IMPROVED FEATURE SELECTION APPROACH TFIDF IN TEXT MINING

LI-PING JING, HOU-KUAN HUANG, HONG-BO SHI

School of Computer & Information TechnologyNorthern JiaoTong University Beijing, 100044
E-MAIL: jlpapplelqy@yahoo.com.cn, jlpapple@sina.com

Abstract:

This paper describes one Feature Selection method (TFIDF). With it, we process the data resource and set up the VSM model in order to provide a convenient data structure for text categorization. We calculate the precision of this method with the help of categorization results. According to the empirical results, we analyze its advantages and disadvantages and present a new TFIDF-based feature selection approach to improve its accuracy.

Keywords:

Text mining; TFIDF; Evaluation function; VSM; Feature selection

1 Introduction

Text mining, also known as text data mining, is a new field to which more and more researchers pay attention. Text mining essentially differs from the traditional data mining. The data processed by traditional data mining is structured, because there is special model and traditional data mining can use this model to describe the special data concretely. However, text data is non-structured or semi-structured, in order to convert these data into feature vector (one structured form), we need at least teens or hundreds of thousands of features for even a moderate-sized text collection. So the first problem to be addressed is how to express these text features conveniently and reasonably, using the least features denotes the most information, which is also our major research goal---text feature subset selection and description.

2 Data Preprocessing

The representation of a problem has a strong impact on the generalization accuracy of a learning system. For the problem of text categorization of a document, which is typically a string of characters, has to be transformed into a representation, which is suitable for the learning algorithm and the classification task.

In view of text classification, we use feature-vector document representation, that is (i.s.), take one document as a set of Term sequences, including term t and term weight w. Then the document will be made up of the pairs of <t, w>. Term T_1, T_2, T_3......T_n represent the features which express the document content. We could treat them as a N-dimension coordinate. Weights W_1, W_2, W_3......W_n represent the value relevant to coordinate. So every document (D) is mapped to the target space as a feature-vector V(d) = (T_1, W_1, T_2, W_2, T_3, W_3......T_n, W_n).

The main purpose of data preprocessing is to deal with the data resource and build up the feature-vectors. We use term weight as the criterion of feature selection. At present, many researchers have studied it and reached some achievements, especially, the research of Yang Yiming of Carnegie Mellon University (CMU) and Mechran Sahami of Stanford are authoritative. There are some common evaluation functions for preprocessing, such as document frequency, mutual information, information gain, expected cross entropy, weight of evidence for text[1] and etc. In experiment, we test the text collection on the base of evaluation function, document frequency. The following briefly introduces this function and its improvement.

The values of the vector elements W_i for a document d are calculated as a combination of the statistics TF(t,d) and DF(t). The term frequency TF(t,d) is the number of times word t occurs in document d. The document frequency DF(t) is the number of documents in which the word t occurs at least once. The inverse document frequency IDF(t) can be calculated from the document frequency.

$$\mathrm{IDF}(t) = \log\left(\frac{|D|}{DF(t)}\right) \qquad (1)$$

|D| is the total number of documents. The inverse document frequency of a word is low if it occurs in many documents and is highest if the word occurs in only one. The value W_i of features T_i for document d is then calculated as the product

$$W_i = \mathrm{TF}(t_i, d) \bullet \mathrm{IDF}(t_i) \qquad (2)$$

W_i is called the weight of word T_i in document d. This word weighting heuristic says that a word T_i is an important indexing term for document d if it occurs frequently in it(the term frequency is high). On the other hand, words which occur in many documents are rated less important indexing terms due to their low inverse document frequency. From the above, we could find that IDF servers as an adjusting function to modulate the term frequency.

