Automatically Selecting Answer Templates to Respond to Customer Emails

Rahul Malik, L. Venkata Subramaniam+ and Saroj Kaushik
Dept. of Computer Science and Engineering,
Indian Institute of Technology, Delhi,
Hauz Khas, New Delhi, India.

+IBM India Research Lab,
Block I, Indian Institute of Technology,
New Delhi, India.

Abstract

Contact center agents typically respond to email queries from customers by selecting predefined answer templates that relate to the questions present in the customer query. In this paper we present a technique to automatically select the answer templates corresponding to a customer query email. Given a set of query-response email pairs we find the associations between the actual questions and answers within them and use this information to map future questions to their answer templates. We evaluate the system on a small subset of the publicly available Pine-Info discussion list email archive and also on actual contact center data comprising customer queries, agent responses and templates.

1 Introduction

Contact center is a general term for help desks, information lines and customer service centers. Many companies today operate contact centers to handle customer queries, where, customers are provided email support from a professional agent. Contact centers have become a central focus of most companies as they allow them to be in direct contact with their customers to solve product and services related issues and also for grievance redress. A typical contact center agent handles a few hundred emails a day. Typically agents are provided many templates that cover the different topics of the customer emails. When an agent gets a customer query she selects the answer template(s) relevant to it and composes a response that is mailed back to the customer. Figure 1 shows an example query, response/template pair.

The work flow for processing emails at contact centers is as follows. When a customer sends in a query by email a human manually triages it and forwards it to the right agent. The agent has a number of predefined response templates from which she selects the most appropriate ones and composes a response email to the customer query by combining the templates and inserting additional text fragments. Incoming mails are in free text format with very little structured content in them. Usually the structured content are email fields like date, subject line, sender id, etc., which anyway do not help in triaging the email query or in answering them. Therefore, any technique that tries to automatically handle the email queries needs to use natural language processing to handle it.

```
Please provide the current status of the rebate
reimbursement for my phone purchases.
```

```
I understand your concern regarding the mail-in rebate.
For mail-in rebate reimbursement, please allow 8-14
weeks to receive it.
```

```
I understand your concern regarding the mail-in rebate.
For mail-in rebate reimbursement, please allow <replace
this> (***weeks or months***) </Replace this> to receive
it.
```

Figure 1: An Example Query, Response/Template Pair

Today's contact centers handle a wide variety of domains such as computer sales and support, mobile phones and apparel. For each of these domains typical answer templates are defined and provided to the agents to aid them in composing their responses. Contact centers have lots of such email data corresponding to each of the domains they handle. The customer queries along with the corresponding agent responses to them are stored and available in abundance.

In this paper we have presented techniques to automatically answer customer emails by

1. Extracting relevant portions from customer queries that require a response
2. Automatically matching customer queries to predefined templates to compose a response

We evaluate the performance of our technique and show that it achieves good accuracy on real life data sets.

2 Background and Related Work

2.1 Key Phrase Extraction

Extracting sentences and phrases that contain important information from a document is called key phrase extraction. Key phrase extraction based on learning from a tagged corpus has been widely explored [Hirao *et al.*, 2002] [Frank *et al.*, 1999]. The problem of extracting key phrases from emails using parts of speech tagging has also been studied [Tedmori *et al.*, 2004]. We mention this body of work in the context of this paper because in a given email it is important to identify the portions that correspond to questions and answers.

2.2 Text Similarity

Text similarity has been used in information retrieval to determine documents similar to a query. Typically similarity between two text segments is measured based on the number of similar lexical units that occur in both text segments [Salton and Lisk, 1971]. Similarity measures have been used to find similar documents and sentences [Metzler *et al.*, 2005]. However, lexical matching methods fail to take into account the semantic similarity of words. Hence, *I remained silent* and *I was quiet* would not be matched. To overcome this several techniques based on measuring word similarity have been proposed. In their work, Wu and Palmer [Wu and Palmer, 1994] measure the similarity of words based on the depth of the two concepts relative to each other in WordNet [1]. In [Mihalcea *et al.*, 2006] several corpus and knowledge-based measures have been evaluated for measuring similarity based on the semantics of the words. In the context of this paper we need to identify similar questions and we need to match a question to its answer.

