Application of Text Mining into Auto Answering System and Improvement of Mining Performance

Koudai Aman, Fukuya Ishino
Graduate School of Information, Production System Waseda University,
Ishino Laboratory. 2-7 Hibikino, Wakamatsuku, Kitakyushu-city,
Fukuoka, 808-0135 JAPAN
Kodai-aman@asagi.waseda.jp
Ishino@waseda.jp

Abstract. Most CRM systems include the text base response function through the Web, which apply the text mining technology. However, there is the critical problem of bad performance of the mining system; low hit rate of expected answers at the beginning stage. The problem is caused by limited knowledge in the system due to the lack of corpus and documents accumulated. Another cause is that the vocabulary is sometimes poor in the customer's short questionnaire. The main purpose of this study is to improve the performance of mining systems by tuning from the user's standpoint, not from the system provider. We experimented with a mining system. We populated corpus to the system and put some questions into the system repeatedly while changing corpus quantity and the effect of keywords. The results suggest that when the corpus quantity is not large enough, the system can be improved by repeating to input the same corpus several times.

1 Introduction

Today's CRM systems have the auto response function (**Fig. 1**) which deal with customer's questions or complaint sentences online instead of call center operators, in order to reduce the human resource costs. Another merit of the system is that they automatically keep records of all customers' questions and complaints making it easier for management to supervise the quality of service and to plan new products. Fig. 1 shows the automatic answering system [1, 2].

Please use the following format when citing this chapter:

Aman, K., Ishino, F., 2006, in IFIP International Federation for Information Processing, Volume 226, Project E-Society: Building Bricks, eds. R. Suomi, Cabral, R., Hampe, J. Felix, Heikkilä, A., Järveläinen, J., Koskivaara, E., (Boston: Springer), pp. 166–175.

A customer asks a question or makes a complaint to the company through its homepage on the web (1). Next, the system selects some candidate Q&As that seem to be similar to the customer's question from a Q&A database of previously asked questions and its answer. The system presents the Q&A candidates to the customer. At this time, if the customer finds a Q&A candidate which matches his or her intention, he/she selects that Q&A candidate (2). If he/she can not find a suitable Q&A candidate in the list, the company's experts respond to the customer's question afterwards (3). The new question and its answer are added to the Q&A database (4). Moreover, the company may analyze this Q&A data by using text mining technology for management purposes. Text mining technology is applied to the system' search function. However, there is the critical problem of bad performance of the mining system. The problem is caused by limited knowledge in the system due to the lack of corpus and documents. Another cause is that the vocabulary is sometimes poor in the customer's question. Moreover, it is not easy for nonprofessionals to operate for the good performance.

The main purpose of this study is to improve the performance of mining systems from the user's standpoint not from the system provider and to propose how to tune the mining system for the nonprofessionals. We experimented with off-the-shelf text mining system. We populated documents corpus to the system and put some questions into the system repeatedly while changing document quantity and the effect of keywords. We measured ranks of target documents. The higher the ranks of the target documents are, the better the system performance is. And put some questions into the system repeatedly while changing document quantity and the effect of keywords. We measured ranks of target documents. The higher the ranks of the target documents are, the better the system performance is.

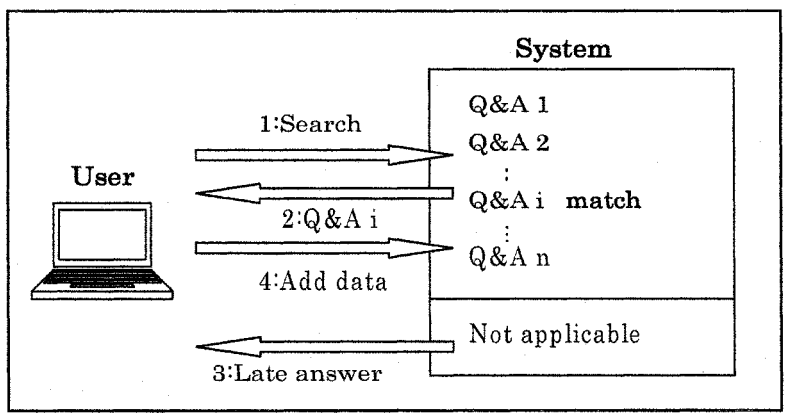

Fig. 1. Auto response function used in most CRM

2 Ming system learning algorithm

The mining system has a concept base. Concept vectors are semantic word expressions which are derived from co-occurrence patterns of words. A set of concept vectors is called a concept base [3, 4]. The system converts words into vectors in a multidimensional space. The system first extracts 20,000 words from documents in descending order of the occurrence number, and then counts the co-occurrence number among the words [1]. The system has a feature of discarding unavailable words, words which appear only once in its documents database. These words do not take part in creating the concept base. The co-occurrence relation is described as a co-occurrence matrix.

