A New Inductive Learning Method for Multilabel Text Categorization

Yu-Chuan Chang[1], Shyi-Ming Chen[2], and Churn-Jung Liau[3]

[1] Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C.
D9315003@mail.ntust.edu.tw
[2] Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, Taiwan, R.O.C.
smchen@et.ntust.edu.tw
[3] Institute of Information Science, Academia Sinica
Taipei, Taiwan, R.O.C.
liaucj@iis.sinica.edu.tw

Abstract. In this paper, we present a new inductive learning method for multilabel text categorization. The proposed method uses a mutual information measure to select terms and constructs document descriptor vectors for each category based on these terms. These document descriptor vectors form a document descriptor matrix. It also uses the document descriptor vectors to construct a document-similarity matrix based on the "cosine similarity measure". It then constructs a term-document relevance matrix by applying the inner product of the document descriptor matrix to the document similarity matrix. The proposed method infers the degree of relevance of the selected terms to construct the category descriptor vector of each category. Then, the relevance score between each category and a testing document is calculated by applying the inner product of its category descriptor vector to the document descriptor vector of the testing document. The maximum relevance score L is then chosen. If the relevance score between a category and the testing document divided by L is not less than a predefined threshold value λ between zero and one, then the document is classified into that category. We also compare the classification accuracy of the proposed method with that of the existing learning methods (i.e., Find Similar, Naïve Bayes, Bayes Nets and Decision Trees) in terms of the break-even point of micro-averaging for categorizing the "Reuters-21578 Aptè split" data set. The proposed method gets a higher average accuracy than the existing methods.

1 Introduction

As the amount of information in the Internet is growing so rapidly, it is difficult for users to find desired information unless it is organized and managed well. Text categorization (TC) is a major research topic in machine learning (ML) and information retrieval (IR) to help users obtain desired information. A document can belong to a

M. Ali and R. Dapoigny (Eds.): IEA/AIE 2006, LNAI 4031, pp. 1249 – 1258, 2006.

single category, multiple categories, or not belong to any category. The goal of automatic text categorization is to utilize categorized training documents to construct text classifiers, which are then used to classify new documents into appropriate categories automatically. Several machine learning methods have been developed to deal with the text categorization problem, e.g., regression models [10], [17], nearest neighbor classification [16], [18], Bayesian probabilistic approaches [1], [12], [14], decision trees [1], [8], [12], inductive rule learning [1], [6], neural networks [18] and on-line learning [6].

In this paper, we present a new inductive learning method for multilabel text categorization. It uses the mutual information measure [11] for term selection and constructs document descriptor vectors for each category based on the selected terms. These document descriptor vectors form a document descriptor matrix and they are also used to construct a document-similarity matrix based on the cosine similarity measure [2]. It then constructs a term-document relevance matrix by applying the inner product of the document descriptor matrix to the document-similarity matrix. It infers the degree of relevance of the selected terms to construct the category descriptor vector of each category. The relevance score between each category and a testing document is calculated by applying the inner product of its category descriptor vector to the document descriptor vector of the testing document. The maximum relevance score L is then chosen. If the relevance score between a category and the testing document divided by L is not less than a predefined threshold value λ, where $\lambda \in [0, 1]$, then the document is classified into that category. We also compare the classification accuracy of the proposed method with that of the existing learning methods (i.e., Find Similar [13], Decision Trees [5], Naïve Bayes and Bayes Nets [14]) in terms of the break-even point of micro-averaging for categorizing the "Reuters-21578 Aptè split 10 categories" data set [21]. The experimental results show that the proposed method outperforms the existing methods.

2 Preliminaries

Machine learning systems deal with categorization problems by representing samples in terms of features in order to apply machine learning methods to text categorization, documents must be transformed into feature representations. The vector space model [13] is widely used in information retrieval for the representation of documents. In the vector space model, a document is represented as a document descriptor vector of terms and every element in the vector denotes the weight of a term with respect to the document. The learning methods presented in [10] and [17] calculate the tf $\times$ idf term weight for each term. The learning methods presented in [1], [8] and [12] use a binary weight 1 or 0 to represent each index term.