2.3 Question Answering

Question Answering (QA) has attracted a lot of research. The annual Text REtrieval Conference (TREC) has a QA track where text retrieval systems are evaluated on common, real-world text collections. The aim of a system in the TREC QA task is to return answers themselves, rather than documents containing answers, in response to a question [Voorhees and Dang, 2006]. In essence what we are attempting to do in this paper is to answer the customer queries by looking at ways in which similar questions have been answered in the past.

2.4 Contact Center Email Processing

A lot of work hasebeenedone on automatic triaging of emails in contact centers [Nenkova and Bagga, 2003] [Busemann *et al.*, 2000]. In these a classifier is learnt based on existing cases and using features such as words in the mail, their parts of speech, type of sentence etc. So when new queries come in they are automatically routed to the correct agent. Not much work has been done on automatically answering the email queries from customers. It is possible to learn a classifier that map questions to a set of answer templates [Scheffer, 2004]. Also there has been some work on automatically learning question-to-answer mappings from training instances [Bickel and Scheffer, 2004]. Our work in this paper describes methods to automatically answer customer queries by selecting response templates.

2.5 Contributions of this Paper

In this paper we present techniques to automatically locate relevant questions in a customer query email and then map each of these to predefined answer templates. By putting together these templates a response email message is composed for the given query email. We automate the process of composing the response. We are not aware of work where a response is generated by putting together existing templates. Our technique is novel in that all the questions in the query email are first identified and the associations between the questions and the corresponding answers in the response emails are identified from the given corpus. In this way even multiple questions in a query email will get identified and addressed. Our technique shows good accuracy.

[1] http://wordnet.princeton.edu/

3 Problem Definition

We have a repository of matching query and response emails that have been exchanged between customers and call center agents. The responses have been composed out of a set of templates which have been provided to the agents. A single query may comprise many questions thus requiring multiple templates to be used in answering it. The problem is that of matching the individual questions in the queries to their corresponding templates.

Given the repository of query-response pairs and templates, the first problem is to learn the associations between questions and templates. Let $\{Q_1, \ldots, Q_m\}$ be the query emails and $\{R_1, \ldots, R_m\}$ be the corresponding responses. A query Q_i comprises the questions $\{q_l, \ldots, q_m\}$ which are matched with templates $\{t_l, \ldots, t_m\}$ used to compose the response R_i. The problem can now be defined as follows: Given a query email Q_i we need to find the set of questions, qs, in it and then match these to the corresponding templates, ts, that are used to compose the response R_i. Now that the associations have been established we can tackle the problem of automatically composing responses for new queries. When a new query comes in, we need to identify the questions in it and then compose a response for it using the mappings that have previously been established.

4 Email Triaging

In a contact center typically customer query emails are manually triaged. In the triaging step, emails are forwarded to the concerned agent handling the particular class of queries. In some cases a single agent handles all the classes of queries and hence the emails are not actually classified. So the classification information is not available. We replace this step with automatic triaging. Since obtaining labelled data for a new process is hard, we use clustering to first identify the different classes and then learn a classifier based on these classes. We classified the emails from the larger pool into an equal number of query and response clusters using text clustering by repeated bisection using cosine similarity. Matching query and response clusters could contain different numbers of emails. So, they were subsequently matched based on maximal matching criteria. The cluster having the maximum match fraction was separated and the remaining clusters were rematched. This way a one-to-one mapping of the query and response clusters was obtained. The non-matching emails for each map were removed to create clean clusters having exact correspondence of query-response emails. An SVM classifier was then trained on the classes created by the clustering. This classifier is then used to triage the emails.

5 Extraction of Question-Answer Pairs

Given a query email we would like to extract the key questions in it. Then we would like to map these questions to

their answers in the response email and determine the templates that have been used.

