HotSpots: Visualizing Edits to a Text

Srinivas Bangalore
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
srini@research.att.com

David Smith
AT&T Labs – Research
180 Park Ave
Florham Park, NJ 07932
dsmith@research.att.com

Abstract

Compared to the telephone, email based customer care is increasingly becoming the preferred channel of communication for corporations and customers. Most email-based customer care management systems provide a method to include template texts in order to reduce the handling time for a customer's email. The text in a template is suitably modified into a response by a customer care agent. In this paper, we present two techniques to improve the effectiveness of a template by providing tools for the template authors. First, we present a tool to track and visualize the edits made by agents to a template which serves as a vital feedback to the template authors. Second, we present a novel method that automatically extracts potential templates from responses authored by agents. These methods are investigated in the context of an email customer care analysis tool that handles over a million emails a year.

1 Introduction

Email based customer care is increasingly becoming the preferred channel of communication for corporations and customers compared to the conventional telephone-based customer care. For customers, email channel offers several advantages – there are no tedious menus to navigate, there is no waiting time to reach an operator, the request can be formulated at the customer's pace and additional material supporting the case can be attached to the email. There is also a record of the service request for the customer unlike the telephone-based customer care. However, there are also limitations of the email channel. The most significant one is that the customer-agent interaction could be drawn out over successive emails spanning over several days as opposed to being resolved in one or two telephone calls. For corporations, the asynchronous nature of email-based customer care offers significant opportunities to reduce operations cost by effective load balancing compared to telephone-based customer care. It is quite common for an email customer care agent to work on several cases simultaneously over a period of a few hours. Email channel also offers higher bandwidth for corporations to send additional information in the form of web links, images and video or audio instructions.

The effectiveness of customer care in the email channel is measured using two competing metrics: Average Handling Time (AHT) and Customer Experience Evaluation (CEE). AHT measures the time taken from when a customer email is opened to the time when the response is sent out. This time is typically averaged over a period of a week or a month for reporting purposes. CEE measures customer satisfaction through a survey of a random subset of customers who have interacted with the email customer care center. These surveys typically involve qualitative and quantitative questions and measure the quality of the interactions along a number of different dimensions. As is the case in many surveys the population responding to such questionnaires is typically small and very often quite biased. We do not use the CEE metric for the work we report in this paper.

As is evident from the definitions of AHT and CEE, it is in the interest of a corporation to minimize AHT while maximizing CEE. In order to reduce AHT, most email customer care systems (Kana, 2008; Genesys, 2008) provide a mechanism for an agent to respond to a customer's email by selecting a predefined template text that can be quickly customized to serve as the response. The template text is usually associated with a problem category it is intended to address and might even be suggested to the agent automatically using classification techniques

Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages 264–273,
Honolulu, October 2008.

applied to the customer's email. Once the template is selected, the agent edits the template text to personalize as well as add case specific details as part of composing a response. Each of the text edits contributes to the handling time of the email. Hence, it is in the interest of the template designer to minimize the number of edits of the template in order to lower AHT.

Although most email management systems provide a mechanism to author the template text, there is typically no mechanism to monitor and track how these templates are modified by the agents when they compose a response. This information is vital to the template authors when creating new versions of the templates that reduce the number of edits and consequently reduce AHT.

In this paper, we present two methods for improving the templates in a principled manner. After describing the related work in Section 2, we present a brief description of the email tracking tool we have developed in Section 3. In Section 4, we present a tool called HotSpots that helps visualize the edits being made by the customer care agents to the templates. This tool provides a visual feedback to the template authors and suggests means of improving the template text based on the edits made by agents. In Section 5, we present a new approach to automatically identify emerging templates – texts that are repeatedly created by agents and are similar to each other but distinct from the current template text. We use AHT as the metric to minimize for automatic identification of emerging templates. We discuss some of the issues concerning this work in Section 6 and conclude in Section 7.

2 Related Work

There are few threads of research that are relevant to the work presented in this paper. First, the topic of email response generation in the context of customer care has been investigated by (Coch, 1996; Lapalme and Kosseim, 2003; Zukerman and Marom, 2007). In (Coch, 1996), the authors model multi-sentence generation of response letters to customer complaints in French. The generation model is carefully crafted for the domain using domain-specific rules for conceptual planning, rhetorical relations and surface word order operators. They show that their approach performs better than predefined templates and slightly worse than human generated responses. In (Lapalme and Kosseim, 2003), the authors explore three different approaches based on classification, case-based reasoning and question-answering to compose responses to queries in an email customer care application for the telecommunication industry. The case-based reasoning approach is the most similar to the template approach we follow. In (Zukerman and Marom, 2007), the authors investigate an approach to assembling a response by first predicting the clusters of sentences to be included in the response text and then applying multi-document summarization techniques to collate the representative sentences into a single response. In contrast, in this paper, due to constraints from the deployment environment, we rely on a template-based approach to response generation. We focus on providing tools for investigating how the templates are modified and suggest techniques for evolving more effective templates based on quantitative criteria.