The dimension of a term's space is an important research topic in text categorization. With a high dimension, a classifier over fits training samples, which may be good for classification purposes, but it is not feasible for classifying previously unseen testing samples. The purpose of term selection is to choose relevant terms for document indexing that yield the highest accuracy rates. The simplest term selection method [19] is based on the frequency of a term's occurrence in documents, where only terms that occur in the highest number of documents are retained. Other term

selection methods are based on information-theoretic functions, such as the DIA association factor [9], information gain measure [6], [12], chi-square measure [4], and mutual information measure [8], [19]. In recent years, an increasing number of categorization methods have applied the mutual information (MI) measure in term selection [3], [7]. The mutual information score between term t_i and category c is defined by

$$MI(t_i, c) = p(t_i, c) \log_2 \frac{p(t_i, c)}{p(t_i)p(c)}, \tag{1}$$

where $p(t_i, c) = N_C(t_i)/N_C$, $p(t_i) = N(t_i)/N$, $p(c) = N_C/N$, $N_C(t_i)$ denotes the number of occurrences of term t_i in category c, N_C denotes the number of occurrences of all terms in category c, $N(t_i)$ denotes the number of occurrences of term t_i in the collection, and N denotes the number of occurrences of all terms in the collection.

There are some rules for determining the threshold value in multilabel text categorization [15], [16]. The threshold value is used when a document may belong to multiple categories. In this paper, we use a variant of the rank-based thresholding (R-cut) measure [16] to assign text documents into categories. We use the following criterion for multilabel text categorization:

$$\frac{Score(c_i, d)}{L} \geq \lambda, \tag{2}$$

where $Score(c_i, d)$ denotes the relevance score between category c_i and document d, $L = \max_j Score(c_j, d)$ denotes the maximum relevance score, λ is a threshold value that controls the degree of multilabel categorization, and $\lambda \in [0,1]$. If the relevance score between category c_j and document d divided by L is not less than the threshold value λ, where $\lambda \in [0,1]$, then the document d is classified into category c_i. The lower the threshold value λ, the more categories a document may belong to. If $\lambda = 0$, then the document belongs to all categories. Thus, the threshold value λ provides us some flexibility to deal with multilabel text categorization problem.

In the following, we briefly review some classifiers [8], namely, Find Similar [13], Decision Trees [5], Naïve Bayes and Bayes Nets [14].

(1) **Find Similar Classifier** [13]**:** The Find Similar method is a variant of Rocchio's method for relevance feedback, which is often used to expand queries based on the user's relevance feedback. In text classification, Rocchio's method calculates the weight w_t of a term t as follows:

$$w_t = \alpha \cdot w_t + \beta \cdot \frac{\sum_{i \in pos} w_{t,i}}{N_{pos}} + \gamma \cdot \frac{\sum_{i \in neg} w_{t,i}}{N_{neg}}, \tag{3}$$

where w_t denotes the weight of term t, N_{pos} denotes the number of positive documents in the category, N_{neg} denotes the number of negative documents in the category, and α, β and γ are the adjusting parameters. The method finds the representative centroid of the positive documents of each category and classifies a new document by comparing it with the centroid of each category by using a specific similarity measure. In [8],

Dumais *et al*. let $\alpha = 0$, $\beta = 1$ and $\gamma = 0$ and use the Jaccard similarity measure to calculate the degree of similarity between the centroid of each category and a testing document.

(2) **Decision Trees Classifier** [5]**:** A decision tree (DT) text classifier is a tree in which the internal nodes are labeled by terms, the branches are labeled by weights, and the leaves are labeled by categories. The classifier categorizes a testing document d_j by recursively test the weights of its terms (i.e., the internal nodes) until a leaf node is reached. The label of the leaf node is then assigned to d_j. The main advantage of the decision tree method is that it is easily interpretable by humans.

(3) **Naïve Bayes Classifier** [12]**:** The Naïve Bayes (NB) classifier (also called the Probabilistic Classifier) is a popular approach for handling classification problems. It uses the joint probability of terms and categories to calculate the probability that the terms of a document belong to certain categories, and then applies the Bayesian Theory to calculate the probability of the document d_j belonging to category c_i:

$$P(d_j \mid c_i) = \prod_{k=1}^{n} P(w_{kj} \mid c_i), \tag{4}$$

$$P(w_{kj} \mid c_i) = \frac{P(w_{kj}, c_i)}{P(c_i)}, \tag{5}$$

where $P(d_j \mid c_i)$ denotes the probability of document d_j belonging to category c_i; $P(w_{kj} \mid c_i)$ denotes the probability of term t_k of document d_j belonging to category c_i, and n denotes the number of terms belonging to document d_j and category c_i. The naïve part of the NB method is the assumption of term independence. This makes NB classifiers far more efficient than non-naïve Bayes methods due to the fact that there is no need to consider the conditional probabilities of terms.

