Classification of Web Documents Using a Naive Bayes Method

Yong Wang, Julia Hodges, Bo Tang
Department of Computer Science & Engineering, Mississippi State University
Mississippi State, MS 39762-9637
ywang@cse.msstate.edu, hodges@cse.msstate.edu, btang@cse.msstate.edu

Abstract

This paper presents an automatic document classification system, WebDoc, which classifies Web documents according to the Library of Congress classification scheme. WebDoc constructs a knowledge base from the training data and then classifies the documents based on information in the knowledge base. One of the classification algorithms used in WebDoc is based on Bayes' theorem from probability theory. This paper focuses upon three aspects of this approach: different event models for the naive Bayes method, different probability smoothing methods, and different feature selection methods. In this paper, we report the performance of each method in terms of recall, precision, and F-measures. Experimental results show that the WebDoc system can classify Web documents effectively and efficiently.

1. Introduction

Document classification refers to the task of "developing a system that is able to automatically classify a text document into a number of categories relevant to the document" [10]. Due to the extensive use of the World Wide Web, the huge amounts of information on the Web make an attractive resource. The lack of logical organization of Web documents makes retrieving relevant information from the Web a laborious and time-consuming task, and motivates the development of automatic Web document classification systems. Automatic document classification is an active and challenging field of research, and an extensive range of algorithms has been proposed. Typically-used methods include the decision tree method [1], k-nearest neighbor method (kNN) [6][11][20], Naive Bayes method (NB) [12][13][19], Bayesian networks [7][10], neural networks (NNet) [2][3][18], support vector machines (SVM) [6][8], and subspace model [11].

This paper describes an automated document classification system, WebDoc (The Web Document Classification System), which was developed by researchers in the Department of Computer Science and Engineering at Mississippi State University. WebDoc uses the Library of Congress classification scheme to classify HTML documents that have been downloaded from the Web. The WebDoc system introduced in this paper was implemented using a naive Bayes method based on Bayes' theorem from probability theory. The study is focused upon two different Naive Bayes models: a multi-variate Bernoulli event model and a multinomial event model. In this paper, two different probability smoothing methods were tested: additive smoothing method and Good-Turing smoothing method. Four feature selection criteria were tested: inverse document frequency (IDF), information gain (IG), mutual information (MI) and χ^2 (CHI). In the WebDoc system, the Library of Congress Subject Headings (LCSHs) is used as the indexes for the Web documents.

The rest of this paper is organized as follows. In section two, we begin with an overview of the WebDoc classification system. We follow that with an introduction of how to use the naive Bayes method in a document classification system in section 3. In sections 4, 5, and 6, we describe the different naive Bayes models, different smoothing methods, and different feature selection criteria used in the WebDoc system separately. In section 7, we presented our experimental results. This is followed by the conclusions and future work in section 8.

2. Overview of WebDoc system

The Web Document Classification System (WebDoc) is a research project in the Department of Computer Science at Mississippi State University (MSU). The goal of the WebDoc system is to use the Library of Congress classification scheme, one of the world's most widely used classification schemes, to classify documents that were downloaded from the Web. WebDoc constructs a knowledge base from the training data and then classifies the documents based on information in the knowledge base. The WebDoc system consists of three major components, as illustrated in Figure 1[16]: the NLP component, the knowledge base construction component, and the index generation component. The NLP tags the original Web document with syntactic and semantic tags (such as noun and astronomy) and parses the document (thus making it possible to isolate sentential components such as noun phrases). The knowledge base construction component builds a knowledge base of information that includes the Library of Congress (LCC) subject headings and their interrelationships as well as other information used during classification. The index generation component generates a set of candidate indexes for each document in a test set of documents. (In this paper, we use the terms subject headings and indexes to mean the same thing.)

