http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 6 (4): 567-572, 2007
ISSN 1812-5638

Automated Classification of Customer Emails via Association Rule Mining

[1]Senthamarai Kannan Subramanian and [2]N. Ramaraj
[1]Department of Information Technology, Thiagarajar College of Engineering
Madurai, India
[2]G.K.M. Engineering College, Chennai, India

Abstract: With the increase in growth of Email Communication, it is necessary to organize the information for faster and easier processing. Usually, the companies receive huge number of emails to the single email address. Customer service center of an organization is one such example. Usually this kind of emails should be responded within a certain amount of time by a responsible person. In order to meet this challenge, automatic categorization of all incoming emails must be extremely useful because it can help route an email to the right person. The focus of the research is to find out whether this classification can be used to automatically route incoming emails to the person in charge using an association rule mining methodology and apriori algorithm. One major advantage of the association rule based classifier is that it does not assume that terms are independent and its training is relatively fast. Furthermore, the rules are human understandable and easy to be maintained or pruned by human being.

Key words: Association rule mining, text classification, customer relationship management, contact center automation, email mining

INTRODUCTION

The explosive growth of on-line communication, in particular e-mail communication, makes it necessary to organize the information for faster and easier processing and searching.

Usually, the companies receive huge number of emails to the Single email address. Customer service center of an organization is one such example. Usually the responsible person should respond this kind of emails within a certain amount of time. In order to meet this challenge, automatic categorization of all incoming emails will be extremely useful because it can help route an email to the right person.

The idea is to find out whether this classification can be used to automatically route incoming emails to the person in charge.

Text classification is the supervised learning task of assigning natural language text documents to one or more predefined classes (also called categories or topics) according to their content. The word supervised in the definition means that all the data in a training set is assigned a category before the training process starts.

Classifying messages into different topics introduces a structure into the information, which alleviates information saturation that occurs when too many messages are received. Separating messages into several groups by topic can also help a user to prioritize the e-mail or suggest further actions. Information inundation may cause information entropy, when incoming messages are not sufficiently organized by topic or content to be easily recognized as important. Classification can also facilitate the search for a particular message because a user can limit his search to only several topics.

In this study, three main categories for classification are considered, which includes enquiry about the company products, prices, delivery, etc., purchase orders, feedback about the company products. The association based classification technique is used to classify the

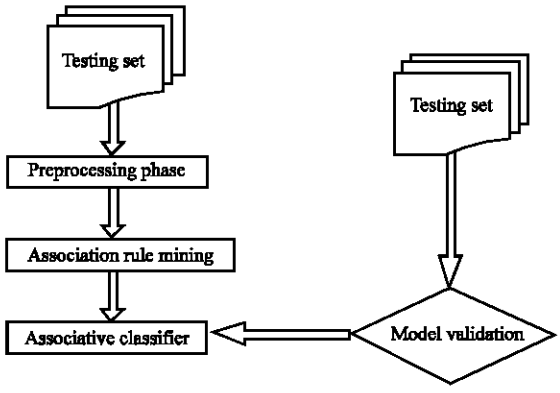

Fig. 1: Text categorization flow chart

Corresponding Author: Senthamarai Kannan Subramanian, Department of Information Technology, Thiagarajar College of Engineering, Madurai, India

incoming emails. Two main phases in this paper are Building the associative classifier and testing the associative classifier (Fig. 1).

This is the new approach to apply associative classification techniques to the task of classifying email messages.

THE PROPOSED WORK

The proposed method is implemented using the Java language. In implementation, there are three parts: Email Preprocessing, Building the associative classifier and validation.

Initially, Email preprocessing should be done before the email is given as input to any of the two phases. Preprocessing steps include stop word removal, stemming. After preprocessing is done on the email sets, resulting data should be converted into the format that can be inputted for generating rules. In the second phase, incoming emails are classified by comparing with the generated rules and assign it to the matching category.

Email preprocessing involves the process of transforming the training dataset into a representation suitable for the Apriori algorithm. This stage extracts the informational words from the data set. It consists of the following two steps: (1) Removal of non-discriminative words and (2) Suffix stripping.

Removal of non-discriminative words: In emails certain words are most frequent and are not discriminative of a message contents, such as prepositions, pronouns and conjunctions. Examples of such words are we, that and this, etc. Some widely used conversational English words, such as \I'm, \isn't, \can't, etc. are of less importance elimination of these terms is performed in this step.

