Text Categorization Using Co-Trained Support Vector Machines with Both Lexical and Syntactic Information

Seong-Bae Park **Jangmin O** **Byoung-Tak Zhang**
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
{sbpark,jmoh,btzhang}@bi.snu.ac.kr

Keywords : Text Categorization, Co-Training, Support Vector Machines, Syntactic Information, Text Chunking

1 Introduction

Automatic text categorization is an important research area in information retrieval and has a great potential for many applications handling text such as routing and filtering. Its aim is to assign a given document to the predefined category to which it belongs. Most of machine learning algorithms applied to text categorization use a simple *bag of words* representation of documents where each feature corresponds to a single word. That is, they use only the distribution of content words but ignore another important factor, *linguistic information* underlying the given documents.

Each document has its own traits in the style. The syntactic information is one of the best measures to capture the stylistic divergence among the different kinds of documents. Although the syntactic features can give much information in categorizing the documents, they are not widely used due to their lack of formal definition and complicated representation. In addition, unfortunately, the current NLP (natural language processing) techniques are not able to provide accurate results in syntax analyzing. However, some studies show that text chunking can give enough information on syntax analysis instead of full parsing to provide syntactic information [4].

Text chunks are nonoverlapping segments in a sentence, and the chunking is a tagging task, where each word in a sentence is assigned a tag which indicates whether this word is inside or outside of a specific chunk type. Many kinds of machine learning algorithms using only local information show successful performance in text chunking [2]. Support Vector Machines among them show the best performance, about 93.48 in F-score. Thus, we can obtain the accurate chunk information using SVMs, though we can not get the accurate syntactic information.

One of the native problems in text categorization is that there are a great number of inexpensive unlabeled documents while there are a few labeled documents, since labeling of documents must be done by human experts. The co-training algorithm is one of the successful algorithms handling unlabeled examples [1]. It is in general applied to the problems where there are two distinct views of each example in the dataset. It learns separate classifiers over each of the views, and augments a small set of labeled examples incorporating unlabeled examples. Its final prediction is made by combining their predictions to decrease classification error. The larger is the variance of the classifiers when both classifiers are unbiased, the better is the performance of the algorithm [5]. Since the co-training uses two classifiers with distinct views, its performance will be better than any single classifier.

The co-training algorithm uses two distinct views V_1 and V_2 when learning from labeled and unlabeled data, and incrementally upgrade classifiers (h_1 and h_2) over each view. Each classifier is initialized with a few labeled examples. At every iteration, each classifier chooses $p + n$ unlabeled examples to add them to the labeled set of examples, L. The selected unlabeled examples are those which each classifier can determine their label with the highest confidence. After that, the classifiers are trained again using the augmented labeled set. The final output of the algorithm is given as a combination of the two classifiers. Given an example $\mathbf{x}$ to be classified, the probability of the possible class c_j is determined by multiplying two posterior probabilities. The class c^* of $\mathbf{x}$ is set to the one with the highest probability:

$$c^* = \arg\max_{c_j \in C} \left(P(c_j|\mathbf{x}) = P_{h_1}(c_j|\mathbf{x}) P_{h_2}(c_j|\mathbf{x}) \right),$$

where C is the set of all possible classes.

2 Two Views for the Co-Training Algorithm

2.1 Two Views

Most applications of the co-training algorithm are on web page classification, because there are two natural distinct views for the web pages, which are a content view and a link view. However, it is not clear how to construct two independent views for the normal documents without link information.

One possible view for text categorization in the co-training is to treat each document as a vector whose elements are the weight to the vocabulary. Most machine learning algorithms applied to text categorization adopt this representation. The drawback of this representation is that (i) it assumes that each word in the document is independent each other, and (ii) it ignores much linguistic information underlying in the document.

Stamatatos showed experimentally that the syntactic information among various kinds of linguistic information is a reliable clue for text categorization [4]. One additional benefit in using syntactic information for text categorization is that it is somewhat independent from term weights. The current natural language processing techniques, unfortunately, are not able to provide accurate syntactic analysis results. However, the *text chunks* are good features enough to provide syntactic information for text categorization. The chunks are obtained with high accuracy with superficial investigation instead of full parsing.