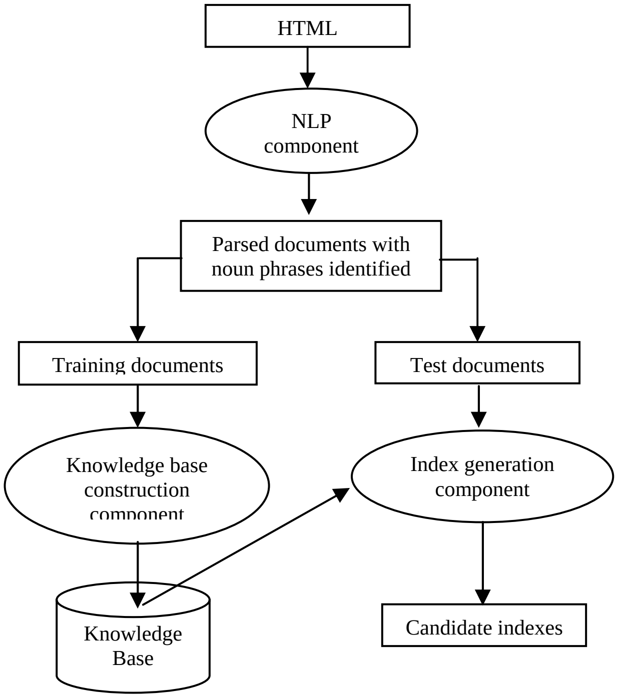
HTML
NLP component
Parsed documents with noun phrases identified
Training documents
Test documents
Knowledge base construction component
Index generation component
Knowledge Base
Candidate indexes

Figure 1. Architecture of the WebDoc system

3. Naive Bayes classification algorithm

The naive Bayes method (NB) is a simple Bayesian classifier based on Bayes' theorem from probability theory. In the WebDoc system, the stem forms of words occurring in the training documents were used as the features to represent each document. The basic steps in the naive Bayes method are as follows:

Training:

- Identify the individual stem words occurring in all the training documents in the training set.
- Generate the feature vector for each document in the training document set and store it along with the correct indexes in the knowledge base.
- Calculate the probability for each index.

Testing:

- Identify the individual stem words occurring in a given test document.
- Generate the feature vector for this document.
- Calculate the probability for this document given each index.
- Calculate the probability for each index in the set of indexes for this document and normalize it with Bayes' theorem, this value is the weight of this index.
- Select the indexes with a weight higher than a predefined threshold as the candidate indexes for this document.

4. Naive Bayes models

Although a naive Bayes classifier is a simple and popular technique used in the document classification area, it has been implemented by different researchers with two different generative models: multi-variate Bernoulli event models and multinomial event models [12]. In the multi-variate Bernoulli model, a binary representation is used for the value of a feature in the feature vector, which mean the possible value for each feature is only 0 or 1. A value of 1 for feature A_i indicates that feature A_i (stem form of a noun phrase) occurred in that document ($x_i = 1$). A value of 0 for A_i indicates that feature A_i did not occur in that document ($x_i = 0$). The occurrence frequencies of these features in the documents are not captured. In this model, a document is seen as an "event" and the absence or presence of words is an attribute of the event. The probability of a feature vector for a test document is the product of all the attribute values, including the probability of occurrence for the features that do occur in the document and the probability of non-occurrence for the features that do not occur in the document. In the multinomial event model, the number of occurrences of each feature Ai (stem form of a noun phrase) in the document is captured and each feature vector is represented by a list of occurrence frequencies of all features. The value 0 of feature Ai means that this feature did not occur in the document. In order to avoid the effect of the varying lengths of the documents, all the occurrence frequencies are normalized before being used. In this model, the occurrences of each individual feature are the "events" and a document is a collection of word events.

5. Probability smoothing

Smoothing is a "technique used to better estimate probabilities when there is insufficient data to estimate probabilities accurately" [4]. The goal of various smoothing techniques is to make the distribution of probabilities more uniform. One principle of the various smoothing methods is the sum of the all probabilities must be 1. Two smoothing methods were used in the WebDoc system, additive smoothing and Good-Turing smoothing.

