# Empirical Evaluations of Automatic Forum Selector

Chen-Huei Chou

***Abstract*—Because of the popularity of the Internet, customer care has been transferred to the Internet-based or Web-based systems. Online discussion forums are common methods used in electronic Customer Relationship Management. However, one emerging problem is that normally companies offer a series of categories of forums to be discussed. Posting messages to incorrect categories may delay the response time and it may take several trials to redirect the messages to the right person in charge or right category for discussion. In this study, we propose the use of text categorization approach to automatically select a target forum category. The empirical evaluations demonstrate the utility of text categorization approach. We also found that decision tree outperformed other machine classifiers.**

***Index Terms*—Text mining, text classification, machine learning, forum selector**

## I. Introduction

Customer care is important to all firms since it would enhance customer satisfaction and loyalty. Traditionally, customers can be taken care of through company's call center. If their questions can not be answered by customer representatives, the questions will be recorded and then dispatched to a person or a team in charge and has the ability to solve the problem. Similarly, customer care through e-mail system operates in a similar manner. The messages will be read and dispatched manually to the right person to be solved. The whole process may take minutes, hours, or days to be delivered to the right person depending on the amount of messages being read by human dispatcher(s).

Because of the popularity of the Internet, customer care has been transferred to the Internet-based or Web-based systems. Online discussion forums are common methods used in eCRM (electronic Customer Relationship Management). Online discussion forums have also been called electronic bulletin boards. "Script-driven forums allow(ing) visitors to share information with others and can help shape a Web site to better serve the customer needs" (Feinberg et al. 2002). Messages can be posted to a specific category by customers and be answered by other customers who have experience or knowledge to deal with the problems, or by customer representatives or employees who are in charge of the problem domain. The messages can be taken care of faster than traditional method such as call center or e-mail systems. However, one emerging problem is that normally companies offer a series of categories of forums to be discussed. A newbie of the forum may not know the most suitable target forum to post and sometimes he/she posts the question in the wrong forum space. Regardless of the waste of forum space, the posts in the wrong forums sometimes are ignored by most readers (visitors) and can not be answered by administrator of that forum category since the administrator may have limited knowledge to the topic supposed to be in other forum categories. As a result, the administrator needs to forward the message to administrators of other forum categories. If the administrator is lucky, the message can be forwarded to the right person in charge in one trial. Otherwise, it will take several trials to redirect the message to the right person or right category for discussion.

Because of the wrong choice of forum category, it may take longer time to reply the messages. Therefore, an automatic technique helping customer select the right forum to ask questions is of great importance. This paper provides an effective way to automatically choose the correct forum category based on the textual content in the message. Specifically, this research attempts to answer the following questions: *Can data mining techniques be used to help consumer automatically choose their target forum to post discussions? What is the suitable attribute size for text classification? Which machine classifier outperforms the others?*

The potential contribution of this study is to provide an effective method helping customers automatically target their forum category. In the study, we empirically evaluate the performance of six machine learning methods used to automatically classify forum messages.

The rest of the paper is organized as follows. First we review the text categorization techniques. We then describe the details of empirical evaluations and discuss the results. Finally, we conclude the paper and outline potential future research directions.

## II. Literature Review

Forum categories are pre-defined and prepared to let customers post their discussions. Customers' messages being posted are textual documents. Based on the content of a message, a relevant forum category is chosen for the post. Text categorization is a technique which can meet this goal to classify those messages, based on their contents, into pre-defined categories. In this section, we review the text categorization techniques.

### *A. Text Categorization*

Text categorization (also called text classification) is to predict a category from a pre-defined set based on the characteristics of natural language texts (Sebastiani 2002). The predicted category is called the *class* (output variable) and the characteristics of natural language texts are the *attributes* (input variables) describing the texts. Text categorization is a supervised learning process, creating a function from training data to associate *attributes* with a

Manuscript received February 28, 2012; revised April 30, 2012.

