Text Categorization Approach For Chat Room Monitoring

EIMAN ELNAHRAWY
Department of Computer Science
University of Maryland
College Park, MD 20742, USA
USA

Abstract: The Internet has been utilized in several real life aspects such as online searching, and chatting. On the other hand, the Internet has been misused in communication of crime related matters. Monitoring of such communication would aid in crime detection or even crime prevention. This paper presents a text categorization approach for automatic monitoring of chat conversations since the current monitoring techniques are basically manual, which is tedious, costly, and time consuming. This paper presents the results of a cross method comparison between the Naive Bayes, the K-nearest neighbor, and the Support Vector Machine classifiers. The objective is to determine the most suitable method for the data of chat conversations that would automate the chat room monitoring task.

Key-Words: Monitoring, Chat conversations, Internet, Machine learning, Text categorization, Artificial Intelligence, Data mining

1 Introduction

The Internet has been used extensively over the past few years in many real life aspects such as sending e-mails, online searching, cooperating in collaborative environments by chatting, or browsing the tremendous amount of online information. Currently, there are several chat tools available on the Internet. Some of them are very popular such as ICQ [1], and mIRC [2]. The emergence of such tools has enriched the communication between the Internet users from all over the world.

Another important issue that emerged with the breakthrough of the Internet is the misuse of technology in communication of crime related matters, abduction of children and young teenagers via e-mails or chat rooms, etc., since these are tougher to wiretap. Monitoring of such tools would aid in crime detection or even crime prevention. In particular, monitoring of chat conversations and classifying the chats as being suspicious or not suspicious is very advantageous.

With this rapid growth of the Internet, the efficacy of text categorization in real life has also become very clear as one of the most important applications of machine learning techniques. Text categorization is the problem of assigning predefined categories to natural language text documents based on their contents [3, 4, 5]. Developing techniques for automatic text categorization is very advantageous since manual classification is tedious, and time consuming. Automatic text categorization is currently used as an efficient tool to classify e-mails (e.g., bulk e-mails), and to guide users in their searches on the Internet (e.g., search engines).

This paper presents a text categorization approach for automation of chat room monitoring. The motivation is that the current monitoring techniques are basically manual [6], which is difficult, costly, and time consuming. Specifically, this paper evaluates the application of three text categorization machine learning methods to the problem of chat room monitoring.

The paper is organized as follows. An overview of the chat room monitoring problem, as well as the proposed text categorization solution for this problem, is presented in Section 2. Section 3 gives an overview of the text cat-

egorization approach. A description of our experiments along with their results are presented in Section 4. Finally, Section 5 concludes this paper and discusses possible extensions.

2 Monitoring Chat Rooms

2.1 Related Work

In general, there is not much research on monitoring chat room conversations for criminal activities. The current monitoring techniques are basically manual [6], which is difficult, tedious, costly, and time consuming. Another approach that has emerged recently is based on sniffing, for example, [6], in order to avoid manual monitoring techniques. The approach is essentially based on monitoring single packets from the chat conversation.

2.2 Log-Based Chat Room Monitoring

The packet sniffing approach is clearly not appropriate for making precise assessment about chat conversations. This is due to the nature of the chat conversation, where the entire conversation should be available in order to make any assessment about it. Monitoring separate parts of the conversation affects the overall context of the conversation; and in turn affects the accuracy of the monitoring technique.

Alternatively, we propose an approach for automatic monitoring of chat conversations based on classifying the entire chat session log off-line using an appropriate classifier. Using this approach, better assessment can be made since all the information about the session is used.

The current available chat tools can easily support logging of chat conversations. Most of these tools utilizes the IRC protocol [7], or a variant of it. In general, they are based on the client-server model where there is one or more central server connected together [1, 2, 7]. This network structure facilitates logging of chat conversations without introducing extra overhead on the server or the client.