3 Feature selection and set up VSM

0-7803-7508-4/02/$17.00 ©2002 IEEE

Before applying feature selection to documents, we use the following two steps on the data source: pruning of infrequent words and pruning of high frequency words. The former means that words are only considered as features, if they occur at least once in the training data. In the experiments described in this paper words had to occur at least 3 times. This removes most spelling errors and speeds up the following 2 stages of features selection. The later is a technique which is supposed to eliminate non-content words like "the", "and", or "for". Additionally, the names of place and people are represented by two fixed-words (Word_Place_Name, Word_Person_Name). Then we could reduce some unnecessary repetitive operations.

Besides these, words have many morphological variants that will not be recognized by term-matching algorithms without additional text processing. In most cases, these variants have similar semantic interpretations and can be treated as equivalents for information retrieval (as opposed to linguistic) applications. For example, act, action, acted, acting, actable are all derived from the word: act, that is, they have the same stem. Therefore, classic stemming algorithms[2] have been created for this system which reduce these variants to a root form. In some cases, stemming-processing significantly improve the categorization efficiency.

After primeval filtering, the features in training set will be selected by the method TFIDF (term frequency / inverse document frequency) mentioned above. Selection of a subset of features used in inductive learning has already been addressed in pattern recognition and machine learning. According to search decision there is forward selection and backward elimination; according to feature selection algorithm, there is filtering approach and wrapper approach[3]. However, most of the feature subset selection methods used in machine learning are not designed for the situations with a large number of features, in other words, they are not suitable for text data mining, because of a great number of features in text. For example, given N features, there will be a feature space whose size is 2^N in optimization search, which can easily result in the time complexity we cannot tolerate. This is an attribute-value representation where documents are represented by the words occurring in it without attention to their ordering.

To select a subset of n features, the n words with the highest TFIDF are chosen. Using TFIDF method scores individual feature T_i and computes its evaluation weight value. With ordering these scores, we set up a threshold for getting expected information from a document and guarantee the percentage of information. Figure1 shows the features and their weights.

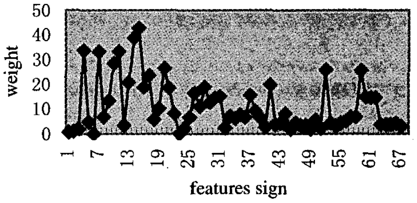

Fig.1 weight versus the selected features in one document

Here, we experiment on the data resource using the different information percentages: 70%， 75%， 80%， 85%， 90%， 95%. Finally, the size of vocabulary resulted from the training set is the following as Table 1:

Table 1. comparison of vocabulary

Percent	Vocabulary
70%	2014
75%	2158
80%	2216
85%	3789
90%	5469
95%	11046

Table 1 shows that the size of vocabulary almost keeps invariable with the change of percentage from 70% to 85%. Thinking of the quantity of information, we adopt 80% as the threshold.

Once have such a vocabulary, we could conveniently convert every document into another representation which can be easily processed by computer. Vector space model (VSM) is one of the better methods[4] used frequently these years, VSM is made up of such vector as V(d) = （T_1,W_1， T_2,W_2， T_3,W_3......T_n,W_n）. At last, the feature subset of every document contains the largest number of information, because this subset is processed by primeval filtering and evaluated by TFIDF.

4 Experimental results and analysis

In experiment, we use the standard text data resource issued by CMU, including train set and test set[5]. All the documents are included in the assigned 135 topic categories. Table 2 shows the data distribution:

Table 2. data distribution

Item	trainset	testset
Number of docs	9603	3299
Average docs per class	71	24
Largest docs in class	4053	1534
Fewest docs in class	0	0

To test set, every document is represented by two kinds of vectors.

$$V(d) = (T_1,B_1,\ T_2,B_2,\ T_3,B_3......T_n,B_n)$$

where T_i represents one feature of the vocabulary, B_i shows whether or not feature T_i occurs in this document d, if yes, set B_i 1, or 0;

$$V(d) = (T_1,N_1,\ T_2,N_2,\ T_3,N_3......T_n,N_n)$$

where T_i represents one feature of the vocabulary, N_i shows the number of feature T_i which occurs in this document d, here, n=2116 represents the size of vocabulary.