5.1 Identification of Questions and Answers

We identify the questions based on the presence of key phrases within them. We follow the approach of [Frank *et al.*, 1999] to identify the key phrases of length up to 3 words. We use a set of training documents for which key phrases are known to generate a naive Bayes model. This model is then used to find key phrases in a new document.

In actual fact the tagged training documents were not available. So we first extracted the most frequent unigrams, bigrams and trigrams and selected from this list. These extracted phrases of length up to 3 words were then marked in the emails as being the key phrases. This tagged set was used to generate a naive Bayes model for the key phrases.

In a query email the questions are identified as those sentences that contain key phrases in them. Similarly in the response email, the replies are identified as those sentences that contain key phrases in them.

5.2 Mapping Questions to Answers

Once the questions and responses have been identified we need to map each question to its corresponding response. A given query email may contain multiple questions and therefore the response may contain multiple answers. To accomplish this mapping we first partition each extracted sentence into its list of tokens. Stop words are removed and the remaining words in every transcription are passed through a stemmer (using Porter's stemming algorithm [2]) to get the root form of every word e.g. *call* from *called*. Now the similarity between each question and answer is calculated to find the correct map. In order to calculate the overlap between the two sentences, we used Word Overlap Measure along with the score being adjusted to take into account the inverse document frequency [Metzler *et al.*, 2005]. The measure is given as:

$$sim(q,r) = \frac{|q \cap r|}{|q|}\left(\sum_{w \in q \cap r} (log(N/df_w))\right)$$

where, N = total number of words, $|q \cap r|$ = collection of common words of q and r after stop-word removal and stemming, df_w = document frequency of w, where w belongs to the common word set of q and r

This gives us a score of similarity between question and answer. In addition, we used a heuristic that if a question is asked in the beginning, then the chances that its response would also be in the beginning are more. So, we biased them with a high weight as compared to those lying in the end which were penalized with a low weight.

So, the expression for score becomes:

$$score(q,r) = sim(q,r) * (1 - |\frac{pos(q)}{(N+1)} - \frac{pos(r)}{(M+1)}|)$$

where, $sim(q,r)$ = similarity score as obtained from above, $pos(q)$ = position of the Question in the set of Questions of that query-email, N = number of questions in that query-email, M = number of answers in that response-email.

Each answer is then mapped to a template. This is done by simply matching the answer with sentences in the templates. The agent typically makes very little changes to the template while composing an answer. As a result of these steps we obtain the question to template pairs.

Multiple questions can match the same template because different customers may ask the same question in different ways. Hence, for example, while *what is my account balance* and *can i know the balance in my account* can be merged into one sentence, *i need to know the carry forward from last month* cannot easily be linked with the earlier sentences. Hence while the mapping method gave the question-to-template mappings, we need to merge similar sentences and keep the minimal set of questions associated with a template. We prune the set of questions by removing questions that have a very high similarity score between them. So that when a new question comes in, we only need to compare it with this minimal set.

6 Answering New Questions

When we get a new query we first identify the questions in it by following the method outlined in Section 5.1. Each of these questions now needs to be answered. To compose the response email the answers to each of these questions are put together.

6.1 Mapping New Questions to Existing Questions

When a new question comes in we need to determine its similarity to a question we have seen earlier and for which we know the template. From the previous discussion we know that many different questions can map to the same answer template. If we are able to determine that the new question is similar to an existing question then we are done. The new question is mapped to the template for the existing question to which it is similar.

The similarity between two concepts is given as [Wu and Palmer, 1994]:

$$Sim(s,t) = 2 * \frac{depth(LCS)}{[depth(s) + depth(t)]}$$

where, s and t denote the source and target words being compared. $depth(s)$ is the shortest distance from root node to a node s on the taxonomy where the synset of s lies LCS denotes the least common subsubmer of s and t. We are using Wordnet.

