Text Categorization Based on a Similarity Approach

Cha Yang Jun Wen

School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China

Abstract

Text classification can efficiently enhance the text processing capability by automatically sorting out them according to defined collection of categories. This paper uses TFIDF method to represent documents, and set the NGramSize value to be 6. Word Frequency vector is used to measure and distinguish different features on documents. The Similarity Approach uses Cosine function to construct the classifier. The experiment results indicate that proposed algorithm yields good performance with the accuracy up to 98%.

Keywords: Text classification, TFIDF, Vector space model, Word frequency, Similarity

1. Introduction

The fast growth and dynamic change of online information have provided us a very large amount of information and lead to information overload. Text (Document) categorization (TC) is an important tool for organizing documents into categorizations by applying statistical methods or artificial intelligence techniques. By this way, the utilization of the documents can be expected to be more effective. As a result, the situation of information overload may be alleviated. The aim of document categorization is to assign a number of appropriate categories to a textual document based on the content. This categorization process has many applications such as document routing, dissemination, or filtering. A large number of techniques have been developed for text classification, including Naive Bayes, Nearest Neighbor, neural networks, regression, rule induction, and Support Vector Machines [1].

Formally, TC consists of determining whether a document di (from a set of documents D) belongs or not to a category Cj (from a set of categories C), consistently with the knowledge of the correct categories for a set of training documents. To perform this task, two main processes need to be carried out. First, the documents must be transformed into a form suitable for automatic processing. This chore includes the removal of tags, the elimination of non-informative words. Then, each importance stem in the document or relevant words are selected and used to represent the documents. This reduction of the vocabulary considered to represent the documents is a crucial step in the process, since it has been noted that it can greatly improve the overall performance. Second, once the representation of the documents is fixed the assignment of the documents to the categories is the next step. Usually, an automatic classifier is induced from a set of correctly labeled examples using statistical methods or machine learning algorithms. Then this classifier is used to assign each new document to one or more categories.

The goal of a text classification system is to determine whether a given document belongs to any of the predefined categories. Since the document can belong to zero, one, or more categories, the system can be a collection of binary classifiers, in which one classifier classifies for one category. These are universal binary classifiers able to find linear or non-linear threshold functions to separate the examples of a certain category from the rest, which are based on the Structural Minimization Risk principle from computational learning theory.

This paper uses the popular feature selection Word Frequency and the effective classifier model for TC. The model evaluation using classification accuracy method indicate that the Similarity Approach take good efficiency on TC. The following chapters below: Chapter 2 introduces the TC's history from '80s to now and how this paper to deal with TC more efficiently. Chapter 3 gives the key technologies—TFIDF, Word Frequency and Similarity Approach in the TC process. In Chapter 4 the experiments was performed, then the results was analyzed. Chapter 5 offered the conclusion of this paper and the future work to improve the TC method.

2. Related works

In the '80s, in order to create the automatic document classifiers in their manual construction, knowledge engineering (KE) techniques are used. Example to build manually, an expert system required set of manually defined rules under the following type:

if (DNF' Boolean formula) then (category) else not(category)

It means that the document was classified under (category) if is satisfied (DNF Boolean formula). And the construe system was built by Carnegie Group for the Reuters news agency, is the typical example for this approach. Since the early '90s, the more effective and powerful approach which has been built and replaced for the KE approach, was machine learning (ML).By extracting the characteristics of a set of documents which have been pre-classified manually under ci by a domain expert, a general inductive process (also called the learner) automatically builds a classifier for a category ci. The advantages of this approach are that construction of a classifier based on an automatic builder of classifier from a set of manually classified documents (learning), not of a classifier. Recent research on machine learning and data mining has provided developed methods and algorithms to construct statistical models of network data including social networks, web-page networks, email tracks, citation networks, and so on. The models can be constructed either directly from data using information extraction algorithms, which are applied mostly on relational database, and semi-structured text, or indirectly from unstructured textual data using text mining techniques.

The VSM (Vector Space Model) is a regular model to represent text documents. It is a method of bag of words, which is used widely in IR, TC, TM and WM. And the TFIDF algorithm is often combined with VSM in TC applications. This representation ignores the sequence in which the words occur and is based on the statistic about single words in isolation. There are many text representations that aim to stand for the features of documents in different domains, such as n-grams represented that employs word sequences of length-grams up to n, document concept categories and so on.