The additive smoothing method is one of the simplest smoothing methods used in practice. In this method, the occurrence frequency of each feature was increased by 1. Then the estimated probability of each feature given a conclusion can be calculated with the following formulation:

$$P(A_i|C_j) = \frac{N_{ij}+1}{\sum_{i=1}^{n} N_{ij} + n}$$

where N_{ij} is the actual occurrence frequency of feature A_i given conclusion C_j and n is the size of feature vector.

The Good-Turing smoothing method is based on the Good-Turing estimate. In this method, the probability of an occurred feature is replaced with a smaller probability. The sum of the smaller probabilities is subtracted from 1.0; this difference is distributed evenly among the unseen features. One of the implementations of Good-Turing smoothing methods is called the Linear Good Turing (LGT) estimate [4]. In this method, let r represent the frequency of a given feature. Then N_r is the number of features with a frequency of r. The value r^* is the estimated frequency, which is calculated based upon the frequencies and the N_r values:

$$r^* = (r+1)\frac{E(N_{r+1})}{E(N_r)}$$

where $E(x)$ is the expectation of the random variable x. Most of the N_r values will be 0 for a large value of r. To account for these 0s, Church and Gale average with each nonzero value N_r the zero N_r values that surround it. Order the nonzero values by r. Let q, r, and t be successive indexes of nonzero values. Replace N_r by Z_r:

$$Z_r = \frac{N_r}{0.5(t-q)}$$

So the expected N_r is estimated by the density of N_r for large r.

Let b represent the slope of the line defined where the x-axis represents $log(r)$ and the y-axis represents $log(Z_r)$. Then r^* is calculated as:

$$r^* = r(1+\frac{1}{r})^{b+1}$$

The probability of each feature that occurred at least once is r^*/N where N is the sum of the frequencies. The difference between 1.0 and the sum of the nonzero probabilities is distributed evenly among the non-occurring features.

6. Feature Selection

The goal of the feature selection is to try to remove non-informative features and reduce the dimensionality of the feature vector [21]. Four feature selection methods were used in the WebDoc system: inverse document frequency (IDF), information gain (IG), mutual information (MI), and χ^2 (CHI Square).

Inverse document frequency is computed based on collection frequency. The collection frequency of a term is the number of documents in which that term occurs [14]. The IDF value of term i is $log(N/N_i)$, where N is the total number of documents in the collection and N_i is the collection frequency of term i [15].

Information gain (IG) is a measure based on entropy [5]. This method measures how much additional information you can get from each feature by including a particular index and select the optimal one. Given a set of training documents whose size is s and the size of each category is s_i, the expected information that is needed to classify a given document is

$$I(s_1, s_2, ..., s_m) = -\sum_{i=1}^{m}\frac{s_i}{s}\log\frac{s_i}{s}$$

For each feature of the feature vector A, assume it has v different values, the information gain of feature A based on the entropy is:

$$\mathrm{E(A)} = \sum_{j=1}^{v}\frac{s_{1j}+...+s_{mj}}{s} I(s_{1j},...s_{mj})$$

The definition of mutual information used in our experiments was adapted from the one used by Yang and Pedersen [21]. As before, let A represent a feature in the feature vector and let C represent a subject heading. The mutual information between A and C is:

$$I(A, C) = log\ \frac{P(A_i \cap C_j)}{P(A_i)\times P(C_j)}$$

The mutual information may be estimated as

$$I(A, C) \approx log\ \frac{a\times n}{(a+c)\times(a+b)}$$

Where n is the size of training documents. a is the number of documents in which both A and C occur. b is the number of documents in which A occurs but C. c is the number of documents in which C occurs but A. The final MI value for a feature is the average of all values for different categories.