Further we need to compute the similarity between the concepts present in the sentences that we want to match. The two sentences are first tokenized, stop words are removed and Porter stemmed. Parts of speech labels are assigned to the tokens. Nouns, verbs, adjectives and adverbs were selected. In addition, we also keep the cardinals since numbers also play an important role in the understanding of text. We form 5 matrices, one for each of the five classes. The rows of the matrix are the tokens from one sentence and the columns are the tokens from the second sentence. Each entry in the matrix $Sim(s,t)$ denotes the similarity as it has been obtained above

[2] http://www.tartarus.org/ martin/PorterStemmer

for the pair. Also, if a word s or t does not exist in the dictionary, then we use the edit distance similarity between the two words.

The results from these are combined together in the following manner to obtain the overall similarity score between the two sentences

The similarity between two sentences is determined as follows [Mihalcea *et al.*, 2006]:

$$Score(q_i, q_j) = \frac{\sum_{s \in q_i} Sim_m(s, q_j) + \sum_{s \in q_j} Sim_m(s, q_i)}{|q_i| + |q_j|}$$

where, $Sim_m(s, q_j)$ is the word in q_j that belongs to the same class as s (noun or verb, etc.) and has the highest semantic similarity to the word s in q_i.

Using the above similarity we compare the new question with the questions seen earlier. When a match is found we select the response template of the matched question as the template for the new question.

7 Evaluation

In this section we present the results over real life data sets.

7.1 Contact Center Data

We chose two sets of data to work on. The Pine-Info discussion list web archive [3]contains emails of users reporting problems and responses from other users offering solutions and advice. The questions that users ask are about problems they face in using pine. Other users offer solutions and advice to these problems. The Pine-Info dataset is arranged in the form of threads in which users ask questions and replies are made to them. This forms a thread of discussion on a topic. We choose the first email of the thread as query email as it contains the questions asked and the second email as the response as it contains responses to that email. It may not contain answers to all the questions asked as they may be answered in subsequent mails of the thread. We randomly picked a total of 30 query-response pairs from Pine-Info. The question sentences and answer sentences in these were marked along with the mappings between them. On average, a query email contains 1.43 questions and the first response email contains 1.3 answers and there are 1.2 question-answer pairs. We show a query and response pair from Pine-Info in Figure 2. The actual question and answer that have been marked by hand are shown in bold. In the example shown two questions (two sentences) have been marked in the query email but only one answer is marked in the response email.

The second data set used was obtained from an actual contact center process relating to mobile phone services. We obtained 1320 query and response emails relating to the mobile phone support process. We also obtained the 570 answer templates that the agents are provided to aid them in answering the customer queries. On an average each response contains 2.3 templates, implying that there are about two key questions per customer query. In Figure 3 a query is shown along with its actual response and the template used in composing the response.

[3]http://www.washington.edu/pine/pine-info

I am working on creating an rpm for our university, and I need to put pine.conf into /usr/local/etc/pine.conf instead of /etc/pine.conf. **I am wondering if there are any extra parameters that I can pass at build time to redirect pine to /usr/local/etc/pine.conf. If there isn't a parameter that I can pass does anyone know of a patch that would do something like that?**

Yes, what I would recommend you is that you take a look at the build script. The port for RedHat, includes a "ALTDOCPATHS = 1", which defines a few extra variables, in particular look at the -DSYSTEM_PINERC="/etc/pine.conf" and so on. That should give you an idea of what to do.

Figure 2: A typical Query, Response Pair from Pine-Info

I signed up for wireless web 2 or 3 months ago and have been charged $5.00 each bill. This time it was $25.00. What is the problem?

I am sorry for the inconvenience caused to you. I have adjusted your account for $20. This will be reflected on your May bill.

I am sorry for the inconvenience caused to you. I have adjusted your account for <replace this> (***$INSERT AMOUNT or MINUTES ***) </Replace this>. This will be reflected on your <replace this> (***month***) </Replace this> bill.