Suffix stripping: This is a process for removing the commoner morphological and inflexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems. The Porter stemming algorithm (or Porter stemmer) is used for doing this process. Ignoring the issue of precisely where the words originate, we can say that a document is represented by a vector of words, or terms. Terms with a common stem will usually have similar meanings.

Frequently, the performance of an IR system will be improved if term groups such as this are conflated into a single term. This may be done by removal of the various suffixes -ED, -ING, -ION, IONS to leave the single term assassinate. In addition, the suffix stripping process will reduce the total number of terms in the IR system and hence reduce the size and complexity of the data in the system, which is always advantageous. Hence those words which are extracted from the previous steps are suffix stripped to increase the efficiency. Unfortunately emails are usually very noisy and simply applying text-mining tools to them, which are usually not designed for mining from noisy data, may not bring good results.

In this study, we formalize the email-cleaning problem as that of non-text data filtering and text data normalization (Tang *et al*., 2005). By filtering of an email we mean a process of removing the parts in the email which are not needed for text mining and by normalization of an email we mean a process of converting the parts necessary for text mining into texts in canonical form like a newspaper style text.

Header, signature, quotation (in forwarded message or replied message), program code and table are usually irrelevant for mining and thus should be identified and removed (in a particular text mining application, however, we can retain some of them when necessary).

A RECENT SURVEY ON EMAIL CLASSIFICATION

An interesting paper (Aas and Eikvil, 1999) dealing with text classification compares SVM to four other common methods, namely Naive Bayes, Rocchio, kNN and C4.5. Features are selected using the information gain criterion which measures the amount of information obtained by knowing the presence or absence of a word and ranks all words accordingly. The report of the experiments shows that SVM shows the best results. Of the other methods, kNN outperforms the decision tree method and the Rocchio algorithm; Naïve Bayes performs worse. Another (Mccallum and Nigam, 1998) investigated how document classification can benefit from word clustering, i.e., forming feature groups by joining similar words. It uses the Bayesian classifier and bases their experiments on the real-world corpora, namely the Yahoo! Data set (web pages). Most relevant to the proposed work, however, are experiments conducted on email corpora. What is special about emails compared to other text documents is that they are typically short and contain a limited amount of structure due to the division into a body and several headers. Several authors (Aas and Eikvil, 1999; Yang and Lie, 1999) have investigated email classification; many of them, however, concentrate on binary classification, where emails are classified under either one of two categories, e.g., spam filtering. Using emails grouped into three folders, they form three two-class problems by defining the emails from each group in turn as positive examples, the emails from the other two

groups as negative ones. To build the feature list, stop words and words appearing in all the emails are removed, the remaining words are stemmed.

In this experiment, the Bayesian classifier outperforms RIPPER, achieving 80% accuracy after A similar tool, Mail Cat (Segal and Kephart, 1999) constructs an email classifier by analyzing the user's existing folders and then uses this classifier to predict the three most likely destination folders. It is thus designed as a categorization aid for users with many folders rather than a tool that categorizes an email directly into a specific folder. The algorithm used is similarity-based and classifies a new message by computing the similarity between the message's word-frequency vector and the weighted folder vectors. Summarizing the results reported in the papers on email classification tasks, the accuracy achieved is usually between 70 and 80%. However, these results are hardly comparable to the results expected for the task at hand, due to the differences in amount of training data, number of categories, email contents and type of categorization.

ASSOCIATION RULE MINING FOR TEXT CLASSIFICATION

Association Rule mining (Han and Kamber, 2002) searches for interesting association or correlation relationships among items in a given large data set. we model email messages as transaction where items are words or phrases from the email. After preprocessing a email message, by eliminating stop words and stemming, emails are represented by sets of cleaned words di = {t1,... tn} as well as category to which they belong cj. Thus, each email in the training set belonging to a category C and with n terms would be represented by a transaction {t1,..., tn, C}. The Apriori algorithm (Agrawal *et al*., 1993) is used for mining frequent item-sets in transactional databases to find frequent sets of words in the emails of the training set. Given the frequent sets of words and topical category assigned to the transaction from which they were extracted association rules are deduced with constraints on the antecedent and consequent of the rules such that the antecedent always contains words while the consequent is exclusively a topical category. Suppose and confidence is the two measures that are used in association rule mining. Support can be defined as fraction of transaction that contains both X and Y. Confidence measure how often items in Y appear in transaction that contain X.

