Category-Based Message Routing System

Yi-Chun Chen
SimpleAct Incorporated
Room 801, 8F, No.54, Section 4, Min Sheng E. Rd.,
Taipei 105
Taiwan
ken@simpleact.com.tw

Abstract

This paper describes a message routing system which is designed for a customer service center. The system can classify each incoming email into predefined categories and then route it to a suitable customer service representative automatically. There are two optional built-in modules for email classification in this system. One is a template-based module which the system administrator can create templates manually for analyzing mail texts. The other is a statistical module that extracts features as indices of each category from training data. Each category can be assigned to one or more agents processing the incoming message categorized.

1 Introduction

While more and more companies provide the email channel for customer services, the problem that the companies have more emails than the service departments can handle occurs. For efficiently processing these massive emails, a message routing system is necessary for classifying the emails and routing them to suitable agents. An intuitive method is the keyword-based approach that is to route emails based on a keyword (eGain, 2002). For example, two keywords, "hardware" and "software", are separately assigned to category A and category B. George is a agent processing the messages in category A and Mary is another agent processing the messages in category B. When a message containing the word, "software", is received, the message will be routed to Mary.

The keyword-based approach is easy-to-implement but the keywords manually set up are not deep enough to analyzing context of messages. For example, a message contains "bring the household transcript to apply the ID card" and another message contains "bring the ID card to apply the household transcript". These two messages both contain the same words but different meanings in context. Another approach is to employ traditional natural language understanding methods, that is, a parser is developed to parse sentences and further analyzes discourses based on the integration of syntactic and semantic information (Allen, 1995). But to analyze the syntactic structure of sentences may cause the problem of ambiguity and besides to build a domain lexicon or a knowledge base is very labor-intensive and time-consuming.

To recover the insufficiency of keywords and reduce the work of building a lexicon manually, we developed two classification module embedded in the system. One is a template-based module which employs templates consisting of sentence-level expressions to identify the concepts expressing in the context of a message. The system administrator can create the templates for analyzing mail texts. The other is a statistical module that extracts features as indices of each category from training data.

2 System Architecture

The message routing system consists of the following components: email broker module, email processing module, classification module, message

dispatch module, email delivering module and database for storing emails, indices of categories, answers and customer service representative (CSR) information.

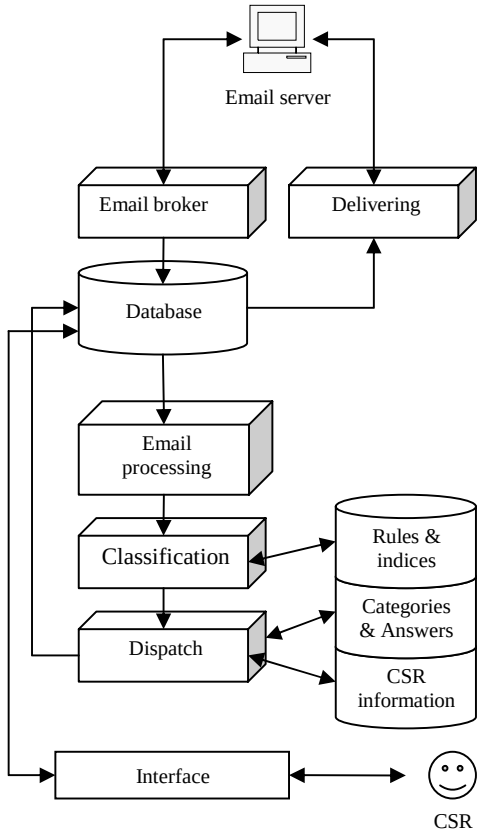

Figure 1. Message Routing System Architecture

As shown in Figure 1, the email broker retrieves emails on the email server with the account and password specified and stores them into the database. Then the email processing module extracts the text of each email including the subject and content. The classification module classifies each email by analyzing the text extracted by the email processing module according to the templates or indices. The classification templates and indices referring to the categories are associated with the data of the categories, answers and CSR information in the database. The dispatch module route each email to a suitable CSR according to the category assigned and some other criteria. The CSR can use the interface to reply emails stored in the database and the system also lists the possible answers for the CSR to choose. Then the delivering module will send the emails to customers on a regular time schedule.