Therefore, we can define two distinct views for normal documents, so that the co-training algorithm can be naturally applied to categorizing them. The two views are:

- **Lexical Information**
  Most machine learning algorithm applied to automatic text categorization are based on *tf·idf*, a commonly used term weighting scheme in information retrieval. The *tf* factor is the estimation of the occurrence probability of a term if it is normalized, and the *idf* is the amount of information related to the occurrence of the term.

- **Syntactic Information**
  Each document is represented in a vector in which the elements are syntactic features, and the features are derived from text chunking. This information can support finding particular or specific style of the documents.

2.2 Text Chunks

A document is represented in a vector whose elements are chunk information. We consider only five types of chunks: *NP*[1], *VP*, *PP*, *ADVP*, and *O*. Table 1 shows the features used to represent documents. Top five features represent how often the grammatical phrases are used in the document, the following five features implies how long they are, and the final feature means how long a sentence is on the average. That is, every document is represented in a 11-dimensional vector.

Since the documents used to text categorization are raw, they must be chunked in the preprocessing

[1] *NP* represents a noun phrase, *VP* a verb phrase, *PP* a prepositional phrase, *ADVP* a adverb phrase, and *O* implies none of *NP*s, *VP*s, *PP*, and *ADVP*s.

Feature	Description
SF1	detected NPs / total detected chunks
SF2	detected VPs / total detected chunks
SF3	detected PPs / total detected chunks
SF4	detected $ADVP$s / total detected chunks
SF5	detected Os / total detected chunks
SF6	words included in NPs / detected NPs
SF7	words included in VPs / detected VPs
SF8	words included in PPs / detected PPs
SF9	words included in $ADVP$s / detected $ADVP$s
SF10	words included in Os / detected Os
SF11	sentences / words

Table 1: Syntactic features for text categorization.

step. In order to chunk the sentences in the documents, the lexical information and the POS (part of speech) information on the contextual words are used. The chunks are determined by Support Vector Machines trained with the dataset of CoNLL-2000 shared task[2]. In addition, Brill's tagger is applied to determine POS of each word in the documents of Reuters-21578.

2.3 Support Vector Machines for Text Categorization

For the classifiers in the co-training algorithm, Support Vector Machines are adopted in this paper, which show significant improvement over other machine learning algorithms when applied to text filtering or categorization problems [3]. At each iteration of the co-training, the most confident $p + n$ examples are selected from the pool of unlabeled examples. The SVMs provide a natural way to calculate the confidence. The margin m for an unlabeled example $\mathbf{x}_i$ is defined as

$$m = y_i(\mathbf{w} \cdot \mathbf{x}_i + b), \qquad (1)$$

where $y_i \in \{-1, +1\}$ is the label predicted by the hyperplane with the trained parameters $\mathbf{w}$ and b. That implies, the margin can be considered to be a distance from $\mathbf{x}_i$ to the hyperplane, assuming that the predicted label is correct. The more distant $\mathbf{x}_i$ lies from the hyperplane, the more confident it is to predict the label of $\mathbf{x}_i$. Since SVMs are not probabilistic models, the final prediction is made by the classifier whose margin is larger than another one.

Class	Accuracy	Increase
earn	96.61%	1.31%
acq	95.21%	1.48%
money-fx	97.12%	0.97%
grain	95.51%	0.00%
crude	97.67%	0.67%
trade	98.42%	0.63%
interest	97.67%	0.49%
ship	98.58%	0.43%
wheat	99.15%	0.24%
corn	99.27%	0.09%
Average	97.52%	0.63%

Table 2: The accuracy improvement by using additional syntactic information.

