Expert Systems with Applications 38 (2011) 11581–11590

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A logistic regression-based smoothing method for Chinese text categorization

Show-Jane Yen[a], Yue-Shi Lee[a,*], Jia-Ching Ying[b], Yu-Chieh Wu[c]

[a] Department of Computer Science and Information Engineering, Ming Chuan University 5, De-Ming Rd, Gweishan District, Taoyuan 333, Taiwan
[b] Department of Computer Science and Information Engineering, National Cheng-Kung University 1, University Road, Tainan City 701, Taiwan
[c] Department of Electronic Commerce, Kai-Nan University 1, Kainan Road, Luzhu Shiang, Taoyuan 33857, Taiwan

ARTICLE INFO

Keywords:
Text classification
N-gram-based classification
Feature selection
Word segmentation
Logistic regression

ABSTRACT

Automatic Chinese text classification is an important and a well-known technology in the field of machine learning. The first step for solving Chinese text categorization problems is to tokenize the Chinese words from a sequence of non-segmented sentences. However, previous literatures often employ a Chinese word tokenizer that was trained with different sources and then perform the conventional text classification approaches. However, these taggers are not perfect and often provide incorrect word boundary information. In this paper, we propose an *N*-gram-based language model which takes word relations into account for Chinese text categorization without Chinese word tokenizer. To prevent from out-of-vocabulary, we also propose a novel smoothing approach based on logistic regression to improve accuracy. The experimental result shows that our approach outperforms traditional methods at least 11% on micro-average F-measure.

1. Introduction

In recent years, with the rapid growth of the World Wide Web there are more and more digital text services on the internet. Examples include Google, Yahoo, etc. Search engine does not only provide user with the information, but also manage the information effectively. In order to make it easy, many search engine systems classify the texts in advance to organize their taxonomy. For example, Lam, Ruiz, and Srinivasan (1999) discuss that “One useful application for automatic categorization is to support effective text retrieval”. Owing to the huge amount of data, it is quite difficult to classify all texts by artificial selection and the contrived arbitrariness. Automatic text classification is a well-studied technique in machine learning and data mining domains, and there are many applications about it, like web classification, information retrieval, information filtering, etc. For example, Jiang (2006) indicate that “In regard to spam filtering, it can be use to classify incoming messages into either legitimate or spam category”.

The methods of automatic text classification help people to classify by machine learning technique. There are two typical types, supervised learning and unsupervised learning. The unsupervised learning is a kind of two step training. First, it clusters all texts according to the feature of texts, and then assigns each cluster to different class for building the classification model. On the contrary, the supervised learning trains the model using the data which is assigned to a class beforehand. The difference relies in the former does not need labeled training data. We follow the supervised learning prototype. Since it is known that the unsupervised learning still far away from state-of-the-art.

Over the past decade, many supervised machine learning techniques have been applied to text categorization problems, such as Naive Bayes classifiers (Sebastian, 2002; Yen, Lee, Lin, & Ying, 2006), support vector machines (Sebastian, 2002; Yen et al., 2006), linear least squares models, neural networks (Sebastian, 2002; Yen et al., 2006), and *k*-nearest neighbor classifiers (Sebastian, 2002; Yen et al., 2006). Yang and Liu experiment on the news text of Reuter (Yang, 1999; Yang & Liu, 1999). They reported that support vector machine and *k*-nearest neighbor classifier achieved the best accuracy in comparison to the other four methods. In 2002, Sebastian (2002) pointed out that SVM had better performance in general case.

However, support vector machine classifiers still have many problems. First, it is designed to solve the binary-class classification problem. Therefore it should be converted to handle multi-class classification problem. Second, the support vector machine classifiers generally could not generate probability. Most support vector machine classifiers represent the similarity between the class and the text using the cosine similarity. Unlike probability, the cosine similarity is very abstractive to represent the similarity between the class and the text. These problems are solved by Tipping (2001). He combines the logistic regression in the support vector machine classifiers to generate the probability. Meanwhile, the multi-class classification problem also can be transformed into

* Corresponding author. Tel.: +886 3 3507001; fax: +886 3 3593874.
E-mail addresses: sjyen@mail.mcu.edu.tw (S.-J. Yen), leeys@mail.mcu.edu.tw (Y.-S. Lee), jashying@idb.csie.ncku.edu.tw (J.-C. Ying), bcbb@db.csie.ncu.edu.tw (Y.-C. Wu).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.036

several binary-class classification problems. Nevertheless, a known problem with the Tipping's approaches is their relative inability to scale with large problems like text classification. Fortunately, Silva and Ribeiro (2006) solved the problem by the method of dimension reduction.