Another thread of relevant research are methods for visualizing texts. There are several methods that have been proposed to provide a visual map of a set of text documents with the focus of illustrating the relatedness of these texts (Card et al., 1999). Using a metric for comparing texts (e.g. n-gram overlap) , the texts are clustered and the resulting clusters are visualized as two or three dimensional color maps. These approaches are useful to depict similarities in a static repository of documents or the return results of a search query. These maps are primarily designed for exploration and navigation through the document space. While the underlying algorithm we use to illustrate the text edits is similar to the one used in text map visualizations, our focus in this paper is to provide a mechanism for template designers to quickly identify the variants of a template sentence created by the agents.

A third thread is in the context of human-assisted machine translation, where a human translator post-edits the output of a machine translation system (Foster et al., 1997; Foster et al., 2002; Och et al., 2003). In order to improve the efficiency of a human translator, the k-best output of a translation system could be displayed as word or phrase choices which are color coded based on the confidence value assigned by the translation model. While the ap-

proach we follow is partly motivated by the post-editing paradigm, there are significant differences in the context we apply this approach. In the context of this paper, the template designer is presented a summary of the set of variants created by each agent for each sentence of the template. The task of the template designer is to use this tool to select (or construct) a new variant for the template sentence with the aim of minimizing the need for editing that sentence in future uses of the template.

3 Email Customer Care

Typically, a large email customer care management center receives over 100,000 emails a month. These centers typically use a customer care management system that offer not only logging and tracking of emails but also tools for improving the efficiency of agents responding to emails. Usually, an incoming customer email is categorized into a set of few topics/issues. The categorization might be done automatically based on regular expressions involving keywords in the email or using weighted classifiers that are trained on data. In order for an agent to respond to an incoming email, these systems provide a text box which allows the agent to author a response from scratch. However, most email customer care systems offer the ability to store a prefabricated response (also called *templates*), instead of agents having to author a response from scratch. These templates are typically associated with a problem category or an issue that they are intended to address.

A template helps an agent compose a well-formed response quickly. It contains hints for information that the agent should enter as well as indications of where that information should be entered in the template. The template might also contain helpful information to the customer in addition to legal verbiage that the customer needs to be aware of.

An agent receives a customer email and after comprehending the issues and consulting the customer records in the database, selects one of the predefined set of templates that best addresses the issues raised in the email. Less frequently, she might even select more than one template to compose the response. She then proceeds to edit and personalize the chosen templates to better suit the customer's email. An example of a 'generic' template – not associated with a specific problem category is shown in Figure 1.

> Greetings Contact.FirstName,
> Thank you for your email in regard to XXXXXXXX.
> I will be happy to assist you with your inquiry.
> XXX BODY XXX
> If I can be of any further assistance,
> please reply directly to this email.
> Thank you for using our company.
> We appreciate your business and continued loyalty.
> Regards,
> Agent.FirstName

Figure 1: An example of a generic template

The process of selecting an appropriate template that addresses the customer's inquiries could be quite tedious when there are hundreds of templates. Email management systems offer tools that suggest appropriate template to use based on the content of the customer's email. These tools are trained using classification techniques on previous email interactions.

As mentioned earlier, there are two metrics that are typically used to measure the effectiveness and efficiency of email responses. Customers are surveyed after their email interaction to assess their level of satisfaction for the service they received. This is usually called the Customer Experience Evaluation (CEE) and includes an evaluation of the customer's total interaction experience with the corporation, not just the last email interaction. A small subset of customers who had an interaction with the email center is randomly chosen (typically in the order of about 10% of customers) and are invited to take part in the follow-up survey. Typically, only a small percent (about 10%) of the customers who receive these invitations respond to the survey; effectively about 1% of the total emails have customer survey scores.

A second metric that is also used to measure the efficiency of an operation is called the average handling time (AHT) which measures the average of

times taken by agents to respond to emails. The handling time includes the time to comprehend the email, the time for database lookup and the time for response composition. It is in the interest of the email customer care operation to minimize AHT and maximize CEE scores.