	W_1	W_2	W_3	・・・	W_n
W_1	c_{11}	c_{12}	c_{13}	・・・	c_{1n}
W_2	c_{21}	⋱			・・・
W_3	c_{31}		⋱		・・ ・
・・・	・・ ・			⋱	・・・
W_n	c_{n1}	・・・	・・・	・・・	W_{nn}

W_n (n=1~20000) ; words in documents
c ; co-occurrence frequency

The system converts the co-occurrence matrix by using principal component analysis to stochastically set main axis in the vector space (**Fig. 2**)

	P_1	P_2	P_3	・・・	P_{100}
W_1	s_{11}	s_{12}	s_{13}	・・・	s_{1100}
W_2	s_{21}	⋱			・・・
W_3	s_{31}		⋱		・・・
・・ ・	・・・			⋱	・・ ・
W_n	s_{n1}	・・・	・・・	・・・	s_{n100}

$P_{1\sim100}$; principal component (PC)
s ; PC score

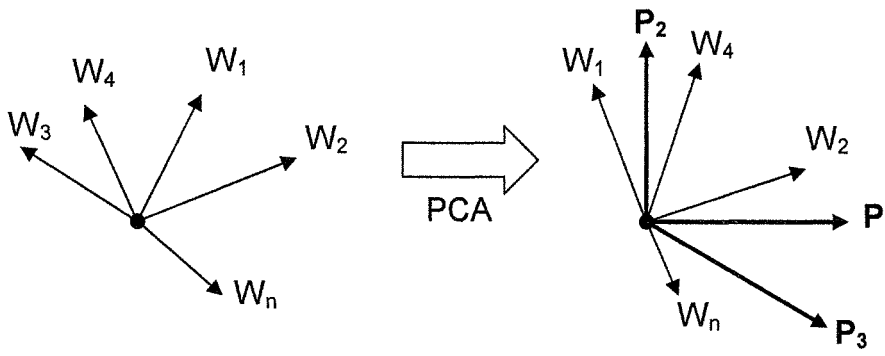

Fig. 2. Compression of co-occurrence matrix by principal component analysis

The vector of a word W_1 is $V_1 = [s_{11}\ s_{12}\ s_{13}\ \cdots\ s_{1100}]$.
The system then calculates similarities among the words from the word vectors. The similarities are the cosine between two word vectors.

The similarity

$$Sim(W_1 \cdot W_2) = \cos\theta = \frac{V_1 \cdot V_2}{|V_1||V_2|}$$

Moreover, a sentence vector is the mean value of all the word vectors included in the sentence.

3 Experiments

The higher the ranks of the target documents are, the better the system performance is.

We first prepared abstracts of 8,000 pocketbooks. These abstracts are the equivalent of the Q&A database which the auto response function owns. We secondly divided these abstracts into 10 folders having 800 abstracts each, and then made the system learn these abstract folders.

The system which holds i books is expressed in S_i (i=800~8,000).

We then randomly selected 100 abstracts from one of the folders. These abstracts were targeted. Before doing a search of the 100 abstracts, we had learned what was written in the abstracts and then made questions for them. We finally put the questions into the system to measure the ranks of the target abstracts. We prepared two types of questions.

Type A

We assumed that customer's questions have no keywords included in the target documents.

Then, we prepared 100 questions Q_{nk}, which have no keywords.

Type B

We assumed that customer's questions have some keywords included in the target documents.

Then, we prepared 100 questions Q_k, which have some keywords.

3.1 Setting co-occurrence range /Edit document data

Co-occurrence ranges among the words are a base element of the concept base. Therefore, the system performances should vary when the ranges are expanded. The ranges are between periods, that is, the ranges are one sentence. Therefore, the system users can expand the ranges by deleting the periods. In this study, we prepared three types of co-occurrence ranges.

Number one is one sentence. Number two is a single document. Number three contains all of the documents.

3.2 Learning common sense

The system does not have a basic knowledge database. The system creates the concept base only by learning documents. Subsequently, the system performance should decrease if the document quantity is not large enough. Therefore, we made the system learn a dictionary.

A set of vectors which created from the abstracts is expressed V_{object}, and a set of vectors from the dictionary is expressed V_{system}. In conclusion, the overall concept vector of system is

$$V_{system} = w \times V_{object} + V_{common}$$

Where, w is a weight of V_{object}.