This paper is to preprocess the documents by Feature Extraction and Feature Selection. The methods TFIDF and Word Frequency were used. And the 200 documents were used for the experiment. The process of TC consists of training and testing.

3. Classification methods

3.1. TFIDF

We use feature-vector to represent documents, that is, 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*>. t_1, t_2, $t_3 \ldots t_n$ represent the features which express the document content. We could treat them as an N-dimension coordinate. w_1, w_2, $w_3 \ldots w_n$ represent the value relevant to coordinate. So every document (*d*) is mapped to the target space as a feature-vector $V(d) = (t_1, w_1, t_2, w_2, t_3, w_{3\ldots\ldots}t_n, w_n)$ [7].

The main purpose of data preprocessing is to deal with the data resource and build up the feature-vectors. We use weight as the criterion of feature selection. 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.

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

$|D|$ 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

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

w_i is called the weight of word t_i in document *d*. This word weighting heuristically says that a word t_i is an important indexing term for document *d* if it occurs in 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. We could find above that IDF servers as an adjusting function to modulate the term frequency.

3.2. Feature selection

Feature selection studies how to select a subset or list of attributes or variables that are used to construct models describing data. Its purposes include reducing dimensionality, removing irrelevant and redundant features, reducing the amount of data needed for learning, improving algorithms' predictive accuracy, and increasing the constructed models' comprehensibility.

Dimension reduction techniques can generally be classified into Feature Extraction (FE) approaches and Feature Selection (FS). FS algorithms select a subset of the most representative features from the original feature space. FE algorithms transform the original feature space to a smaller feature space to reduce the dimension. Though the FE algorithms have been proved to be effective for dimension of

data sets the text domain reduction, high dimension often fails in many FE algorithms due to their high computational cost. Thus FS algorithms are more popular for practical text data dimension reduction problems.

The performance of a feature subset should be evaluated based on a certain ground, which is achieved by Evaluators. Evaluation function is to measure and distinguish the classification capabilities of different features on documents. In fact various evaluation functions are already applied, such as: MI (Mutual Information), IG (Information Gain), ECE (Expected Cross Entropy), OR (Odds Ratio), WET (the Weight of Evidence for Text), WF (Word Frequency) [2].

The evaluation function of Word Frequency is:

$$Freg(F) = TF(W) \qquad (3)$$

In this paper, we implemented Word Frequency method which is simple and efficient for Chinese document classification.

3.3. Text categorization via similarity classifier

There are two major purposes of text similarity measure: the first one is to find out all of the similar (or related) documents from a large document collection, such as IR (Information Retrieval), TM (Text Mining), WM (Web Mining) and TC (Text Classification/Clustering); the other is to find out the copies of a document from the collection, such as (Document Copy Detection) DCD. That causes different requests for similarity measure. The former wants to return those related documents that are far away from other categories. The latter one needs to distinguish the almost same documents (copies) from other similar documents [11].

Cosine function, dot product and proportion function are commonly used similarity measures. Usually, we define the similarity value in [0, 1] so that cosine and proportion functions are widely used. Let F (A) and F (B) be document A and B word frequency vectors, then the similarity of A and B in cosine function is Scos (A, B):

$$S_{\cos}(A,B) = \frac{\sum_{i=1}^{n} \alpha_i^2 \times F_i(A) \times F_i(B)}{\sqrt{\sum_{i=1}^{n} \alpha_i^2 \times F_i^2(A) \times \sum_{i=1}^{n} \alpha_i^2 \times F_i^2(B)}} \qquad (4)$$

where αi is the word weight vector, Fi (A), Fi (B) are the respective number of occurrences of the ith word in A and B. Obviously Scos (A, B) = Scos (B, A) called symmetric similarity. The similarity of A and B in proportion function is S% (A, B):

$$S_{\%}(A,B) = \frac{|F(A) \cap F(B)|}{|F(A) \cup F(B)|} = \frac{\sum_{i,j=1}^{n} \alpha_i (F_i(A) \oplus F_j(B))}{\sum_{i=1}^{n} \alpha_i F_i(A) + \sum_{j=1}^{n} \alpha_j F_j(B)} \qquad (5)$$