C. H. Chou is from College of Charleston, SC, USA (e-mail: chouc@cofc.edu).

desired *class*. The training data is a set of textual documents $\{d_1, d_2, ..., d_j\}$ with their corresponding class $\{c_1, c_2, ..., c_i\}$. A machine learning algorithm, a *classifier,* is used to find the function $c_i = f(d_j)$ from correct pairs of $< d_j, c_i >$ consistently, where document $d_j$ is represented as a composite of term weights (explained in the indexing section). Once the classifier is built, a correct prediction is made when a document $d_j$ can be assigned to the correct class $c_i$. Various techniques have been successfully applied in several text categorization problems such as webpage classification (Chen and Hsieh 2006), junk email filtering (Sakkis et al. 2003), and online deception detection (Zhou et al. 2004).

### B. Indexing

Indexing of document $d_j$ is to generate a form of presentation which is informative and representative to the document. The contents of the documents are basically strings of characters. However, documents represented as strings of characters are not suitable for machine learning algorithms (Sebastiani 2002).

Information Retrieval research suggests that word stems (or bag or words) work well as representation units/attributes and the ordering of these units in a document has minor importance (Joachims 1998). Following the bag-of-words indexing approach, a document $d_j$ is described by a vector of term weights $\overrightarrow{d_j} = \left\langle w_{1j}, w_{2j}, ..., w_{|T|j} \right\rangle$, where $w_{ij}$ is the weight of *i*-th term (word stem) in document $d_j$ and $T$ is a set of selected term or attribute. Among term weighting schemes, TF (term frequency) and TF/IDF (term frequency/inverse document frequency) have been found to be effective and are most commonly used in text categorization (Baeza-Yates and Ribeiro-Neto 1999).

### C. Dimensionality Reduction

Dimensionality $|T|$ is the size of term weights in a vector $\overrightarrow{d_j} = \left\langle w_{1j}, w_{2j}, ..., w_{|T|j} \right\rangle$. It will increase when the size of dataset increase. For instance, $|T|$ of a dataset with 2,000 documents is usually larger than $|T|$ of a dataset with 1,000 documents if no duplicated files are included. As a result, when the size of a dataset is large, the dimensionality $|T|$ is also large. When the dimensionality is too large, it slows down the learning methods for classification and overfitting occurs for some methods. Dimensionality reduction can reduce the problem of overfitting (Sebastiani 2002). Overfitting refers to the phenomenon in which a model trained by learning algorithm may well describe the relationship between predictors and outcome in the training data, but may subsequently fail to provide valid predictions in testing data which are not included in the training data.

Dimensionality reduction can be performed in two ways. First, dimension can be reduced by getting rid of stop-words and removing suffixes. The second way is to choose the top rank of attributes by feature selection, using information-theoretic term selection functions.

It is suggested to perform stop-words removal before indexing (Sebastiani 2002). Stop-words are frequent units which do not provide too much meaning. They include articles such as “an”, “the”, prepositions such as “in”, and conjunctions such as “and”, “although”. Moreover, since terms with common stem usually have similar meaning, they can be combined as a term by suffix stripping process. For example, connect, connected, connecting, connection, and connections are the ones with the same stem except different suffixes -ed, -ing, -ion, and -ions. By removing the suffixes, they can be merged into “connect” as the only presentation. Porter stemming algorithm is commonly used to remove suffixes (Porter 1980).

Since the dimension is usually still high after stop-words removal and Porter suffix stripping, further reduction is needed. Information-theoretic feature selection functions can be used to further reduce the dimension. Among feature selection functions, Yang and Pedersen (1997) reported that Information Gain was more effective than Document Frequency, Mutual Information, and Term Strength functions. Also, Information Gain gave the best result for k-Nearest Neighbor classifier when 98% of attributes were removed.

## III. Empirical Evaluation

We conducted an empirical evaluation, comparing six classification techniques. In this section, we describe the dataset we collected and details of experiments.

### A. Data Collection

The goal of the study is to examine the performance of text classification techniques used to automatically assign written messages to a target forum category. Sun Developer Network online forum (http://forum.java.sun.com/index.jspa) is a professional official forum, maintained by Sun Microsystems, Inc., discussing Java programming techniques and issues from various aspects. We collected 25,501 messages from 10 different main categories (e.g. Java Run Time, cryptography, etc.). Each message was collected from one of the 10 categories. Table 1 shows the distribution of messages in the 10 forum categories on Sun Developer Network. All messages were collected from Web-based forum, written in HTML (Hypertext Markup Language). HTML is used to format and display contents in the webpages. It doesn’t provide additional meaning of contents. In this study, we apply text mining techniques to classify forum textual messages, disregarding the HTML format in the messages.