2.3 Text Categorization and Chat Room Monitoring

A number of approaches such as Bayes classifiers and other machine learning techniques including K-nearest neighbors classifiers, decision trees, Support Vector Machine, neural nets, and many others have been applied to the text categorization problem [3, 8]. In general, there is much ongoing research on comparing the performance of different text categorization techniques in real life aspects using several data sets [3, 4, 5, 9, 10, 11]. The published results have shown different evaluations of the classifiers using different data collections. These results indicated that the performance of the evaluated techniques is highly dependent on the data collection [4, 10]. The Naive Bayes, K-nearest neighbors, Rocchio, neural nets, and decision trees methods are the most popular evaluated methods. Recently, the Support Vector Machine method showed good results when applied to the text categorization problem. This recent research motivated the choice of the Naive Bayes, K-nearest neighbors, and the Support Vector Machine methods in this study. Specifically, this paper presents the results of a cross method comparison between the Naive Bayes, the K-nearest neighbor, and the Support Vector Machine classifiers. The objective of this study was to determine the most suitable method for the data of this particular problem that would automate the chat room monitoring task. To the best of our knowledge, there is no published cross method comparison for this particular type of data.

3 Text Categorization Approach

This section gives an overview of representation of text data, as well as existing text categorization algorithms.

3.1 Data Representation

Text categorization is not a trivial problem. The complexity of the problem lies in how to define a similarity metric between the documents, and then how to implement a computationally efficient algorithm to solve the problem given this similarity metric. The documents are first transformed to a suitable representation [8, 12, 3]. The most used approach for representing the documents is to con-

vert each document to a vector of terms (features) where the terms are selected from the vocabulary of the documents collection. The words that carry no information are called stopwords and they are usually removed from the collection vocabulary, for example, the words "the", "of", etc [3, 13]. The value of each term in the vector is the count of number of times the word appears in the document. Some terms may have a zero value if the corresponding word does not appear anywhere in the document.

3.2 Text Categorization Algorithms

An overview of the Naive Bayes, the K-nearest neighbors, and the Support Vector Machine classifiers is given in this section. Several variants of these methods and other methods such as Decision Trees, Rocchio, Linear Least Squares Fit, Neural Nets, and Maximum entropy are also applied to the text categorization problem [8, 14]. For more details, please refer to [8, 12, 15].

3.2.1 Naive Bayes Method

The Naive Bayes classifier [3, 8, 12] is one of the most effective probabilistic approaches currently known for text categorization. The method is considered *naive* due to its assumption that every word in the document is conditionally independent from the position of the other words given the category of the document. Despite the fact that this assumption is not true in general, the Naive Bayes classifier is surprisingly effective for this particular problem. The classifier learns a set of probabilities from the training data during the learning phase. It then uses these probabilities and the Bayes theorem to classify any new documents. The category of a new document is determined in two steps as follows:

- An estimate of the probability of the new document belonging to each class given its vector representation is calculated.
- The class with the highest probability is chosen as a predicted categorization.

3.2.2 K-Nearest Neighbors Method (KNN)

The K-nearest neighbors algorithm [3, 8] is one of the most basic learning methods. It assumes that all documents correspond to points in an n dimensional space where n is the dimension of the document vector representation. The nearest neighbors of a document are defined in terms of the standard Euclidean distance. The method does not have a training phase and the main computation occurs when we need to classify a new document. The classifier then classifies the new document based on the classification of its K nearest neighbors in the training data using the Euclidean distance as a distance metric.

3.2.3 Support Vector Machine Method (SVM)

The Support Vector Machine technique [3, 9, 15, 12] has been applied recently to the text categorization problem. It finds a hypothesis that minimizes the true error, where the true error is the probability of misclassifying an unseen and randomly selected test instance. The basic idea is to integrate the dimension reduction and the classification problems which gives a good generalization when applied to a wide variety of classification problems even in high dimensional data. The dimension reduction is performed by transforming the original data vectors, using combinations of the vector variables, from the original vector space to a new space using kernel functions. The transformation step is necessary since it is not always possible to find linear decision surfaces between the categories in the original vector space. This means that linear decision surfaces between the categories in the new space correspond to non-linear surfaces in the original space. The major drawback of this method is the use of optimization techniques which are very challenging and computationally expensive when the data set is large.

4 Experimental Results

In this study, a comparison between the Naive Bayes, the K-nearest neighbor, and the Support Vector Machine with linear kernel function text categorization methods was performed with respect to the accuracy of classification and the speed of the method. The accuracy of classification is a measure of how well the method performs on a new unseen and randomly selected data instance, while

the speed is a measure of how fast the method learns the training data and how fast it classifies new data instances. The goal is to determine the most suitable method for the classification of chat session logs. The choice of the linear kernel function was due to the fact that most text categorization problems are linearly separable [9, 12]. This section presents the nature of the training and the test data sets, the results of this experiment, and a discussion of these results.