where $F_i(A) \oplus F_j(B)$ means that:

$$F_i(A) \oplus F_j(B) = \begin{cases} F_i(A) + F_j(B) & w_i = w_j \\ 0 & w_i \neq w_j \end{cases}$$

$$i, j = 1,2,3,...,n \qquad (6)$$

For similarity, the copies' (same documents) value is 1 and the more overlapped words between documents the higher score. But they cannot distinguish the subset copies from partly overlapped documents. As we know that A is included in B is different from B is included in A, i.e. $A \subset B \neq B \supset A$. So the measurement of $A \subset B$ should be different from that of $B \supset A$. However the similarity does not satisfy that.

In order to find out subset document copy, Shivakumar and Garcia-Molina (1995) proposed RFM (Relative Frequency Model). The subset measure of document A to be a subset of document B to be:

$$Subset(A,B) = \frac{\sum_{w_i \in c(A,B)} \alpha_i^2 \times F_i(A) \times F_i(B)}{\sum_{i=1}^{n} \alpha_i^2 F_i^2(A)} \qquad (7)$$

It is obvious that Subset (A, B) ≠ Subset (B, A) and Subset (A, Ac) = 1 if Ac is a copy of A. Hence we call this type as similarity measure. The final RFM similarity measure between two documents A and B is:

$$S_{RFM}(A,B) = \max\{Subset(A,B), Subset(B,A)\} \qquad (8)$$

The Subset (A, B) may be greater than 1. In order to regularize the similarity value in [0, 1], the final RFM similarity of documents A and B is:

$$S_{RFM}(A,B) = \min\{1, \max\{Subset(A,B), Subset(B,A)\}\} \qquad (9)$$

The RFM is derived from cosine function. Similar to that, we define another asymmetric similarity that derived from proportion function. We

call it IPM (Inclusion Proportion Model). The inclusion proportion of $A \subset B$ is:

$$Incl(A,B) = \frac{|F(A) \cap F(B)|}{|F(A)|} = \frac{\sum_{i,j=1}^{n} \alpha_i (F_i(A) \oplus F_j(B))}{2 \times \sum_{i=1}^{n} \alpha_i F_i(A)} \quad (10)$$

Incl (A, B) ≠ Incl (B, A) and Zncl (A, Ac) = 1 if Ac is a copy of A. The same as RFM, the final IPM similarity of documents A and B is:

$$S_{IPM}(A,B) = \min\{1, \max\{Incl(A,B), Incl(B,A)\}\} \quad (11)$$

From experiments we believe the RFM and IPM are both excellent for subset copy detection.

4. Experiments

Figure 1 is a basic Text Categorization model. TC is composed of two parts—training and testing. In the training process, the first thing is to build the feature of the training text sets and then get the feature sets. The feature extraction algorithm is TF-IDF and the feature selection algorithm is Word Frequency. During the classification, we use the similarity approach to construct the classifier to get the category of the text sets for testing.

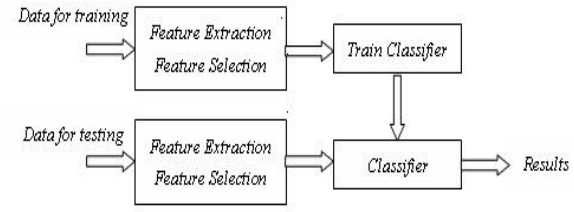

Fig. 1: Text Categorization Model.

4.1. Data set

For this experiment we used 200 documents downloaded from sina.com. 120 documents of them are for training and others for testing. These categories are: society, economy, music, and computer.

4.2. Performance measurements

Classification performance is measured using both recall and precision. In this case, recall is the proportion of the correct documents that are assigned to a category by the algorithm. Precision is the proportion of documents assigned to a category that belong to that category. Text categorization is essentially a series of dichotomous results and so both micro and macro averaging can be used to generate an overall performance over the set of categories used.