### B. Data Preparation

Before performing pre-processing of text categorization, the uniqueness of the files should be checked. Duplicated messages will increase the weighting of specific terms in the messages and decrease the coverage of other terms in other non-included messages which can be substituted for these duplicated ones. Therefore, pair-wise byte-by-byte comparison on the documents collected in the dataset was performed to ensure the uniqueness of files before dataset pre-processing.

### C. Dataset Pre-Processing

Besides the uniqueness of contents, we further prepared the dataset suitable for machine text classifiers. The pre-processing includes two main processes: indexing and dimensionality reduction. We used TF weighting scheme to represent the forum messages. After removing stop-words and applying Porter stemming algorithm (Porter 1980) for suffix stripping, we found 26,559 unique terms in the dataset. Previous studies suggested that Information Gain is an effective relevance measure to rank the dependency between attributes and predicted class. However, there is no suggested number of attributes to be selected for machine learners. We performed evaluations on six machine learning algorithms based on different sizes of attributes included for model building.

## IV. Evaluation

We developed a document indexing program based on the WekaIndex tool  and used the Weka data mining software (Witten and Frank 2005) for the experiments. Six machine learning methods (Na ïve Bayes, Na ïve Bayes Multinominal, Neural Network, Support Vector Machine, k-Nearest Neighbor, and Decision Tree) were used for performance evaluations. Weka’s default parameters were kept for most algorithms. For learning methods, two hidden layers and 50 training epochs were used for Neural Network. We used a small number of hidden nodes to prevent severe overfitting, as the ratio between the number of training examples and the number of input nodes was low. Also, three neighbors were set for k-Nearest Neighbor method. The decision tree method was J4.8, Weka’s implementation of C4.5. All experiments in the study were conducted in the same environment in which a personal computer equipped with Intel Core i5-2500 3.30GHz CPU and 16 GB RAM and installed Microsoft Windows 7 Ultimate edition with service pack 1. A 10-fold cross-validation method was applied for performance evaluation in terms of accuracy rate.

## V. Results

In this section, we reported our experimental results in answering our research questions. Since Information Gain does not suggest an appropriate size of attributes for classification, we first examine the performance over six learners in using different attribute sizes. Figure 1 shows the performance comparisons of machine learning methods based on different attribute sizes. Although we had over 26,000 attributes, most learning methods performed well when using a small portion of attributes ranked by Information Gain. Except for Support Vector Machine, other methods reached their peak performance when a small number of attributes were included for classification. Support Vector Machine consistently improved its performance when more top ranked attributes were added. However, Na ïve Bayes, Na ïve Bayes Multinominal, Neural Network, k-Nearest Neighbor, and Decision Tree reached their peak performance when 42, 39, 36, 28, and 68 top ranked attributes were used, respectively.

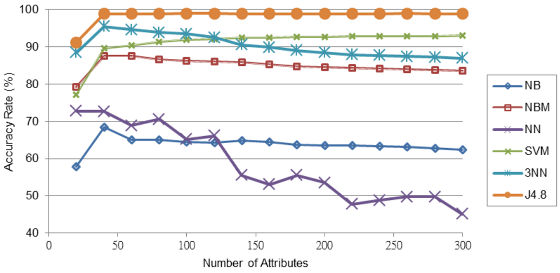

Fig. 1. Performance comparisons of machine learning methods based on different attribute sizes

TABLE I: Distribution of Messages in Forum Categories

| Categories | Number of Messages |
|---|---|
| Native Methods | 4,407 |
| Serialization | 1,145 |
| Collections: Lists, Sets, and Maps | 1,690 |
| Internationalization | 1,903 |
| Java Virtual Machine (JVM) | 4,472 |
| Java Runtime Environment (JRE) | 3,792 |
| Cryptography | 3,251 |
| Java Secure Socket Extension | 1,567 |
| Signed Applets | 996 |
| Security Applications, APIs, and Issues | 2,278 |

TABLE II: Mean Accuracy Rate of 20 Runs of 10-fold Cross-Validation Evaluations

| Machine Learning Methods | Accuracy Rate (%) |
|---|---|
| Na ïve Bayes | 67.99 |
| Na ïve Bayes Multinomial | 87.76 |
| Neural Network | 72.60 |
| Support Vector Machine | 89.84 |
| 3-Nearest Neighbor | 95.46 |
| Decision Tree | 98.94 |