Similar to the approach presented in (Antonie and Zaiane, 2003; Zaiane and Antoni, 2002), we model text documents as transactions where items are words or phrases from the document. After pre-processing a text document, by eliminating stop words (i.e., terms that are too frequent in the data collection and insignificant) and stemming (i.e., transforming words in their canonical form), documents are represented by sets of cleansed words di = {t1, tn} as well as the category to which they belong Cj. Thus, each document in the training set belonging to a category C and with n terms would be represented by a transaction {t1,.., tn, C}. The apriori algorithm is used for mining frequent item-sets in transactional databases to find frequent sets of words in the documents of the training set.

APPLICATION OF APRIORI ALGORITHM ON EMAILS

Apriori (Agrawal and Srikant, 1994) is an influential algorithm for mining frequent item sets for Boolean association rules. It employs an iterative approach where k frequent item sets are used to explore (k+1) frequent item sets. First, the set of frequent 1-itemsets is found. This set is denoted as L1. L1 is used to find L2. The set of frequent 2-itemsets is used to find l3 and so on until no more frequent k-item sets can be found. The finding of each Lk requires full scan of the database. The apriori property used is, All nonempty subsets of a frequent item set must also be frequent. It belongs to a special category of properties called anti-monotone property. It states, If a set cannot pass a test, all of its supersets will fail the same test as well.

Figure 2 is the representation of our proposed system. In this study, the modified version of apriori algorithm is used for classification. After the frequent item sets are generated by the algorithm, rules will be generated by mapping the frequent item set to the corresponding category (Table 1).

Preprocessed data should be converted into binary data format before mining association rules. Datasets that is being preprocessed is represented in the form of the

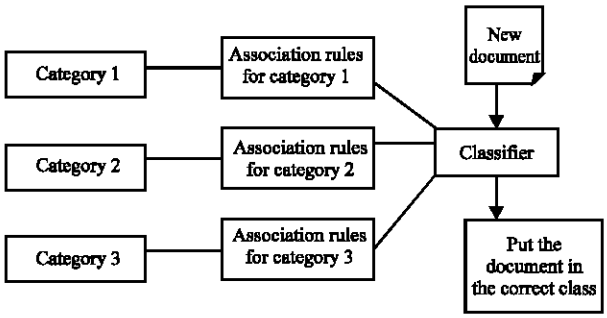

Fig. 2: Architecture of associative classifier

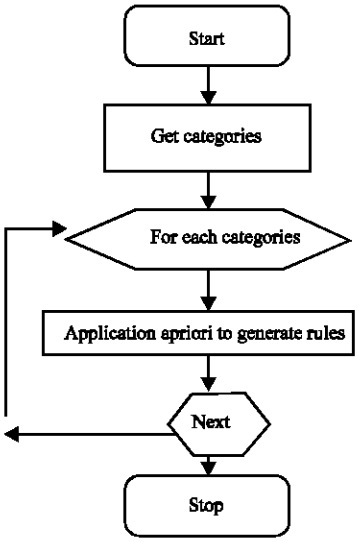

Fig. 3: A Binary form of training email set

Table 1: A binary form of training email set

	I1	I2	I3	I4	….
E1	1	1	0	0	0
E2	0	1	1	0	1
E3	0	0	0	1	1
….	1	1	1	0	0

table where row correspond to email and column represent the words in the email. If an email contains the word in one of the column, corresponding cell will be marked with 1, otherwise 0. Table will be formed for each category and association rules will be generated separately for each category (Table 1).

The following is the various items and their corresponding related themes as indicated by the Emails.

I1-> prize/sent/recommend/others
I2-> discount/product/problem/others
I3-> delivery/quotation/defect/others
I4-> guarantee/balance/satisfy/others

The output of apriori algorithm will be a set of rules for each category. These rules will be later used in testing stage (Fig. 4). Sample classification rules generated using Apriori algorithm

I1 ^ 13 ^ I7 ^ I10 ➔C1
I2 ^ 19 ^ I12 ➔C3
I1 ^ 12 ^ I13 ^ 14 ➔C2

TESTING STAGE

In the testing stage, rules generated in the training stage must be used to classify the incoming email. Initially the mails are to be extracted from the company email address together with the mail header and body. Extracted