The system performances vary when the weight w varies. The weight w can be changed by making the system learn copies of the abstracts. We controlled the system to learn copies of the abstracts. For example, if the system learns the same abstracts two times, the weight is w =2.

4 Results and discussion

We evaluated the system performance by measuring the ranks of the target abstracts. When internet users carry out searches on the Internet, they don't check all of the pages. Most users review only up to ten of the top pages. If they do not find suitable documents in the top ten pages, they input a new search request. In this paper, we evaluated the system performance by measuring the rate R_{10}, a ranking of no more than 10, in the 100 abstracts. **Fig. 3** shows the results of increasing the number of abstracts in the system. When the questions had some keywords, the rate R_{10} was about 36% in the S_{800}, 50% in the S_{8000}. When the abstracts were increased to 8,000 from 800, the rate was 14% better for Q_{nk}. When Q_k was entered to S_{800}, the rate was 75%. We found out that when the questions have some keywords, the rate R_{10} doubles.

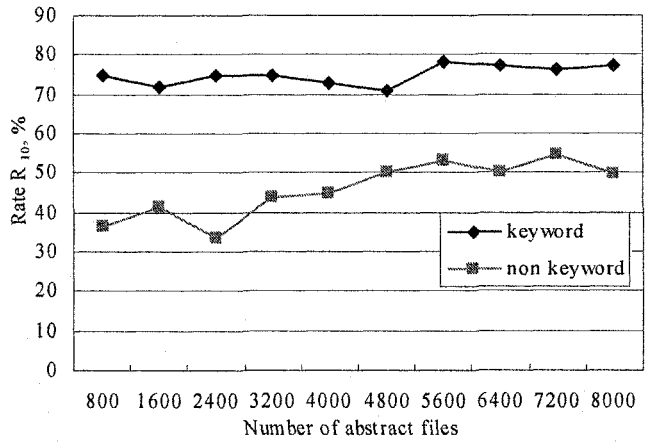

Fig. 3. Effect of keywords contained in questions

Next, **Fig. 4** shows the results of changing the co-occurrence range. The rate R_{10} increased and the system performance improved by expanding the range from one sentence to a single document, because the co-occurrence frequency increased and greater co-occurrence variations were created. However, the system performance didn't improve when the ranges were expanded to include multiple documents. The system users should set the single document ranges by deleting the periods.

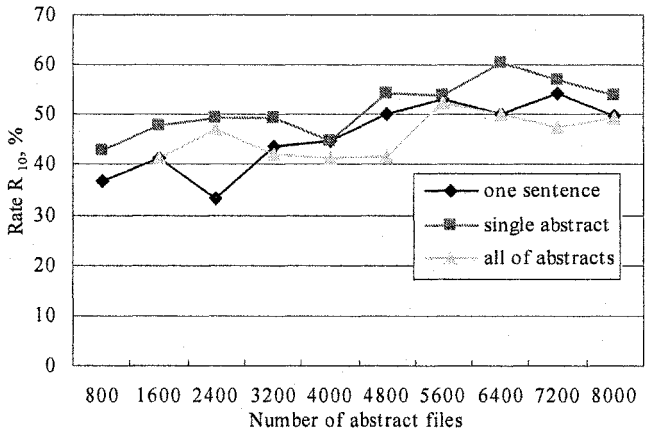

Fig. 4. Relation between system performance and co-occurrence range pattern

Fig. 5 shows the results of learning common sense, the dictionary. In S_{8000}, the rate R_{10} was about 60 % and higher by 5 % thanks to having the dictionary. However, the rate decreased a little in $S_{800\sim3200}$, because the system has and uses more common sense than knowledge created from the abstracts. The number of bytes in the 800 abstracts was 0.7MB, and the number of bytes in the dictionary was 35MB. The system users should tune the ratio of the dictionary to the object documents to balance them.

Next, we measured the rates while changing the ratio by making the system learn the same abstracts repeatedly. The results are showed in **Fig. 6**. The concept vector in the system is a composition of the dictionary vectors V_{common} and the abstract vectors Vo_{bject}. The overall concept vector V_{system} varies by tuning the ratio of the dictionary to the size of abstracts (**Fig. 7**). Therefore, the system performance varies. In this paper, the system learning the same 800 abstracts w times is expressed in $S_{800\text{-}w}$. The rate R_{10} was 38% in $S_{800\text{-}1}$, and 56% in $S_{800\text{-}8}$. The rate was 16% higher by making the system learn the same 800 abstracts 8 times. In $S_{800\sim3,200}$, the performance improved by the repeated learning, the system had more common sense than knowledge learned from the abstracts. The effect of the repeated learning is reduced in $S_{4800\sim8,000}$.