(4) **Bayes Nets Classifier** [14]**:** In [14], Sahami utilizes the Bayesian network for classification, which relaxes the restrictive assumptions of the Naïve Bayes classifier. A 2-dependence Bayesian classifier allows for the probability that each feature is directly influenced by the appearance/non-appearance of at most two other features.

3 A New Inductive Learning Method for Multilabel Text Categorization

In this section, we present a new inductive learning method for mulitlabel text categorization. The mutual information measure shown in Eq. (1) is used to select the top K terms that have the highest MI scores for a category. Assume there are N documents and K selected terms in category c. We use a K × N document descriptor matrix to represent the binary weights of the K selected terms in each document. A column in the document descriptor matrix is a document descriptor vector based on the K selected terms. For example, assume that there are 5 documents d_1, d_2, d_3, d_4, d_5 and 4 selected terms t_1, t_2, t_3, t_4 in category c. Fig. 1 shows an example of a 4 × 5 document

descriptor matrix A. Each column of the document descriptor matrix A represents the document descriptor vector of a document. For example, from the second column of the document descriptor matrix A, we can see the document descriptor vector $\overline{d_2}$ of the document d_2, where $\overline{d_2} = [1 \quad 0 \quad 1 \quad 0]$. It indicates that the terms t_1 and t_3 are appearing in the document d_2 and the terms t_2 and t_4 are not appearing in the document d_2.

$$A = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}.$$

Fig. 1. Document descriptor matrix A

We use the "cosine similarity measure" [2] to construct a document-similarity matrix S, shown as follows:

$$S(i, j) = \frac{A(i) \cdot A(j)}{\|A(i)\| \times \|A(j)\|}, \tag{6}$$

where the value of $S(i, j)$ indicates the degree of similarity between document d_i and document d_j, $S(i, j) \in [0, 1]$, and $A(i)$ and $A(j)$ denote the ith and the jth column vectors of the document descriptor matrix A, respectively. For the document descriptor matrix A shown in Fig. 1, we can get its document-similarity matrix S, as shown in Fig. 2.

$$S = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 1 & 1/\sqrt{2} & \sqrt{3}/2 & 1/2 & 1/\sqrt{2} \\ 1/\sqrt{2} & 1 & 1/\sqrt{6} & 1/\sqrt{2} & 1/2 \\ \sqrt{3}/2 & 1/\sqrt{6} & 1 & 0 & 2/\sqrt{6} \\ 1/2 & 1/\sqrt{2} & 0 & 1 & 0 \\ 1/\sqrt{2} & 1/2 & 2/\sqrt{6} & 0 & 1 \end{bmatrix} \end{array}.$$

Fig. 2. Document-similarity matrix S

We can obtain the term-document relevance matrix R by applying the inner product of the document descriptor matrix A to the document-similarity matrix S, shown as follows:

$$R = A \cdot S, \tag{7}$$

where the value of $R(i, j)$ denotes the relevance degree of term t_i with respect to document d_j. Therefore, for the above example, we can get the term-document relevance matrix R, as shown in Fig. 3.

$$R = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & d_4 & d_5 \end{array} \\ \begin{bmatrix} 2.2 & 2.4 & 1.3 & 2.2 & 1.2 \\ 1.9 & 1.1 & 1.9 & 0.5 & 1.5 \\ 3.3 & 2.6 & 3.1 & 1.2 & 3 \\ 2.6 & 1.6 & 2.7 & 0.5 & 2.5 \end{bmatrix} \end{array}.$$

Fig. 3. Term-document relevance matrix *R*

We use Eq. (8) to get the category descriptor vector $\overline{v_c}$ for category *c*,

$$\overline{v_c} = R \cdot \bar{1}, \tag{8}$$

where $\bar{1} = [1, 1, ..., 1]^{\mathrm{T}}$. Thus, for the above example, we can get

$$\overline{v_c} = R \cdot \bar{1} = R \cdot [1 \quad 1 \quad 1 \quad 1]^{\mathrm{T}} = [9.3 \quad 6.9 \quad 13.2 \quad 9.9]^{\mathrm{T}}.$$

Then, we use the weight-averaged method to normalize $\overline{v_c}$. Thus, for the above example, $\overline{v_c}$ is normalized into $[0.24 \quad 0.17 \quad 0.34 \quad 0.25]$. Finally, we refine the weight v_{c_i} of the *i*th term in the category descriptor vector $\overline{v_c}$ into w_{c_i} to obtain the refined category descriptor vector $\overline{w_c}$, where

$$w_{c_i} = v_{c_i} \times \log_2 \frac{|C|}{cf_i}, \tag{9}$$

w_{c_i} denotes the refined weight of the *i*th term in the refined category descriptor vector $\overline{w_c}$, $|C|$ denotes the number of categories, and cf_i denotes the number of category descriptor vectors containing term t_i. This refinement reduces the weights of the terms that appear in most of the categories and increases the weights of the terms that only appear in a few categories.