The classification model's evaluation functions are:

Classification Accuracy:

$$\text{Accuracy(M)} = \sum_{ex} \text{P(ex)Accuracy(M, ex)}$$

$$= \text{P}(\hat{\text{C}}(\text{ex}) = \text{C}(\text{ex}))$$

$$\text{Accuracy(M, ex)} = \begin{cases} 1; & \hat{\text{C}}(\text{ex}) = \text{C}(\text{ex}) \\ 0; & \text{other} \end{cases} \quad (12)$$

Precision:

$$\text{Precision(M, targetC)} = \text{P(targetC} \mid \hat{\text{targetC}}) \quad (13)$$

Recall:

$$\text{Recall(M, targetC)} = \text{P}(\hat{\text{targetC}} \mid \text{targetC}) \quad (14)$$

4.3. Results

The performance presented in this subsection is evaluated from the accuracy of an approach to predict the category of test documents. The prediction accuracy is defined using the percentage of test documents that arc correctly categorized. For a document in the data set that was assigned to several categories, the prediction is considered to be correct if it was one of the given categories.

In addition to the alternatives or test document representation at described above, we also performed experiments where normalized term frequency (TF) and TF-IDF weighting schemes were used to estimate the membership degree of words occurring in documents. TF-IDF is a well-studied weighting scheme from information retrieval that assigns the weight of a term proportional to the occurrence Frequency of the term in each document and inversely proportional to the total number of documents to which the term occurs in a given document collection. However, the performance obtained by employing these two weighting schemes is even worse. Fine-tuning the membership degrees

of words in documents degrades the performance of the similarity approach.

120 documents were trained to construct the classifier. And then 80 documents got the best category of Ci were tested. Every testing document had its result with the rank of every category by the classification model's evaluation function. The smaller the rank value is, the better the testing text gets the category. And since we know what the text came from, we can evaluate how well the classifier is doing with this method.

5. Conclusions

This paper uses a similarity approach for text categorization. With TFIDF and Word Frequency, the experimental results indicate that proposed algorithm yields good performance on TC.

There are some works to do to optimize the text categorization method in the future. First, we will try to find the best feature extraction and selection method handling polysemy and synonymy for the Chinese text classification. Second, an improved similarity approach is required to take much efficiency of TC.

Acknowledgement

The work is partially funded by Young Fund of Electronic Science & Technology of China.

References

[1] J.W. Han, Micheline Kamber, *Data Mining Concepts and Techniques, Second Editon*, China Machine Press, 2007.

[2] Z.Z. Shi, *Knowledge Discovery*, Tsinghua University Press, 2005.

[3] G. Forman, an Extensive Empirical Study of Feature Selection Metrics for Text Classification. 2003.

[4] F. Sebastiani, Machine learning in automated text categorization. *ACM Computing Surveys*, 34, 2002.

[5] V. Vapnik, *the Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.

[6] X.H. Wang, Retrieval-based Chinese Text Mining Technology Study and Design. Nov. 2004.

[7] X. Luo, D.L. Xia, P. Yan, Improved feature selection method and TF-IDF formula based on word frequency differentia. *Computer Applications*, 25(9), 2005.

[8] T. Joachims, a Probalilistic Analysis of the Tocchio Algorithm with TF-IDF for Text Categorization. *Prof. of the 14th International Conference on Machine Learning,* ICML97, 1997.

[9] Y. Yang, Pedersen Jo, a Comparative Study on Feature Selection in Text Categorization.

[10] X.Y. Chen, Yi Chen, L. Wang and Y.F. Wang, Text Categorization Based on Frequent Patterns With Term Frequency. *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, Shanghai, 2004.

[11] G. Yu, Y.J. Pei, Z.Y. Zhu and H.Y. Chen, Research of text similarity based on word similarity computing. *Computer Engineering and Design*, 27 (2), 2006.

[12] Y.T. Zhang, L. Gong, Y.C. Wang, an Improved TF-IDF Approach for Text Classification. 2005.

[13] Y. Jiang, Z.H. Zhou, a Text Classification Method Based on Term Frequency Classifier Ensemble. 2006

[14] X.Y. Chen, the Key Techniques Research on Text Mining. 2005.

[15] JanBakus, Mohamed S. Kamel, Higher order feature selection for text classification, 2006.