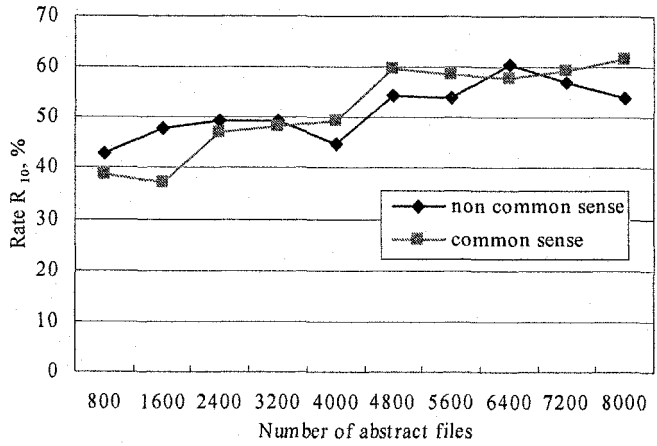

Fig. 5. Effect of learning common sense

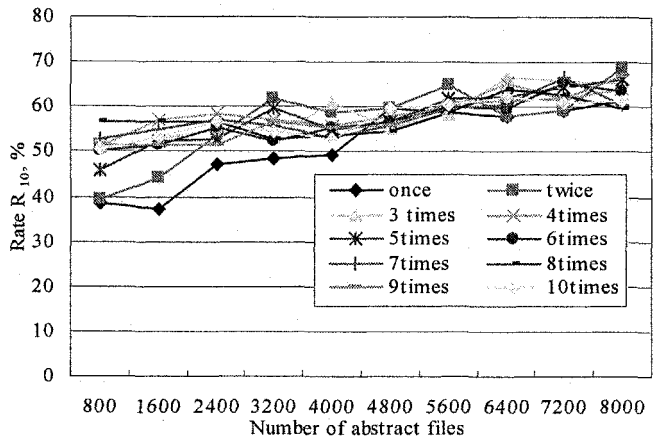

Fig. 6. Effect of repeated learning same documents

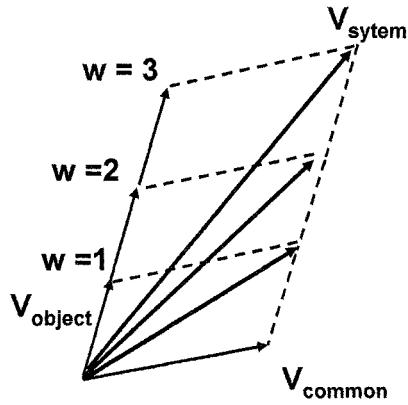

Fig. 7. Concept base variation by tuning the ratio

The experiment results show that when the co-occurrence range is a single document and $S_{8000\text{-}2}$ learns the dictionary, the system performs best. The rate R_{10} was 68 %. The rate is 33 % higher than S_{800} not tuned.

In order to obtain peak performance, the system should be operated in the best condition.

The system users can achieve better performance by trying the results and the measuring methods presented in this study.

5 Conclusion

The performance of text mining systems used in today's CRM systems does not meet user's expectations. This is specially true when the questions that the users input do not contain keywords. Through this study we discovered that the performance could be considerably improved and the rate R_{10} of about 70% was obtained by applying the following techniques.

- The co-occurrence ranges among words should be kept to a single document.
- The common sense of the system can be improved by inputting a dictionary.
- The ratio of the dictionary to the targeted documents should be balanced. The users should tune this ratio as necessary.

References

1. Koudai Aman and Fukuya Ishino, Performance Improvement of Text Mining, General Conference of IEICE, 2006, p. 44.

2. Koudai Aman, Satoshi Watanabe and Fukuya Ishino, A Proposal for Better Performance of Text Mining, in: Proceedings of International Conference on Operations and Supply Chain Management, Bali International Convention Center, The Westin Resort, Nusa Dua, Bali, 2005, pp. 25-33.

3. Yasuhiro Takayama, Text mining based on Concept Extraction for eCRM, Technical report of IEICE No. 3, 2003, pp. 19-23.

4. J. L. Neto, A. D. Santos, C. A. A. Kaestner and A. A. Freitas, Document clustering and text summarization, in: *4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000)*, (The Practical Application Company, London, 2000), pp. 41-55.