Assume that the document descriptor vector of a testing document d_{new} is $\overline{d_{new}}$. We can then apply the inner product to calculate the relevance score $Score(c, d_{new})$ of category *c* with respect to the testing document d_{new} as follows:

$$Score(c, d_{new}) = \overline{d_{new}} \cdot \overline{w_c}. \tag{10}$$

We calculate the relevance score of each category with respect to d_{new}, rank these relevance scores, and then assign d_{new} to multiple categories according to Eq. (2). In other words, we choose the maximum relevance score *L* among them. If the relevance score between a category and the testing document divided by *L* is not less than a predefined threshold value λ, where $\lambda \in [0, 1]$, then the document is classified into that category.

4 Experimental Results

We have implemented the proposed multilabel text categorization method to classify the "Reuters-21578 Aptè split 10 categories" data set [21] using Delphi Version 5.0 on a Pentium 4 PC. The "Aptè split 10 categories" data set contains the 10 top-sized categories obtained from the "Reuters-21578 Aptè split" data set [20], where each category has at least 100 training documents for training a classifier. We chose the "Aptè split 10 categories" data set as our experimental data set, because it accounts 75% of the "Reuters-21578 Aptè split" data set. Table 1 shows the category names of "Aptè split 10 categories", the number of training samples for each category, and the number of testing samples for each category. There are totally 6490 training documents and 2547 testing documents in the "Aptè split 10 categories" data set.

Table 1. The number of training and testinging samples for each category of the "Aptè split 10 categories" data set

Category Names	Number of Training samples	Number of Testing samples
Earn	2877	1087
Acq	1650	719
Money-fx	538	179
Grain	433	149
Crude	389	189
Trade	369	118
Interest	347	131
Wheat	212	71
Ship	197	89
Corn	182	56

Several evaluation criteria for dealing with classification problems have been used in text categorization [15], [16]. The most widely used measures are based on the definitions of precision and recall. If a sample is classified into a category, we call it "positive" with respect to that category. Otherwise, we call it "negative". In this paper, we use the following micro-averaging method [15] to evaluate the recall and the precision of the proposed method, where

$$\text{Recall} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + \sum_{i=1}^{|C|} FN_i}, \tag{11}$$

$$\text{Precision} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + \sum_{i=1}^{|C|} FP_i}, \tag{12}$$

TP_i denotes the number of correctly classified positive samples for category c_i, FN_i denotes the number of incorrectly classified negative samples for category c_i, FP_i denotes the number of incorrectly classified negative samples for category c_i, and |C| denotes the number of categories.

If the values of the precision and the recall of a classifier can be tuned to the same value, then the value is called the break-even point (BEP) of the system [12]. BEP has been widely used in text categorization evaluations. If the values of the precision and the recall are not exactly equal, we use the average of the nearest precision and recall values as the BEP.

Based on the mutual information measure [11] for term selection, we select the top 300 terms for training classifiers. Table 2 compares the break-even point of the proposed method with those of four existing learning methods [8], namely, Find Similar, Decision Trees, Naïve Bayes and Bayes Nets. From Table 2, we can see that the proposed method gets a higher average accuracy than the existing methods.

Table 2. Breakeven performance for Reuters-21578 Aptè split 10 categories

Methods / Category	Find Similar	Naïve Bayes	Bayes Nets	Decision Trees	The Proposed Method ($\lambda = 0.87$)
Earn	92.9 %	95.9 %	95.8 %	97.8 %	97.5 %
Acq	64.7 %	87.8 %	88.3 %	89.7 %	95.1 %
Money-fx	46.7 %	56.6 %	58.8 %	66.2 %	79.2 %
Grain	67.5 %	78.8 %	81.4 %	85.0 %	84.7 %
Crude	70.1 %	79.5 %	79.6 %	85.0 %	84.4 %
Trade	65.1 %	63.9 %	69.0 %	72.5 %	85 %
Interest	63.4 %	64.9 %	71.3 %	67.1 %	81 %
Ship	49.2 %	85.4 %	84.4 %	74.2 %	85.4 %
Wheat	68.9 %	69.7 %	82.7 %	92.5 %	79.8 %
Corn	48.2 %	65.3 %	76.4 %	91.8 %	78.2 %
Average	64.6 %	81.5 %	85 %	88.4 %	91.3 %