The Rainbow [14, 13] software package was used in this experiment. The package contains several text categorization classifiers. The classifiers were first trained using the training data set, then they were used for classifying new instance.

4.1 Training Data

Two different data sets were used for training the classifiers in this study. The first data set is a collection of newsgroup messages. The second data set is a collection of chat session logs.

The first data set consists of 20,000 newsgroup messages drawn from 20 different newsgroups. The collection contains 1000 documents from each of the 20 newsgroups. The motivation behind the choice of this particular data set is that the structure of a newsgroup message is very similar to a chat session log since both documents basically represent a discussion about certain topic(s). Therefore, the results obtained using the newsgroups collection can help in making assessment about the behavior of the three compared methods on chat session logs.

The second data set consists of 40 chat session logs collected from four different web sites on the Internet. The topics for these sessions are Investment, Satellite, Java, and Space. It contains 10 long chat session logs (20 KB each) for each of the four topics.

The Support Vector Machine method uses a very challenging optimization problem during the training phase, which becomes even harder with the increase in the data dimensionality, as in the case of our data collections. This suggested using a subset of the newsgroup data set in the comparison between the Support Vector Machine and the other two classifiers. This subset consists of 100 randomly selected messages from each newsgroup.

To summarize, only the Naive Bayes and the K-nearest neighbor methods were cross evaluated using the first data collection which consists of 20,000 newsgroup messages. All three methods were cross evaluated using a subset of the newsgroup messages. The three methods were also evaluated using the second data set which consists of 40 chat session logs.

4.2 Test Data

For each data set, the data was split randomly into a training set and a test set using three different percentages, 80%, 60%, and 50% of the data going into a training set. The training data was used for training the classifier while the test data was used in the evaluation of the classifier.

4.3 Results

Training Data	Naive Bayes		K-nearest	
	Accuracy	Std	Accuracy	Std
80%	81.91%	0.28	35.45%	0.0
60%	80.89%	0.09	35.78%	0.0
50%	80.85%	0.23	36.99%	0.0

Table 1: Results for the complete newsgroups data set

Training Data	Naive Bayes		K-nearest		SVM	
	Accuracy	Std	Accuracy	Std	Accuracy	Std
80%	68.1%	0.82	35.55%	0	47.95%	13.49
60%	66.22%	0.67	38.73%	0	51.02%	14.3
50%	64.86%	0.6	38.14%	0	49.39%	13.8

Table 2: Results for the mini newsgroups data set

Training Data	Naive Bayes		K-nearest		SVM	
	Accuracy	Std	Accuracy	Std	Accuracy	Std
80%	100%	0	97.5%	0	41.81%	9.52
60%	100%	0	96.25%	0	54.96%	10.85
50%	100%	0	96.0%	0	57.99%	9.86

Table 3: Results for the chat session logs data set

The training and the testing were repeated five times for each experiment for validation. The average accuracy and the average standard error were then computed. For the K-nearest neighbors, K was set to 30 for the newsgroup data set, and to 3 for the chat session logs data set. A linear kernel function was used in the Support Vector

Machine method. Table 1 summarizes the results for the newsgroup data set; Table 2 summarizes the results for the mini newsgroup data set, and Table 3 summarizes the results for the chat session logs data set.

The experimental results obtained by using the newsgroup data set in training show the following.

- The Naive Bayes classifier is simple and fast in learning and in classification. In addition, it performs surprisingly well and significantly outperforms the K-nearest neighbor method.
- The Support Vector Machine classifier is very computationally expensive which makes it very slow to learn compared to the Naive Bayes classifier even for small data sets. However, it gives better results than the K-nearest neighbor classifier with respect to the accuracy of classification.
- The K-nearest neighbor performed poorly for the newsgroup messages data collection. This is likely due to the high dimensionality of the feature space. It is also slower than the Naive Bayes classifier with respect to the classification time. This difficulty makes it inefficient to use in real time applications or in applications that require a quick response.