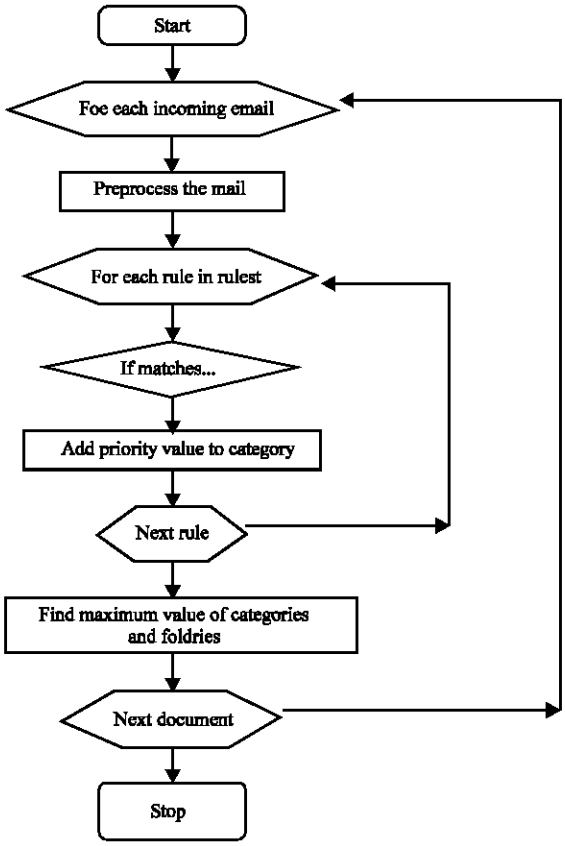

Fig. 4: Flow chart for association rule generation

emails should be preprocessed before comparing with the generated rules (Fig. 4).

Processing such as lexical analysis, stop list word removal and stemming should be applied to the extracted data. Resulting emails should be compared with each rule and the email is categorized to the most exactly matching category. If the left part of the rule matches with the email, count the category of the rule of the document. For this purpose, a priority value for each generated rule is assigned.

Higher priority value will be assigned to large frequent item sets and vice versa. Since the priority is assigned manually, we can increase the classification accuracy. If a rule matches with the email, corresponding priority value will be added to the category.

Finally, the category with the highest value will be classification for that email. After an email has been classified to the particular category, email will be forwarded to the respective folders in the system.

RESULTS

Initially for this study, 800 email sets are used for each category to train the system. These email sets were preprocessed and used for generating the rules using