3.1 Email Customer Care Analysis Tool

We have designed and developed an Email Customer Care Analysis Tool (ECAT) to help analyze the operations of the email care center. It provides an end-to-end view from the activities involved in answering emails to the results of subsequent customer care surveys. In addition, ECAT also provides insights into how the agents are editing templates as well as guides template authors in designing more effective templates.

ECAT is a web-based tool and offers a birds-eye summary of the operations aggregated by region, the template used, and the customer satisfaction survey results. Using this tool, analysts can drill down through a series of views until they are eventually presented with the results of a single survey or a single email interaction.

One of the most useful functions of the tool is that it shows the extent to which agents edit the templates in the process of creating responses to customer emails. The degree to which a template is edited is based on Levenshtein string edit distance metric (Levenshtein, 1966). This metric measures the number of edits (substitution, deletion and insertions) of words that are needed to transform a template into a response. The number of edits is normalized by the number of words in the template. These morphing scores can be viewed for a single email or averaged per agent or per template used. The scores range from 100 to 0, with 100 representing a template which hadn't been edited at all.

The tool also allows the morphing score to be viewed alongside the handling time for an email, in other words the amount of time that the agent spends gathering data and actually composing a response. Handling time is an important metric since it is directly related to cost of operating the email customer care center. The more editing an agent does, the more time they take to respond to a customer. So, the number of templates, their precise wording and the ease with which agents can distinguish them obviously have significant influences on overall handling time.

Beyond the confines of the email centers themselves, the CEE is the most important elements in gauging the effectiveness of the agent. The survey asks customers to rate their overall satisfaction with the email reply to their question. Five is the highest score which equates with 'Extremely Satisfied' while a one equals 'Extremely Dissatisfied.' Customers are also asked to rank the email in terms of it's content, clarity, professionalism and the length of time it took to receive a reply. The customer is also allowed to enter some free text so that they can say how satisfied they were, or not, with how an inquiry or problem was dealt with. Customers can also say whether they called the company using a telephone channel before turning to the email channel.

The survey files, all of which can be accessed in their entirety from within the ECAT tool, also contain information on what templates were used when replying to the customer. They also tell the analyst who the replying agent was and whether this was the first or a subsequent email in communications between the customer and the company.

The ECAT tool juxtaposes this CEE score with the template morphing score to show correlations between customer satisfaction and the degree to which the template had been edited. This data is graphed so that the analyst can immediately see if heavy editing of a template is leading to higher CEE. Heavy editing with a low customer rating could mean that the template is not helping the agent to respond correctly to the customer.

4 HotSpots

We designed the HotSpots tool that provides insights to the template authors on how templates are being edited by the agents when creating responses. It suggests methods for improving the next version of the template so as to reduce edits by agents and hence reduce the handling time for an email. In this section, we discuss the algorithm and the visualization of the information that aids template authors in improving the efficacy of the templates.

The HotSpots algorithm proceeds in two steps as shown in Algorithm 1. The first step creates an alignment between the template string and the re-

Algorithm 1 Compute HotSpots for a template T given a response set $\mathcal{R}$

$EdEv = \phi$
$T = s_1 s_2 \dots s_n$
$T_s = \{s_i | 1 \leq i \leq n\}$
$R_s = \{r_i^j | R_j \in \mathcal{R}, R_j = r_1^j r_2^j \dots r_{m_j}^j, 1 \leq i \leq m_j\}$
$Index : \{T_s \cup R_s\} \rightarrow \mathbb{I}$
$T_{in} = Index(s_1) Index(s_2) \dots Index(s_n)$
for all $R \in \mathcal{R}$ **do**
  $R = r_1 r_2 \dots r_{n_R}$
  $R_{in} = Index(r_1) Index(r_2) \dots Index(r_{n_R})$
  // compute distance with sentences as tokens and return the alignment and score
  $(alignment, score) = IndDist(T_{in}, R_{in})$
  // for each of the sentences in T, update its map
  for all $s_i \in T$ **do**
    $in = Index(s_i)$
    if $(s_i, \epsilon) \in alignment$ **then**
      $EdEv[in].map = EdEv[in].map \cup \{*delete*\}$
    else // $(s_i, r_j) \in alignment$
      $EdEv[in].map = EdEv[in].map \cup \{r_j\}$
    end if
  end for
end for
// Cluster the response sentences aligned for each template sentence
for all $s_i \in T$ **do**
  $in = Index(s_i)$
  $Cl = KmedianCl(EdEv[in].map, ncl)$
end for

Algorithm 2 KmedianCl: Compute k centroids for a set of strings S using k-median clustering algorithm

