. 2).

The common techniques used for feature extraction are the stop-list to remove common words such as the, a, of, to, is and was and the word stemming to reduce different words from the same stem, for example children and childhood→child. Then, each feature can be weighted by using a Boolean vector indicating if the feature appeared in a given document or a numeric vector derived from the frequency of occurrence.

Applying other complicated text processing techniques, such as pos-tagging and n-grams (phrase) generation may also improve performance. However, there is a trade-off between their substantial preprocessing time and small benefits in text categorization.

Formation of vocabulary set: The vocabulary as the name suggests gives a comprehensive list of all the keywords available in the training corpus. Each of the document that is being given as input to train contains a list of keywords. All such files containing the corresponding keywords are listed and are concatenated together to form the actual vocabulary file. No special feature reduction or dimensionality reduction needs to be done in this vocabulary.

Learning using trainer: The learning section in this project involves the feature vectors obtained in the previous phase as inputs, used to train the tool about the available categories and their corresponding features. This sets up the whole project towards the next and final phase of predicting the type of the input document (Fig. 3).

Categorization using runner: The actual task of the Runner basically works on the input knowledge base obtained from the training documents given as input to the previous phase. The training process gives information about the documents types which is being updated to the knowledge base (Fig. 4).

With the help of the vocabulary it forms the feature vector for the inputted test document. Then it divides the feature vector into separate sets according to the number of keywords available in the testing document.

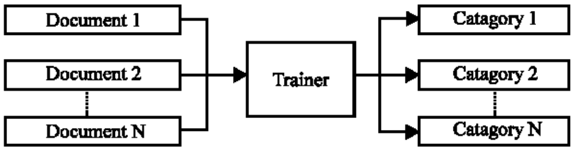

Fig. 3: Learning using trainer

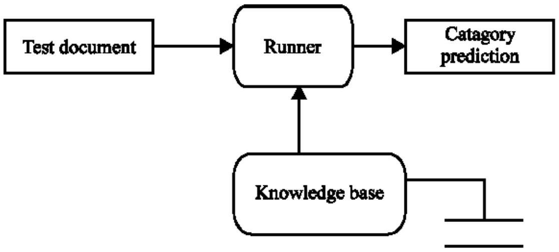

Fig. 4: Categorization using runner

Table 1: Contingency table for category ci

		Label by human expect	
Category *ci*		Yes is correct	No is correct
Label by the system	Predicted yes	TP_i	FP_i
	Predicted no	FN_i	Tn_i

RESULTS AND DISCUSSION

We conducted extensive comparative experiments on standard collection for text classification the Reuters-21578 (http://www.daviddlewis.com/resources/testcollections/reuters21578/). While a number of different conventional performance measures are available for the effectiveness evaluation for text categorization, the definition of almost all measures is based on the same 2×2 contingency table model that is constructed as shown in Table 1. In this table, 'YES' and 'NO' represent a binary decision given to each document dj under category Ci Each entry in the table indicates the number of documents of the specified type:

TP_i: The number of true positive documents that the system predicted were YES and were in fact in the category ci.

FP_i: The number of false positive documents that the system predicted were YES, but actually were not in the category ci.

FN_i: The number of false negative documents that the system predicted were NO, but were in fact in the category ci.

TN_i: The number of true negative documents that the system predicted were NO and actually were not in the category ci.

Table 2: Contingency table for categorization

Category *ci*		Yes	No
Label by the system	Predicted yes	1800	200
	Predicted no	200	1800

The standard performance measures for classic information retrieval research are recall and precision that has been also frequently adopted for the evaluation of text categorization.

The *precision* is the proportion of documents which are both predicted to be YES and are actually correct, against all documents that are predicted YES.

The Recall measures the proportion of documents that are predicted to be YES and correct, against all documents that are actually correct.

These measures are computed as follows.

$$\text{Recall} = \frac{TP_i}{TP_i + FN_i} \text{ if } TP_i + FN_i > 0$$

$$\text{Precision} = \frac{TP_i}{TP_i + FN_i} \text{ if } TP_i + FN_i > 0$$

For a sample of 2000 documents, the recall and precision measures are shown in Table 2.

Tp_i = 1800 , TN_i =1800 , FP_i = 200 , FN_i = 200 then Recall and Precision is calculated as Recall = 0.9 Precision = 0.9. Other performance measures that are purely based on the contingency table are Accuracy and Error. The accuracy and error are defined as the proportion of documents that are correctly predicted and the proportion of documents that are wrongly predicted

They are defined as follows:

$$\text{Accuracy} = \frac{TP_i +}{|D|} \text{ where } |D| = TP_i + FP_i + FN_i + TN_i > 0$$

$$\text{Error} = \frac{FP_i +}{|D|} \text{ where } |D| = TP_i + FP_i + FN_i + TN_i > 0$$

Tp_i = 1800 , TN_i =1800 , FP_i = 200 , FN_i = 200 then Accuracy and Error is calculated as Accuracy = 0.90 Error = 0 .10

CONCLUSION AND FUTURE WORKS

Overall, the performance of the algorithm was slightly lower than expected, although the results are sufficiently promising to warrant further investigation. For example, we expected that truncating the vocabulary would have helped, by reducing the danger of over fitting, but that was clearly not the case, although smaller vocabulary speeds up the classification. Our heuristics for deducing the document categorization in the reuters data set could be improved further and the vocabulary is further refined to reduce the redundant data. The work at present contains 3 modules running separately as 3 programs. This can be enhanced to a single package that does the preprocessing, feature set formation ,learning and the prediction processes together. The presently available benchmark techniques have shown good efficiencies in this area. This project can be made to be implemented for the whole of the 21578 documents available in the reuters corpus and then compared against the performance of the presently available methods like SVM, Rocchio algorithm, Bayes, Naïve Bayes etc.

REFERENCES

Basili, R., A. Moschitti and M.T. Pazienza, 2000. Robust inference method for profile-based text classification.

Dumais, S.T., J.C. Platt, D. Hecherman and M. Sahami, 1998. Inductive Learning Algorithms and Representations for Text Categorization, pp: 148-155.

http://www.daviddlewis.com/resources/testcollections/reuters21578/.

Ian, H.W. and E. Frank, 2005. Practical Machine Learning Tools Techniques.

Lewis, D.D. and M. Ringuette, 1994. A comparison of two learning algorithms for text categorization.

Lewis, D.D., R. Schapire, J.P. Callan and R. Papka, 1996. Training algorithms for linear text classifiers.

Lee, K.H., J. Kay and B.H. Kang, 2002. Lazy Linear classifier and Rank-in-ScoreThreshold in Similarity-Based Text classification.

Sebastiani, F., 2002. Machine learning in automated text Categorization. ACM Computing Surveys, 34: 1.

Yang, Y. and J. Pederson, 1997. Feature Selection in Statistical Learning of Text Categorization.