The χ^2 (CHI) method is a method similar to the MI method [21]. Assume d is the times when none of A and C occurs, the estimation of χ^2 value of A and C is:

$$\chi^2 (A, C) = \frac{n\times(ad-cb)^2}{(a+c)\times(b+d)\times(a+b)\times(c+d)}$$

7. Experiments

A total of 722 documents downloaded from the Web were used as the data in our experiments. All these documents have been assigned the correct LCSHs by an expert librarian. The 5-fold cross-validation method was used to divide the documents into a training set and test set. The performance of the WebDoc system was evaluated by the well-known measures of precision, recall, and F-measure. For each experiment, all the candidate indexes were filtered according to their weight. In order to make a comparison, all the weights were normalized into the range from 0 to 1. Then the threshold was set from 0 to 0.9 with step 0.1.

The experimental results for two naive Bayes models were given in table 1. The comparison results for different smoothing methods were listed in table 2. In the table 3 and table 4, the experimental results for four feature selection methods were listed.

8. Conclusions and future work

WebDoc is an automated classification system that assigns Web documents to appropriate Library of Congress subject headings based upon the text in the documents. In this paper, the architecture and design of WebDoc were presented. WebDoc used the Bayes' theorem as basic algorithm and was implemented with two different models: a multi-variate Bernoulli event model and a multinomial event model. Two different probability smoothing methods and four different feature selection measures were applied in the WebDoc. Our experimental results indicate that:

First, compared with the previous versions of WebDoc, whose results were reported in [17], we obtained an increase in the F-measure of almost 20 percentage points (i.e., 67.19%).

Second, compared with the reported results of other automated document classification systems, the performance of WebDoc is favorable, especially considering that some of those researchers whose systems had higher recall, precision, and/or F-measures than ours were not attempting to classify documents as unstructured and varied as the Web documents that we worked with. For example, in [13], the total number of categories used by Quek's web document classification system is only seven (Course, Student, Faculty, department, Staff, Research project, and other). And the experimental data is limited to the homepages of the computer science departments of four universities. In [20], Yang made a comparison of ten different classification algorithms on the Reuters corpus, which is a standard data set for the evaluation of document classification systems. The BEP value achieved by yang's naive Bayes method is 66%, which is also similar to the performance of WebDoc system.

Third, in the WebDoc system, the multinomial event model classifier had a better performance than the multi-variate Bernoulli event model. This result is consistent with that in [12].

Fourth, two smoothing methods, additive smoothing and the Good-Turing smoothing methods, increased the recall value of the classifier greatly but decreased the precision. The F-measure results demonstrate that when a higher threshold is set, both smoothing methods are helpful for generating more correct indexes and did improve the performance of the classifier.

Fifth, although four different feature selection methods were used in the WebDoc, none of them improved the performance notably.

Based on the current results of the WebDoc system, the following are appropriate areas of future study:

Table 1. Multi-variate Bernoulli Model (MB) vs. Multinomial Model (MN)

Thre.	Precision		Recall		F-Measure	
	MB	MN	MB	MN	MB	MN
0	0.14	0.14	1	1	0.25	0.25
0.1	0.17	0.3	0.87	0.99	0.29	0.46
0.2	0.2	0.41	0.87	0.92	0.32	0.57
0.3	0.23	0.55	0.87	0.78	0.36	0.65
0.4	0.24	0.61	0.87	0.71	0.38	0.66
0.5	0.27	0.63	0.87	0.68	0.41	0.65
0.6	0.32	0.63	0.87	0.66	0.47	0.65
0.7	0.35	0.66	0.83	0.63	0.49	0.64
0.8	0.53	0.68	0.69	0.48	0.6	0.57
0.9	0.63	0.72	0.63	0.43	0.63	0.54