Since the machine learners had their peak performance when different attribute sizes were used, we used their average size of 43 attributes in order to further compare performance of different learners. Table 2 shows the results of mean performance of 20 runs of 10-fold cross-validation evaluations over the six machine learning methods. Na ïve Bayes gave the poorest result, compared to others. It is a decent learner when making binary decisions with accuracy rate over 88% (Chou et al. 2010). However, it gave around 68% of accuracy rate classifying messages into 10 categories. Also, Neural Network overfitted badly when more attributes were included. Decision tree outperformed others and k-Nearest Neighbor toped at the second place. Based on Analysis of Variance (ANOVA) analysis, we found that the performance of the six machine learning methods was significantly different at 0.001 level.

## VI. Conclusions and Future Directions

The results show that text classification techniques can be used as a tool to automatically and effectively classify forum messages. Different machine learners had peak performance when different attribute sizes were used. In our experiments, the peak performance happened when around 40 (out of 26,559) top ranked attributes were used. Roughly 0.16% of attribute space is enough for machine learners to effectively classify forum messages. Compared to previous study (Yang and Pedersen 1997) using 2% of attributes, our study suggested far less percentage of attributes being used for effective text classification. Based on our comparison on machine learners' performance evaluations, decision tree outperformed others and k-Nearest Neighbor toped at the second place. In addition, we also found that the performance of the six machine learning methods was significantly different (at 0.001 level). In summary, text classification technique can definitely supplement the process of manually looking for the appropriate forum target category and would potentially enhance customer satisfaction and loyalty. Moreover, because of the use of content-based classification techniques, the organizations can save more labor work without manually dispatching new messages once the classifier is trained.

This work opens up several areas for future research. Forum categories can be sometimes further grouped into another upper level of categories. For instance, Java Virtual Machine and Java Runtime Environment can further grouped into Runtime Environment. Hierarchical classification techniques can be adopted to evaluate the performance. It would be interesting to compare the effectiveness of hierarchical classification techniques with additional levels of categories over regular classification methods used in this study. Finally, more and more companies are collecting customers' feedback and comments over their official websites and e-mail systems. In order to reduce the turnaround time, text classification approach can be used to dispatch the messages in a similar fashion.

## References

[1] R. B. Yates and B. R. Neto, *Modern information* retrieval, Addison-Wesley, 1999.

[2] R. C. Chen and C. H. Hsieh, "Web page classification based on a support vector machine using a weighted vote schema," *Expert Systems with Applications* vol.31, no.2, pp. 427-435. 2006.

[3] C. H. Chou, A. P. Sinha, and H. Zhao, "A hybrid attributes selection approach for text classification," *Journal of the Association for Information Systems* vol.11, no.9, pp. 491-518. 2010.

[4] R. A. Feinberg, R. Kadam, L. H. Hokama, and I. Kim, "The state of electronic customer relationship management in retailing," *International Journal of Retail & Distribution Management*, vol. 30, no. 10, pp. 470-481. 2002.

[5] T. Joachims, "Text categorization with support vector machine: learning with many relevant features," in *Proc. of the 10th* European *Conference on Machine* Learning, 1998, pp. 137-142.

[6] M. F. Porter, "An algorithm for suffix stripping," *Program.* vol. 14, no. 3, pp. 130-137. 1980.

[7] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and P. Stamatopoulos, "A memory-based approach to anti-spam filtering for mailing lists," *Information Retrieval* vol. 6, no. 1, pp. 49-73. 2003.

[8] F. Sebastiani. Machine learning in automated text categorization. *ACM* Computing *Survey*, vol. 34, no. 1, pp.1-47. 2002**.**

[9] Y. Yang, and J.O. Pedersen. A comparative study on feature selection in text categorization. *Proc. of 14th International Conference on Machine Learning*, pp. 412-420. 1997.

[10] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques. 2$^{nd}$ ed*, Morgan Kaufmann, 2005.

[11] L. Zhou, J. K. Burgoon, D. Twitchell, and T. Qin, "A comparison of classification methods for predicting deception in computer-mediated communication," *Journal of Management Information Systems* vol. 20, no. 4, pp. 139-166. 2004.