There are two routing types in our system: category-based and skill-based. In the first type, agents are assigned to answer categories and routing is performed automatically on the basis of the category and employs routing criteria. Therefore if an incoming message is classified to a category, any agent assigned to the category could deal with it. In the second type, routing is performed on the basis of the answer and the agent's skill. An agent and an answer can be assigned one or several skills. If the answer of an incoming message is obtained and the skills assigned to the answer are equal to those an agent possesses, the agent could deal with the message. The two routing types employ the following routing criteria to select a suitable agent to process the message: forwarding, load balance, and push/pull as shown in Table 1.

Criteria	Description
Forwarding	The incoming messages are automatically routed to a group or a specific agent when following cases occur: some specified sender mail address, no answer matched, illegal attached files, and *etc*.
Load Balance	The incoming message goes to the agent who has the least workload.
Push/Pull	Automatically reroute messages when an agent logout, on vacation, or otherwise unavailable.

Table 1. Routing Criteria

3 Classification Module

There are two optional built-in modules for message classification in our system. One is template-based module which the system administrator can create template s manually for analyzing mail texts. The other is statistical module that extracts features as indices of each category from training data.

3.1 Template-Based Module

We propose a template-based module to classify messages by employing templates consisting of sentence-level expressions. There are two kinds of the expressions described below:
Definition *S1*:

A sentence-level expression *S1* is characterized as:
S1 = <T_1, O_1, T_2, O_2, T_3,…,T_{n-1}, O_{n-1}, T_n> where
For each T_k in *S*, k =1 to n, T_k is a string.
For each O_k in *S*, k =1 to n-1, O_k is an operator, "And" or "Or" representing the relation between T_k and T_{k+1} where the precedence is "And" > "Or".

Definition *S2*:
A sentence-level expression *S2* is characterized by a 3-tuple:
S2 = <*X*, *Rm*, *Y*> where
X is a string.
Y is a string.
Rm is an operator and *m* is an integer.

The expression *S1* is used for comparing each sentence in a message and the expression *S2* is used for comparing the content of a message with a text window which the initial string is *X* and the size of window is *m*. A template consists of one or several expressions *S1* or *S2*. For example in (1), we can create a template like <雷射印表機, And, 單色> and <推出, R10, 彩色雷射印表機> to compare with the text in the example. The former expression denotes a sentence containing two substrings, "雷射印表機" and "單色", in order. The later expression denotes a string of 10 characters appearing in text which contains two substrings, "推出" and "彩色雷射印表機", and the initial substring is "推出".

Example (1):
科技不斷進步，台灣雷射印表機過去主要以單色爲主，在廠商推出彩色雷射印表機之後，也邁入彩色時代。

(En.) Technology progresses continuously. Laser printers are most monochrome in Taiwan before. After some companies introduce color laser printers, (Taiwan) entered into a period of color.

3.2 Statistical Module

In Addition to the template-based classification module, we also developed a statistical module for message categorization. The difference between the statistical module and the template-based module is that training samples are essential for extracting the features as indices of categories. In this module, the tasks of message classification consist of feature extraction, feature selection and text classification.

Due to the nature of Chinese language, there is no blank inserted to separate words from each other. In the task of feature extraction, we employ Bi-gram model to extract features from training samples that belong to each category (Yang *et al*., 1993; Chen, 2001). The process starts from the first character of the sentence and combines 2 consecutive characters to form a bi-gram. Then it goes on to the second and repeats the grouping further on until the end. Consequently, all possible overlapping bi-grams are obtained. For example, the features extracted from the first sentence of example (1), 科技不斷進步, are: [科技, 技不, 不斷, 斷進, 進步].

In the task of feature selection, we first compute the weight of each feature by a frequency-based scheme, such as TF-IDF (Term Frequency and Inverse Document Frequency) (Joachims, 1997; Tokunaga and Iwayama, 1994; Tsay and Wang, 2000). Then a threshold value is set for filtering off the features having less weight than the threshold value. After the task of feature selection finished, the features of each category are taken as indices storing into the database for message classification.