Table 2. Experimental results of smoothing methods

Thre.	Precision			Recall			F-Measure		
	NO	Add	GT	NO	Add	GT	NO	Add	GT
0	0.14	0.14	0.14	1	1	1	0.25	0.25	0.25
0.1	0.3	0.16	0.18	0.99	1	1	0.46	0.28	0.31
0.2	0.41	0.19	0.21	0.92	1	1	0.57	0.32	0.35
0.3	0.55	0.22	0.23	0.78	1	1	0.65	0.35	0.37
0.4	0.61	0.23	0.25	0.71	1	1	0.66	0.38	0.39
0.5	0.63	0.25	0.27	0.68	1	1	0.65	0.41	0.42
0.6	0.63	0.3	0.33	0.66	1	1	0.65	0.46	0.5
0.7	0.66	0.39	0.45	0.63	0.97	0.94	0.64	0.56	0.61
0.8	0.68	0.56	0.58	0.48	0.8	0.79	0.57	0.66	0.67
0.9	0.72	0.63	0.63	0.43	0.7	0.69	0.54	0.66	0.66

Table 3. Experimental results of feature selection methods (IDF vs. IG)

Thre.	Precision		Recall		F-Measure	
	IDF	IG	IDF	IG	IDF	IG
0	0.14	0.14	1	1	0.25	0.25
0.1	0.22	0.28	1	0.99	0.36	0.44
0.2	0.3	0.37	0.99	0.93	0.46	0.53
0.3	0.39	0.47	0.93	0.81	0.55	0.59
0.4	0.5	0.53	0.82	0.74	0.62	0.62
0.5	0.59	0.57	0.73	0.69	0.65	0.62
0.6	0.63	0.6	0.68	0.67	0.65	0.63
0.7	0.66	0.62	0.65	0.64	0.66	0.63
0.8	0.69	0.66	0.57	0.5	0.63	0.57
0.9	0.74	0.7	0.43	0.44	0.55	0.54

Table 4. Experimental results of feature selection methods (MI vs. χ^2)

Thre.	Precision		Recall		F-Measure	
	MI	χ^2	MI	χ^2	MI	χ^2
0	0.14	0.14	1	1	0.25	0.25
0.1	0.23	0.28	1	0.98	0.37	0.43
0.2	0.29	0.37	0.99	0.93	0.45	0.53
0.3	0.34	0.46	0.93	0.8	0.5	0.58
0.4	0.4	0.51	0.82	0.74	0.54	0.6
0.5	0.48	0.55	0.75	0.7	0.59	0.62
0.6	0.57	0.58	0.69	0.68	0.62	0.63
0.7	0.62	0.6	0.67	0.64	0.64	0.62
0.8	0.64	0.65	0.58	0.5	0.61	0.57
0.9	0.67	0.68	0.47	0.41	0.55	0.51

First, Try different classification algorithms and attempt to combine them. Several researchers have found improvement in the performance of their text classification systems when they used some other classification method such as a K-nearest neighbor method [6][11][20] and Support Vector Machines [6][8]. It would be interesting to experiment with various ways of accomplishing these methods and comparing the resulting performance with that reported here.

Second, collect more data for the feature selection. Feature selection is an important topic in the data mining area and the methods used in WebDoc were reported to work well in many other systems [9][21]. The most likely reason for their failure to improve the performance of WebDoc is the lack of a large number of training documents. The small number of documents is also an important reason that affects the training of the classifier.

9. Acknowledgments

We are indebted to Dr. Lois Boggess and the students who worked under her direction (Janna Hamaker, Don Goodman, and Emily Stone) for their contributions to the WebDoc project, particularly the natural language processing component. We also wish to express our appreciation to David Mays, who served as our expert classification librarian and whose classifications of the Web Documents served as the "gold standard" in evaluating the performance of WebDoc.

This work has been supported by grant number IIS973480700050383 from the National Science Foundation.

10. References

[1] S. J. Cunningham and B. Summers, "Applying Machine Learning to Subject Classification and Subject Description for Information Retrieval," *Proc. 2nd New Zealand International Two-stream Conference on Artificial Neural Networks and Expert Systems*, 1995, IEEE Computer Society, pp. 243-246.

[2] J. Farkas, "Neural Networks and Document Classification," *Proc. Canadian Conference on Electrical and Computer Engineering*, vol. 1, 1993, IEEE Computer Society, pp. 1-4.