The experimental results obtained by using the chat session logs data set show that the K-nearest neighbor classifier competes with the Naive Bayes classifier with respect to the classification accuracy, but it is still considerably slower than the Naive Bayes classifier. This result is likely due to the fact that the feature space (vocabulary) is small due to the limited size of the data set. Also, the Support Vector Machine classifier behaves poorly as compared to the other two classifiers. It is clear that we need to use a larger training data set for the chat session logs to confirm these results and to make an assessment about the performance of the three classifiers for this particular data set.

5 Conclusions and Future Work

The study showed that the problem of automatically monitoring chat rooms can be solved efficiently by using the appropriate text categorization methods. The choice of the appropriate categorization method depends on two factors: the accuracy of the classification and the efficiency of the method. A comparison between the Naive Bayes, the K-nearest neighbor, and the Support Vector Machine classifiers was performed. The objective of this study was to determine the most suitable method for the classification of chat sessions logs that would help in automatically monitoring the chat rooms, and in avoiding the manual techniques.

Two different training data sets were used for training the classifiers in this experiment, a collection of newsgroup messages, and a collection of chat session logs. The study showed that the Naive Bayes classifier is simple and fast while the Support Vector Machine classifier is very computationally expensive compared to the other two classifiers even for small data sets. The K-nearest neighbor classifier performed poorly when applied to the newsgroups data set. In general, the Naive Bayes classifier outperformed the K-nearest neighbor and the Support Vector Machine with respect to the classification accuracy. Also its training and classification times were considerably short. The results suggest that a simple Naive Bayes algorithm might be an appropriate choice for this problem since its training and classification times are considerably short and it also performs well with respect to the accuracy of classification. However, further experiments should be conducted to confirm this suggestion.

One important extension to this study is to obtain a larger chat data set. Once one is available, an extensive cross method comparison can be performed and other techniques such as Rocchio and linear least squares fit can be included in the study. Also, the impact of combining existing methods, using an ensemble classifier, on the performance can be studied. Another extension is to study the impact of reducing the feature space (vocabulary) using an appropriate reduction on the overall performance of the methods especially the Support Vector Machine. A possible reduction is to drop the the least or the most frequent words. Also, the impact of using different kernel functions in the Support Vector Machine method on its performance can be investigated.

References

[1] "ICQ Inc., the ICQ Internet Chat Service home page." http://www.icq.com, last accessed on April

1st, 2002.

[2] "Tjerk Vonck and mIRC Corporation, the mIRC home page." http://www.mirc.com, last accessed on April 1st, 2002.

[3] K. Aas and L. Eikvil, "Text categorisation: A survey," June 1999.

[4] Y. Yang, "An evaluation of statistical approaches to text categorization," *In Information Retrieval journal*, vol. 1, no. 1-2, pp. 69–90, 1999.

[5] D. D. Lewis and M. Ringuette, "A comparison of two learning algorithms for text categorization," in *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, (Las Vegas, US), pp. 81–93, 1994.

[6] A. Meehan, G. Manes, L. Davis, J. Hale, and S. Shenoi, "Packet sniffing for automated chat room monitoring and evidence preservation," in *Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security*, June 2001.

[7] J. Oikarinen, "Internet Relay Chat RFC." RFC1459, May 1993.

[8] T. Mitchell, *Machine Learning*. McGraw Hill, 1997.

[9] T. Joachims, "Text categorization with Support Vector Machines: Learning with many relevant features," in *Proceedings of ECML-98, 10th European Conference on Machine Learning* (C. Nédellec and C. Rouveirol, eds.), no. 1398, (Chemnitz, DE), pp. 137–142, Springer Verlag, Heidelberg, DE, 1998.

[10] B. Bigi, A. Brun, J. Haton, K. Smaili, and I. Zitouni, "A comparative study of topic identification on newspaper and e-mail," in *Proceedings of the String Processing and Information Retrieval Conference (SPIRE2001)*, 2001.

[11] Y. Yang and X. Liu, "A re-examination of text categorization methods," in *22nd Annual International SIGIR*, (Berkley), pp. 42–49, August 1999.

[12] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*. Massachusetts Institute of Technology, 2001.

[13] A. K. McCallum, "Naive Bayes algorithm for learning to classify text," 1997. http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html, last accessed on April 1st, 2002.

[14] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering." http://www-2.cs.cmu.edu/~mccallum/bow, last accessed on April 1st, 2002, 1996.

[15] V. Cherkassky and F. Mulner, *Learning from Data*. John Wiley, 1998.