Figure 3: A Typical Query, Response/Template Pair

7.2 Measures

We measure the accuracy of the system by comparing the email response generated by a human agent with that generated by our system. We use two criteria for comparison. In the first case, we say that the system is correct only if it generates the exact answer as the agent. In the second case we allow partial correctness. That is if the query contains two questions and the system response matched the agent response in one of them then the system is 50% correct for that query.

7.3 Experiments

In this section we show the results of our techniques both on the pine dataset and the contact center emails. In pine there are no templates. So in effect we are only testing whether we are able to map the manually extracted questions and answers correctly. We used the 30 annotated pairs of query and response emails to measure this. As already mentioned we are looking for an answer in the first response to the query. Hence, all questions in the query may not get addressed and it may not be possible to get all the mappings. In Table 1 we show the numbers obtained. Out of a total of 43 questions only 36 have been mapped to answers in the manual annotation process. Using the method presented in Section 5.2 we are able to find 28 of these maps correctly.

For the contact center emails we tested whether our method could accurately generate the responses for the queries. We built our system using 920 of the 1320 query-response pairs of emails available to us. We then tested using the remaining 400 pairs of emails. We ran two sets of experiments on these emails. In the first case we didnt use class information to generate the response emails and in the second case we triaged

Table 1: Results on Pine-Info Dataset for Question-Answer Mapping

total mails	total qns	total ans	actual maps	correct maps	% correct maps
30	43	39	36	28	77.78%

the emails first to aid in the generation of templates.

In Table 2 we show the results for generating the response email without using any class information. Hence, in this case, the question-to-template maps are established using the 920 training email pairs. For each template the questions associated with them are found using the techniques of Section 5.2. For new questions in the testing set the response is automatically generated as discussed in Section 6.1. We generated the exact response that the human generated in 61% of the cases. Considering partial correctness this accuracy was about 73.4%.

Table 2: Results on Contact Center Dataset without Triaging

total email pairs	training set	testing set	total correct resp.	% correct resp.	% partial correct
1320	920	400	244	61	73.4%

For the case of triaging the emails. We first clustered the 920 query and response sets separately. We used CLUTO package [4] for doing text clustering. We experimented with all the available clustering functions in CLUTO but no one clustering algorithm consistently outperformed others. Also, there was not much difference between various algorithms based on the available goodness metrics. Hence, we used the default repeated bisection technique with cosine function as the similarity metric. The clustering results using CLUTO are shown in Table 3. We chose to cluster into 5 clusters because other values for the total number of clusters resulted in lower purity numbers in CLUTO. We also generated the top descriptive feature for each cluster using CLUTO and found this to be very close to the actual label for that class which was assigned by a human. We found that the average number of templates varied quite a lot between the classes. Many templates were used in two classes or sometimes even more. We used libSVM [5] to train an SVM classifier on the clusters that we obtained. We used the rbf kernel with $\gamma = 0.00012207$ $\rho = -0.121623$. This classifier was used for triaging the emails. The resulting overall accuracy for the classification was over 89%. In this case, the question-to-template maps are established using the training email pairs for each class separately. For each template the questions associated with them are found using the techniques of Section 6.2. For new questions in the testing set the email is first triaged using the classifier that has been learnt. Then the response is automatically generated as discussed in Section 6.3. The questions in this email are only matched to questions from the same class. We generate the exact response that the human generated in

[4] http://glaros.dtc.umn.edu/gkhome/views/cluto

[5] http://www.csie.ntu.edu.tw/c̃jlin/libsvm

Table 3: Results of Clustering Contact Center Dataset

Actual class	descriptive CLUTO label	total no. emails	total templates in class	templates per email
Rebate	rebate	62	48	2.1
Cancelling	cancel	55	42	3.4
Charges	charge	277	214	1.5
Account payment	payment	238	194	2.2
General queries	phone	688	397	2.6
Overall		920	570	2.3

over 79% of the cases. Considering partial correctness this accuracy goes up to about 85%. These results are shown in Table 4.