[3] J. Farkas, "Improving the Classification Accuracy of Automatic Text Processing Systems Using Context Vectors and Back-propagation Algorithms," *Proc. Canadian Conference on Electrical and Computer Engineering,* vol. 2, 1996, IEEE Computer Society, pp. 696–699.

[4] W. A. Gale, "Good-Turing Smoothing Without Tears," *Journal of Quantitative Linguistics*, vol. 2, 1995, pp. 217-237.

[5] J. Han, and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, California, 2001.

[6] J. He, A. Tan, C. Tan, "Machine Learning Methods for Chinese Web Page Categorization," *Proc. 2nd Chinese Language Processing Workshop (ACL'2000),* Hong Kong, 8 October 2000, pp. 93-100.

[7] J. Her, S. Jun, J. Choi, and J. Lee. "A Bayesian Neural Network Model for Dynamic Web Document Clustering," *Proc. IEEE region 10 conference on TENCON 99*, vol. 2, no. 1, 1999, IEEE Computer Society, pp. 415-418.

[8] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," *Proc. 10th European Conference on Machine Learning (ECML-98),* 1998.

[9] D. Koller and M. Sahami, "Toward Optimal Feature Selection," *Proc. 13th International Conference on Machine Learning (ICML)*, Bari, Italy, Jul. 1996, pp. 284-292.

[10] W. Lam and K. Low, "Automatic Document Classification Based on Probabilistic Reasoning: Model and Performance Analysis," *Proc. 1997 IEEE International Conference on Computational Cybernetics and Simulation*, vol. 3, 1997, IEEE Computer Society, pp. 2719-2723.

[11] Y. Li and A. K. Jain, "Classification of Text Documents," *Proc. 14th International Conference on Pattern Recognition,* vol. 2, 1998, pp. 1295-1297.

[12] McCallum and K. Nigam. "A Comparison of Event Models for Naive Bayes Text Classification," *Working notes of the 1998 AAAI/ICML workshop on learning for text categorization,* 1998.

[13] Y. Quek, *Classification of World Wide Web Documents*, Master Thesis, Carnegie Mellon University, 1997.

[14] G. Salton, "A Blueprint for Automatic Indexing," *SIGIR Forum* vol. 31, no. 1, 1997, pp. 23-36 (reprinted from *SIGIR Forum,* vol. 16, no. 2*,* 1981).

[15] G. Salton and C. Buckley, "Term Weighting Approaches in Automatic Text Retrieval," *Information Processing and Management*, vol. 24, no. 5, 1988, pp. 513-523.

[16] B. Tang, *Knowledge Discovery in Context-Based Automatic Web Document Indexing,* Ph. D. Dissertation Proposal, Department of Computer Science, Mississippi State University, Mississippi, 1999.

[17] B, Tang and J. Hodges, "Web Document Classification with Positional Context," *Proc. International Workshop on Web Knowledge Discovery and Data Mining (WKDDM'2000)*, Keihanna Plaza, Kyoto, Japan, Apr. 2000.

[18] N. Vlajic and H. C. Card, "An Adaptive Neural Network Approach to Hypertext Clustering," *Proceedings: International Joint Conference on Neural Networks, IJCNN '99*, vol. 6, 1999, IEEE Computer Society, pp. 3722-3726.

[19] B. Wang, S. Zhou, and Y. Hu, "Naive Bayes-Based Gradual Chinese Documents Categorization," *Proc. World Multiconference on Systemics, Cybernetics and Informatics*, Vol 2, Orlando, Florida, July, 2001, IEEE Computer Society, pp. 516 - 521.

[20] Y. Yang, "An Evaluation of Statistical Approaches to Text Categorization," *Journal of Information Retrieval*, vol. 1, no. 1/2, 1999, pp. 67-88.

[21] Y. Yang and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization," *Proceedings: Fourteenth International Conference on Machine Learning (ICML'97)*, 1997, pp. 412-420.