5 Conclusions

In this paper, we have presented a new inductive learning method for multilabel text categorization. The proposed method uses a mutual information measure for term selection and constructs document descriptor vectors for each category based on the selected terms. These document descriptor vectors form a document descriptor matrix and they are also used to construct a document-similarity matrix based on the cosine similarity measure. It then constructs a term-document relevance matrix by applying the inner product of the document descriptor matrix to the document-similarity matrix. The proposed method infers the degree of relevance of the selected terms to construct the category descriptor vector of each category. The relevance score between each category and a testing document is calculated by applying the inner product of

its category descriptor vector to the document descriptor vector of the testing document. The maximum relevance score L is then chosen. If the relevance score between a category and the testing document divided by L is not less than a predefined threshold value λ, where $\lambda \in [0, 1]$, then the document is classified into that category. From the experimental results shown in Table 2, we can see that the proposed method gets a higher average accuracy than the existing methods.

Acknowledgements

This work was supported in part by the National Science Council, Republic of China, under Grant NSC 94-2213-E-011-003.

References

[1] Aptè, C., Damerau, F.J., Weiss, S.M.: Automatic Learning of Decision Rules for Text Categorization. ACM Transactions on Information Systems 1 (1997) 233–251
[2] Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)
[3] Bekkerman, R., Ran, E.Y., Tishby, N., Winter, Y.: Distributional Word Clusters vs. Words for Text Categorization. Journal of Machine Learning Research (2003) 1183–1208
[4] Caropreso, M.F., Matwin, S., Sebastiani, F.: A Learner-Independent Evaluation of the Usefulness of Statistical Phrases for Automated Text Categorization. In: Chin, A.G. (eds.): Text Databases and Document Management: Theory and Practice. Idea Group Publishing, Hershey PA (2001) 78–102
[5] Chinkering, D., Heckerman, D., Meek, C.: A Bayesian Approach for Learning Bayesian Networks with Local Structure. Proceedings of Thirteen Conference on Uncertainty in Artificial Intelligence. Morgan Kaufman, San Franscisco California (1997) 80–89
[6] Cohen, W.W., Singer, Y.: Context-Sensitive Learning Methods for Text Categorization. ACM Transactions on Information Systems. 17 (1999) 141–173
[7] Dhillon, I.S., Mallela, S., Kumar, R.: A Divisive Information-Theoretic Feature Clustering Algorithm for Text Classification. Journal of Machine Learning Research 3 (2003) 1265–1287
[8] Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive Learning Algorithms and Representation for Text Categorization. Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management. Bethesda MD (1998) 148–155
[9] Fuhr, N., Buckley, C.: A Probabilistic Learning Approach for Document Indexing. ACM Transactions on Information Systems 9 (1991) 323–248
[10] Fuhr, N., Pfeifer, U.: Probabilistic Information Retrieval as Combination of Abstraction Inductive Learning and Probabilistic Assumptions. ACM Transactions on Information Systems 12 (1994) 92–115
[11] Hankerson. D., Harris, G.A., Johnson, P.D., Jr.: Introduction to Information Theory and Data Compression. CRC Press, Boca Raton Florida. (1998)
[12] Lewis, D.D., Ringuetee, M.: Comparison of Two Learning Algorithms for Text Categorization. Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (1994) 81–93

[13] Rocchio, J.J.: Relevance Feedback in Information Retrieval. In Salton G. (eds.): The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice Hall New Jersey (1971) 313–323
[14] Sahami, M.: Learning Limited Dependence Bayesian Classifiers. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press, (1996) 335–338
[15] Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34 (2002) 1–47
[16] Yang, Y.: An Evaluation of Statistical Approaches to Text Categorization. Information Retrieval 1 (1999) 69–90
[17] Yang, Y., Chute, C.G.: An Example-based Mapping Method for Text Categorization and Retrieval. ACM Transactions on Information Systems 12 (1994) 252–277
[18] Yang, Y., Liu, X.: A Re-examination of Text Categorization Methods. Proceedings of SIGIR-99 22th ACM International Conference on Research and Development in Information Retrieval. Berkeley, California (1999) 42–49
[19] Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. Proceedings of ICML-97 14th International Conference on Machine Learning. Nashville, TN (1997) 412–420
[20] Reuters-21578 Aptè split data set, http://kdd.ics.uci.edu/data-bases/reuters21578/reuters21578.html
[21] Reuters-21578 Aptè split 10 categories data set, http://ai-nlp.info.uniroma2.it/moschitti/corpora.htm