Nonetheless, these "standard" approaches to Chinese text categorization has so far been using a document representation in a word-based 'input space' (He, Tan, & Tan, 2003), i.e. as a vector in some high (or trimmed) dimensional Euclidean space where each dimension corresponds to a word. The advantages of this method are effective and high feasibility, but it assumes each word is independent with each other. Hence we need to solve word segmentation first for Chinese text. Most text classifiers treat each feature are mutually independent, word segmentation in Chinese is a difficult problem and we could not ensure that every words which is segmented by some word segmentation process are mutually independent. To solve it, character level N-gram models (Cavnar & Trenkle, 1994; Damashek, 1995; Peng & Schuurmans, 2003; Peng, Huang, Schuurmans, & Wang, 2003; Teahan & Harper, 2001) could be applied. These approaches generally outperform traditional word-based methods in term of accuracy. Moreover, these approaches can avoid the word segmentation step models the word-by-word relations (Dumais, Platt, Heckerman, & Sahami, 1998; Peng, Huang, Schuurmans, & Cercone, 2002).

By following this line, in this paper, we present a novel N-gram-based smoothing estimator using the logistic regression for Chinese text categorization. One main feature is that the method captures the dependence relations between Chinese characters from the given training data and can be generalized to handle unknown words in the testing text. By means of the proposed logistic regression-based smoothing algorithm, it shows even better empirical results in accuracy than the other smoothed method. Our method is also able to be combined with the conventional feature selection criterion, such as chi-square and information gains.

The remainder of the paper is organized as follows: Section 2, the concept of the related works is introduced, and the proposed text categorization method is proposed in Section 3. Section 4 presents the experiments of our approach, and in Section 5, we draw the conclusion and future remarks.

2. Related works

Text classification is a well-studied technique in data mining and machine learning technique over the past decades. There are, however, many novel text categorization algorithms had been proposed in resent years. As we know, text classification models can be grouped into many kinds of models. The most famous two types are vector space models and probabilistic models. Generally speaking, one should select features first from the training data before model building. In this section, we briefly reviews the literatures that are most related to our method.

2.1. Relevance vector machines

RVM, pioneered by Tipping (2001) is a sparse learning algorithm, similar to the SVM in many respects but capable of delivering a fully probabilistic output. It was reported to have nearly identical performance to, that of SVM in several benchmarks (Tipping, 2001). This can lead to significant reduction in the computational complexity of the decision function, thereby making it more suitable for real-time applications (Silva & Ribeiro, 2006). The RVM was proposed by Tipping (2001), as a Bayesian treatment of the sparse learning problem. The RVM preserves the generalization and sparsity of the SVM, yet it also yields a probabilistic output, as well as circumventing other limitations of SVM, such as the need for Mercer kernels (Joachims, 1998) and the definition of the classification error/margin size trade-off parameter C. This generally entails a cross-validation procedure, which is wasteful both of data and computation.

For an input vector x, an RVM classifier models the probability distribution of its class label $d \in \{+1, -1\}$ using logistic regression as

$$p(d = 1|x) = \frac{1}{1 + e^{-f_{RVM}(x)}}, \tag{2.1}$$

where $f_{RVM}(x)$, the classifier function, is given by

$$f_{RVM}(x) = \sum_{i=1}^{N} \alpha_i K(x, x_i), \tag{2.2}$$

where $K(\cdot,\cdot)$ is a kernel function, and x_i, $i = 1, 2, \ldots, N$, are training samples. Unfortunately, RVM obviously does not scale well with the number of training example. For example, when the training instances scales twice, it appends $6 \sim 12$ times larger training time (Tipping, 2001).

2.2. Näive Bayes

In this model, a document d is normally represented by a vector of K attributes $d = (v_1, v_2, \ldots, v_K)$. The naive Bayes model assumes that all of the attribute values v_1', are independent given the category label c. Thus, a maximum a posteriori (MAP) classifier can be constructed as follows.

$$\arg\max_{c \in C} \left\{ P(c) \times \prod_{j=1}^{K} P(v_j|c) \right\}. \tag{2.3}$$

In the following, we take an example to describe the naive Bayes model. Given a document d = "*I want to eat Chinese food.*" which we wish to classify, and let the frequency of words of the training data for each class given in Table 2.1 and the number of words in each class given in Table 2.2.

In this example, we assume $P(C_1) = 0.41$, $P(C_2) = 0.47$ and $P(C_3) = 0.12$. Then we calculate the probability of the document d is classified into some class.

$$C_1 : 0.41 \times \left(\frac{3437}{1928374} \times \frac{1215}{1928374} \times \frac{3256}{1928374} \times \frac{938}{1928374} \times \frac{213}{1928374} \times \frac{1056}{1928374} \right) = 2.29 \times 10^{-20},$$

$$C_2 : 0.47 \times \left(\frac{3955}{1029385} \times \frac{1777}{1029385} \times \frac{3256}{1029385} \times \frac{918}{1029385} \times \frac{228}{1029385} \times \frac{1772}{1029385} \right) = 3.35 \times 10^{-18},$$

Table 2.1
The frequency of words of the training data.

	I	want	to	go	out	eat	Chinese	food	...
C_1	3437	1215	3256	357	275	938	213	1506	...
C_2	3955	1777	3256	358	274	918	228	1772	…..
C_3	895	1955	3256	359	273	128	213	1789	….

Table 2.2
The number of words.

C_1	C_2	C_3
1928374	1029385	1564738