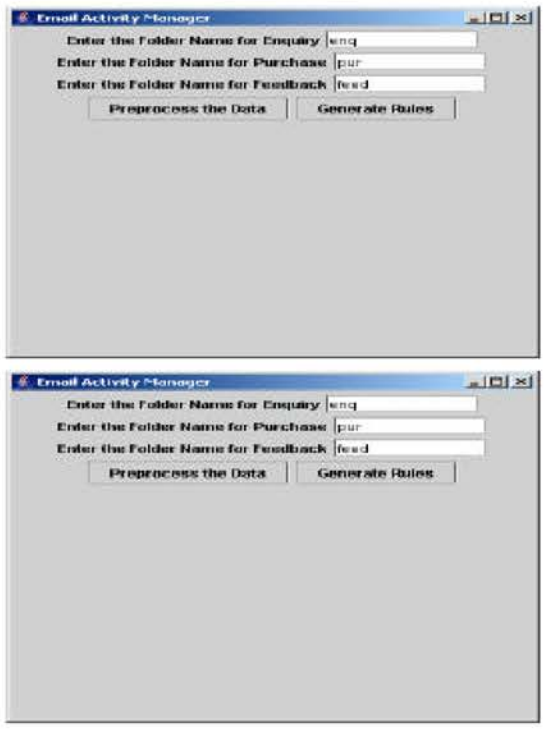

Fig. 5: Folder specification for training

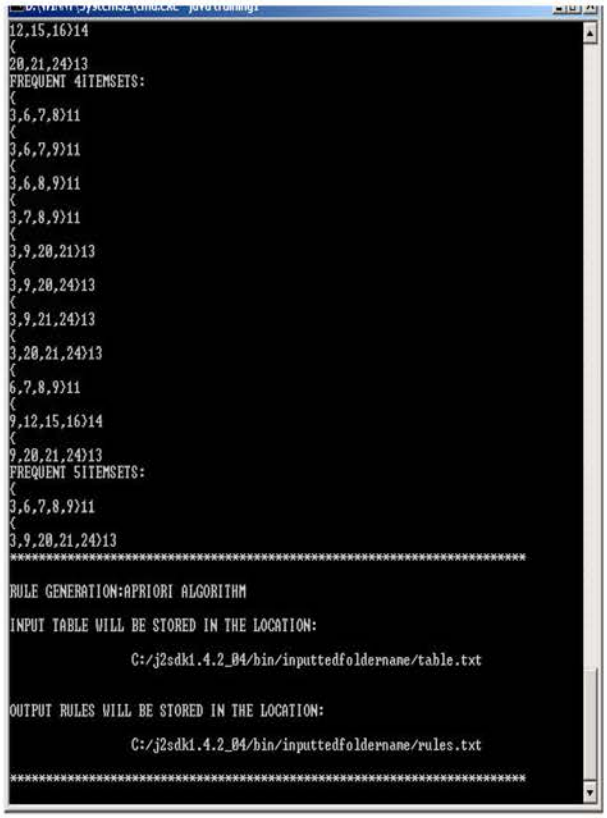

Fig. 6: A sample output of apriori algorithm

Fig. 7: Testing stage interfacex

Table 2: Accuracy table

Category	No. of emails for testing stage	No. of correctly classified emails	No. of miss-classified emails	Accuracy (%)
Enquiry	200	180	20	90
Purchase	200	180	20	90
Feedback	200	180	20	90

apriori algorithm. In testing stage, mails are extracted using Java Mail API and those mails were used to test the system.

In the Fig. 5 the page that will be displayed. The user can select any one of the following options. This is the training stage. The user has to specify the corresponding folders that have to be trained.

Figure 6 displayed shows the list of rules that are generated using the apriori algorithm.

Figure 7 shows testing stage. The user has to specify the host name, email-id and the password for testing the mails.

Table 2 displays the email set that was used for testing training stage. For the purpose of calculating the accuracy, the data set is divided into two data sets: 80% (2400 emails) were used for training; the remaining 20% (600 emails) were used as testing data. As the test data sets should reflect the class distribution in the training data, it contains 200 emails of the enquiry category, 200 emails of the purchase category and 200 emails of the feedback category. The Table 2 shows the range of accuracy for each category.

Accuracy mainly depends on the training set used. Accuracy can be defined by percentage of emails that are correctly classified. It is very sensitive to small changes in the training data.

CONCLUSIONS

The emails are automatically extracted using mail extractor and folderised, so that the corresponding person will handle the emails. The proposed methodology has proved to reduce the burden of the customer service manager. Faster services will be provided due to the earliest reply of the incoming mails.

In this study, association rule mining algorithm is used for email classification. The proposed methodology

provides evidence that association rule could be used in automatic text categorization efficiently and effectively. One major advantage of the association rule based classifier is that it does not assume that terms are independent and its training is relatively fast. Furthermore, the rules are human understandable and easy to be maintained or pruned by human being.

FUTURE WORK

We are working to improve the classifier by using algorithms that are based on confidence measure.

Moreover Automatic reply to the incoming mails that are categorized is also being carried out along with enabling this application to work in Mobile Environment.

REFERENCES

Aas, K. and A. Eikvil, 1999. Text Categorization: A survey. Technical report, Norwegian Computing Center, June.

Agrawal, R., T. Imielinski and A. Swami, 1993. Mining association rules between sets of items in large databases., Proceedings of 1993 ACM-SIGMOD International Conference on Management of Data, Washington, DC., pp: 207-216.

Agrawal, R. Srikant, 1994. Fast algorithm for mining association rules, Proc. VLDB Conference., Santiago, Chile, pp: 487-499.

Antonie, M. and Osmar R. Zaïane, 2002. Text Document Categorization by Term Association. IEEE International Conference on Data Mining (ICDM'2002), Maebashi City, Japan, December 9-12, pp: 19-26.

Han, J. and M. Kamber, 2001. Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers.

Jie Tang, Hang Li, Yunbo Cao and Zhaohui Tang, 2005. Email Data Cleaning, KDD'05,Chicago, USA.

Mccallum, A. and K. Nigam, 1998. A comparison of event models for Naïve Bayes text classification, AAAI WS Methodology of Applying Machine Learning.

Segal, R.B. and Jeffrey O. Kephart, 1999. Mail Cat: An intelligent assistant for organizing email. Third International Conference on Autonomous Agent.

Yang, Y. and X. Liu, 1999. A re-examination of text categorization methods. International ACM-SIGIR Conference on Research and Development in Information Retrieval.

Zaïane, O.R. and M. Antonie, 2002. Classifying text documents by associating terms with text categories. Proceedings of the thirteenth Australasian Database Conference (ADC'02), Melbourne, Australia, January 28.