It is seen that there is a marked jump from 61% to 79% in the accuracy numbers for complete match if the emails are triaged. We do not notice such a large jump in the case of partial correctness. The reasons for this could be twofold. Firstly, the step of triaging the emails limits the search space over which the questions need to be matched. Secondly, if a query lands up in a wrong cluster due to misclassification then it stands no chance of being partially correct because that particular class may not have the template for it.

Table 4: Results Contact Center Dataset with Triaging

CLUTO class label	total no. emails	classification accuracy	% correct responses	% partial correct
rebate	16	87.6	82.1	86.4
cancel	16	83.4	78.2	78.8
charge	74	88.9	74.2	85.4
payment	74	91.6	83.4	88.7
phone	212	92.7	82.1	86.3
overall	400	89.4	79.1	84.8

Also it should be mentioned here that often there are questions from customers in reply to which an agent does not use any of the existing templates but actually composes her own reply.

8 Conclusions

In this paper we have introduced an automatic email response generation technique. Given a corpus of query-response email pairs and answer templates, we determine the typical questions for the given answer templates. When a new query email is presented we identify the questions in it and find the nearest questions to these that had been identified along with their answer templates. The answer templates relating to the questions are put together and composed into a response.

The accuracies obtained using the methods presented in this paper point to the fact that such a system is practically possible and useful to a contact center agent. When composing a response to a customer query the agent has to select from a few hundred templates. The system presented in this

paper helps improve agent efficiency and speed. The templates need some modification and filling in which still has to be done by the agent.

In future we plan to improve our system to handle questions for which the predefined templates do not work. We would also like to attempt to fill in some of the details in the templates to further aid the agent.

References

[Bickel and Scheffer, 2004] Steffen Bickel and Tobias Scheffer. Learning from message pairs for automatic email answering. In *Proceedings of the European Conference on Machine Learning,*, pages 87–98, Pisa, Italy, September 20-24 2004.

[Busemann *et al.*, 2000] Stephan Busemann, Sven Schmeier, and Roman G. Arens. Message classification in the call center. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 158–165, Seattle, Washington, April 29-May 4 2000.

[Frank *et al.*, 1999] Eibe Frank, Gordon W. Paynter, Ian H. Witten, Carl Gutwin, and Craig G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proc. Sixteenth International Joint Conference on Artificial Intelligence,*, pages 668–673, San Fransisco, CA, 1999.

[Hirao *et al.*, 2002] Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. Extracting important sentences with support vector machines. In *Proceedings of 19th International Conference on Computational Linguistics (COLING)*, pages Vol. 1, 1–7, Taipei, Taiwan, August 24–September 1 2002.

[Metzler *et al.*, 2005] Donald Metzler, Yaniv Bernstein, W. Bruce Croft, Alistair Moffat, and Justin Zobel. Similarity measures for tracking information flow. In *Proceedings of the Conference on Information and Knowledge Management,*, pages 517–524, Bremen, Germany, October 31-November 5 2005.

[Mihalcea *et al.*, 2006] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the American Association for Artificial Intelligence,*, Boston, MA, USA, July 2006.

[Nenkova and Bagga, 2003] Ani Nenkova and Amit Bagga. Email classification for contact centers. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, pages 789–792, Melbourne, FL, USA, March 2003.

[Salton and Lisk, 1971] G. Salton and M. E. Lisk. *Computer Evaluation of Indexing and Text Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

[Scheffer, 2004] Tobias Scheffer. Email answering assistance by semi-supervised text classification. *Journal of Knowledge and Process Management*, 13(2):100–107, 2004.

[Tedmori *et al.*, 2004] Sara Tedmori, Thomas W. Jackson, and Dino Bouchlaghem. Locating knowledge sources through keyphrase extraction. *Intelligent Data Analysis*, 8(5), 2004.

[Voorhees and Dang, 2006] Ellen M. Voorhees and Hoa Trang Dang. Overview of the trec 2005 question answering track. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC-2005)*, pages 119–128, Gaithersburg, MD, November 2006.

[Wu and Palmer, 1994] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 133–138, Las Cruces, New Mexico, June 27-30 1994.