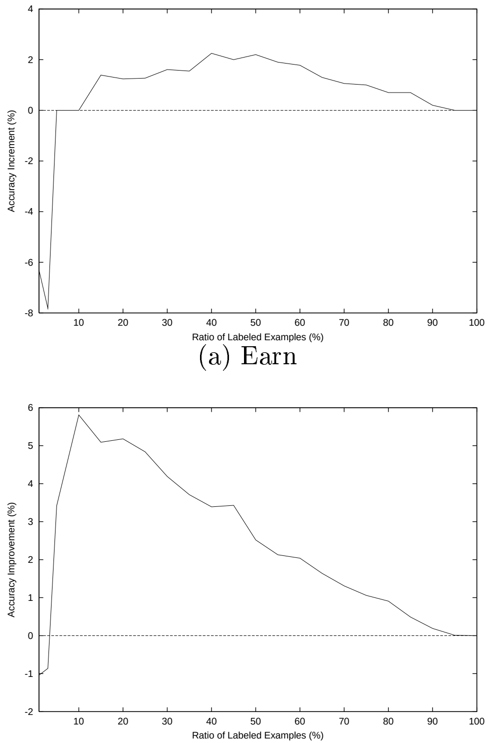
3 Experiments

We use the Reuters-21578 corpus as a data set which is most commonly used benchmark corpus in text categorization. It consists of the Reuters newswire articles, and has 135 kinds of topics while only major 10 of them are used for experiments. There are three versions to divide this corpus into a training set and a test set: "ModLewis", "ModApte", and "ModHayes". Among them "ModApte" which is most widely used is employed in this paper. In this version, there are 9603 training documents, 3299 test documents, and 27863 unique words after stemming and stop word removal.

When we do not consider unlabeled examples, the effect of using syntactic information is given in Table 2. The accuracy is increased by 1.48% at maximum for 'acq', and by 0.63% on the average. Even though we can expect high accuracy with only lexical information $tf \cdot idf$, the additional improvement on accuracy is obtained. This improvement is caused by the syntactic information.

Figure 1 shows the effectiveness of unlabeled ex-

[2] http://lcg-www.uia.ac.be/conll2000/chunking



(a) Earn

(b) Acq

Figure 1: The improvement in accuracy by using additional unlabeled examples.

amples involved in the co-training algorithm. The X-axis represents the ratio of labeled examples to total examples, while Y-axis is the accuracy improvement by unlabeled examples. For ‘earn’ topic, the unlabeled examples play a positive role when more than 10% of training examples are labeled. So is for ‘acq’, when more than 7% of training examples are labeled.

However, even when we obtain the highest improvement by unlabeled examples, it does not reach to the best performance when we know the label of all the training examples beforehand. For example, the improvement is 5.81% when 10% of examples are labeled in ‘acq’. In this case, the accuracy is just 89.93% while the accuracy with 100% labeled examples is 95.21% (see Table 2). This implies that some of the unlabeled examples are mislabeled during the co-training process. The effectiveness of unlabeled examples can be maximized when the number of labeled examples is small. To fill a gap between the difference in accuracy, human intervention is needed. But, it is still a open problem when to intervene in the process.

4 Conclusions

We have presented an approach to text classification incorporating both the lexical and the syntactic information of documents. For this purpose, we adapted the co-training as an automatic classification learner, which is a partially supervised learning algorithm. Using the syntactic information improved the classification accuracy in the experiments on the Reuters-21578 corpus. While the effectiveness of unlabeled examples is experimentally proved, another problem is caused that we need a method to overcome the misguide of a partially supervised algorithm.

References

[1] A. Blum and T. Mitchell, “Combining Labeled and Unlabeled Data with Co-Training,” In *Proceedings of 11th Annual Conference of Computational Learning Theory*, pp. 92–100, 1998.

[2] CoNLL, *Shared Task for Computational Natural Language Learning* (CoNLL), http://lcg-www.uia.ac.be/conll2000/chunking, 2000.

[3] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features,” In *Proceedings of the European Conference on Machine Learning*, pp. 137–142, 1998.

[4] E. Stamatatos, N. Fakotatis and G. Kokkinakis, “Automatic Text Categorization in Terms of Genre and Author,” *Computational Linguistics*, 26(4), pp. 471–496, 2000.

[5] K. Tumer and J. Ghosh, “Error Correlation and Error Reduction in Ensemble Classifiers,” *Connection Science*, Vol. 8, No. 3, pp. 385–404, 1996.