In the task of text classification, we employ a method similar to *k*-NN algorithm (Yang *et al*., 2002; Ko and Seo, 2002). First the text part of an incoming message is processed into a bi-gram sequence. Then the classification module classifies the message by comparing the bi-gram sequence with the features of each category. The score of the message corresponding to each category is counted by the weights of matched features. After the scores are obtained, the message is assigned to the category having the highest score.

4 Experiment

For evaluate our system performance objectively, we describe a test and result supplied by one of our customers, an ISP company. The company collects hundreds of real emails for training and testing the system.

In the test, 50 categories are predefined. There are 10 samples for each category used to training the system and 2 test samples for each category are used to testing (total 500 training samples and 100

test samples). The system is configured to output the most possible three categories and the right category must appear in the three categories. The result shows that 95% test samples are computed their categories correctly.

5 Conclusion

We have described our message routing system in the previous sections. At present we have already had customers using the system such as ISP (Internet Service Provider) companies and some of them used the system over one year. In our experience, the performance of employing the techniques of natural language processing in the message routing system is promising to some extent; however, there are still some problems need further investigation.

One of the problems is the category's taxonomy. In our system, we provide an interface for administrator to build the taxonomy tree of answers. The administrator may create some category nodes in the taxonomy tree having similar concept. This problem will affect the precision of classification. For example, a software company may have several versions of a product. When the administrator build the taxonomy tree for trouble shooting, one question will have several answers because of the software's version. In this situation, we usually suggest that the administrator can create one category node for the same questions and further create several answers in this category for a CSR to choose the proper one.

Another problem is the quantity and quality of train samples. The training samples are generally taken from the incoming messages of customers. Some categories do not have enough training samples because customers ask those questions less frequently. Besides, the incoming messages always contain some unnecessary information for classification, such as the user's ID, telephone number, and address. In our system, the administrator can set a minimum number of training samples. If the number of training samples of a category is less than the minimum, these training samples will not be processed. We also provide a interface for the administrator to refine the training samples.

In a customer service center, in addition to email, customers may use other channels to ask questions or services, such as telephone and fax. For integrating these channels into our system, other techniques will be involved, such as OCR (Optical Character Recognition), ASR (Automated Speech Recognition) and TTS (Text to Speech). We will further investigate the workable ways to enhance our system in the future.

References

James Allen. *Natural Language Understanding 2nd ed*. The Benjamin/Cummings Publishing Company, Inc., 1995.

Hongbiao Chen. 2001. *Looking for Better Chinese Indexes: A Corpus-based Approach to Base NP Detection and Indexing*, Ph.D. thesis, Guangdong University of Foreign Studies.

eGain. 2002. *The Foundation for a Successful Email Management System*. White Paper.

Thorsten Joachims. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In Douglas H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 143-151, Nashville, US. Morgan Kaufmann Publishers, San Francisco, US.

Youngjoong Ko and Jungyun Seo. 2002. Text Categorization using Feature Projections. *Proceedings of COLING-2002, the 19th International Conference on Computational Linguistics*.

T. Tokunaga and M. Iwayama. *Text categorization based on weighted inverse document frequency*. Technical Report 94 TR0001, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, 1994.

Jyh-Jong Tsay and Jing-Doo Wang. 2000. Design and Evaluation of Approaches to Automatic Chinese Text Categorization. *Computational Linguistics and Chinese Language Processing (CLCLP)*, 5(2): 43-58.

Yang Y., Slattery S., and Ghani R. 2002. A study of approaches to hypertext categorization, *Journal of Intelligent Information Systems, Volume 18, Number 2*.

Yun-Yen Yang, Keh-Jiann Chen, Ching-Chun Hsieh, and Shu-Mei Chen. 1993. A Study of Document Auto-Classification in Mandarin Chinese. In *Proceedings of ROCLING VI*, Hsinchu, Taiwan.