On the base of VSM, we classify data resource with Naive Bayse method. Table 3 shows experimental results on the test set.

Table 3 Classification results

	Alum	Coffee	Gas	Oilseed	...	Retail
Alum	50	0	0	0	...	0
Coffee	0	82	0	0	...	0
Gas	0	10	0	0	...	0
Oilseed	0	0	0	93	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Retail	0	0	0	0	...	150

Experimental results suggest that with TFIDF method, the classification accuracy is up to 76%. Compared with the method using IDF as evaluation function, it is better because the accuracy of the later is just 61%, however, this cannot reach the expected result. Taking this point into account, we make the following conclusions:

In order to extrude the important words out and restrain the subordination words, TFIDF approach uses IDF function to adjust the feature weight. Whereas IDF hardly perform complete adjusting function for its simple structure, so this method cannot work well in practice. When the number of documents which belong to one class is a bit small, classification results will make mistake, which has something to do with the document frequency formula[6]: IDF function. It is distinctly unreasonable that calculating the feature weight is on the base of the number of all documents in train set. For instance, feature T in the class Gas occurs frequently in this class but the number of documents including the feature T is very small, that is, high term frequency. However, TFIDF depends on the document frequency formula rather than taking both of the results into account.

For this reason, the process adjusting the feature weight should be changed. Instead of IDF, we use mutual information (MI) evaluation function to remedy this weakness, i.s. make term frequency formula[6] as adjusting function.

$$\text{MutualInfoTxt(F)} = \Sigma\ P(C_i) * (\log (P(W|\ C_i)/ P(W))) \qquad (3)$$

This formula shows how to calculate the frequency of one word involved in all class sets with basic information theory[7]. MI measures the reduction in entropy that is achieved, if one random variable is conditioned on another one. However, we find that MI function can't take the term frequency into account. So in order to remedy the shortage as well, we decide to adjust TF(t) with MutualInfoTxt(F), just as follows:

$$\text{WeightofTxt(F)} = \text{TF(t)} * \text{MutualInfoTxt(F)} \qquad (4)$$

where TF(t) represents feature frequency in one document d, i.s. term frequency, the value is positive proportion to the number of term times and inverse proportion to the total term times in document d. And then this improved formula considers not only the characteristics of term frequency but also the characteristics of document frequency.

With the improved feature subset selection method, we preprocess the given document sets and set up new VSM. To our surprise, the final classification precision is up to 88% with the same classifying method: Naive Bayse classifier, which is very exciting.

5 Conclusions

More and more researchers gradually pay attention to data preprocessing of text mining. The core of preprocessing is text feature subset selection, i.s. to reduce the number of terms without damnifying any classification precision and to improve the classification effect and rate. Data preprocessing will be key technology to which every field has to refer. In the future work, we will make our text mining system more perfect, especially, with improving the feature selection approach, so that we finally improve the classification precision and efficiency[8].

References

[1] Yang Y, Pedersen JO. A Comparative Study on Feature Selection in Text Categorization

[2] http://www.cs.cmu.edu/~mccallum/bow/rainbow// the term frequency algorithm

[3] John G H, KohaviR, Pfleger K. Irrelevantfeatures and the subset selection problem, 1998

[4] Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM, 1975.18(5): 613~620

[5] http://www.cs.cmu.edu/~mccallum/bow/rainbow

[6] Joachims T. A Probabilistic Analysis of theRocchio Algorithm with TF-IDF for TextCategorization. Proc of the 14th InternationalConference on Machine Learning ICML97, 1997. 143~151

[7] Dunja Mladenic. Machine Learning on non-homogeneous distributed text data. Ljubljana, 1998

[8] Yang Y. An evaluation of statistical approach to text categorization. In Technical Report CMU-CS-97-127, CS Dept. CMU, 1997