$c_s = \phi$ // centroid of string s's cluster
$ce_i = \phi$ // centroid of cluster i
$numcl = 0$ // number of cluster created so far
$Cl_i = \phi$ // members of cluster i
while $(numcl \leq k) \wedge (numcl \leq |S|)$ **do**
  if $numcl = 0$ **then**
    $ce_0 = \underset{c \in S}{argmin} \sum_{s \in S} Dist(c, s)$
  else // select the string (s) that is farthest from its centroid (c_s)
    $ce_{numcl} = \underset{s \in S}{argmax}\ Dist(c_s, s)$
  end if
  // Move strings to the closest cluster and compute centroids until the set of centroids don't change
  repeat
    for all $s \in S$ **do**
      $i^* = \underset{0 \leq i \leq numcl}{argmin}\ Dist(ce_i, s)$
      $c_s = ce_{i^*}$
      $Cl_{i^*} = Cl_{i^*} \cup \{s\}$
      // Computed the closest cluster centroid ce_i to s.
    end for
    // Recompute the cluster centroids ce_i
    for all i such that $0 \leq i \leq numcl$ **do**
      $ce_i = \underset{c \in Cl_i}{argmin} \sum_{s \in Cl_i} Dist(c, s)$
    end for
  until set of centroids does not change
  $numcl = numcl + 1$ // new cluster added
end while

sponse string. For the purposes of this paper, we consider the alignments between the template text and the response text with sentences as tokens instead of a word-based alignment. The rationale for this tokenization is that for template developers the visualization of the edits is expected to be more meaningful when aggregated at the sentence level rather than at the word level. In the second step, using the sentence-level alignment we compute the edit events (insertion, deletion and substitution) of the template sentences in order to create the response. All the edits events associated with a template sentence are then clustered into k clusters and the centroids of the k clusters are displayed as the potential changes to that sentence. We next describe these two steps in detail as illustrated in Algorithm 1 and Algorithm 2.

Given a set of responses $\mathcal{R}$ that agents create using a template T, Algorithm 1 proceeds as follows. Each of the sentences in the template and the set of responses are mapped into an integer index (Line 1). The template T and each of the responses in R are split into sentences and mapped into index sequences (Line 6 and Line 9). The alignment between the two index strings is computed in Line 10. This is a dynamic programming algorithm similar to computing Levenshtein distance between two strings, except the cost function used to compute the match between tokens is as shown below.

From the alignment that maps s_i to r_j, we collect the set of response sentences associated with each template sentence (Line 13-16). These sentences are then clustered using k-median clustering method (illustrated in Algorithm 2) in Line 22.

In Algorithm 2, we illustrate the method of clustering we use to summarize the set of sentences we have collected for each template sentence after the alignment step. The algorithm is similar to the k-means algorithm (Duda et al., 2001), however, given that we are clustering strings instead of real numbers (as is typical in applications of k-means), we restrict the centroid of a cluster to be one of the members of the set being clustered, hence the name k-median algorithm (Martnez-Hinarejos et al., 2003). The distance function to measure the closeness of two strings is instantiated to be an n-gram overlap between the two strings.[1]

The algorithm iterates over three steps until the data is partitioned into k clusters (Line 5). The first step (Lines 6-10) is the initialization of a centroid for a new cluster. Initially when the data is not partitioned into any cluster, the median string of the data set is used as the initial centroid. For subsequent iterations, the farthest point from all the centroids computed thus far is used as the centroid for the new cluster. In the second step (Lines 11-16), each member of the data set is assigned to the nearest cluster based on its distance to that cluster's centroid. Finally, in the third step (Lines 17-20), the cluster centroids are recomputed based on the new cluster memberships. Steps two and three are repeated until there are no changes in the cluster memberships and cluster centroids. This completes the introduction of a new cluster for the data.

For the purposes of our task, we use up to a four-gram overlap to measure distance between two strings and use $k = 5$ for clustering the data.

4.1 Visualizing the HotSpots

The HotSpots page was created within the ECAT tool to surgically dissect the way in which templates were being morphed. For a given template, as shown in Figure 2, the analyst is presented with a copy of the texts from the current and previous versions of that template. Each sentence in the two versions of the template are color coded to show how frequently the agents have changed that sentence. This involved running the HotSpots algorithm against approximately 1,000 emails per template version. A sentence that is colored red is one that was changed in over 50% of the emails that were responded to using that template. An orange sentence is one that was edited in between 30% and 50%, green is between 10% to 30% and blue is between 0% and 10%. The more often a sentence is edited the 'hotter' the color.

The analyst can see the typical substitutions for a sentence by hovering the mouse over that sentence. The typical sentences computed as the centroids of the clusters created using Algorithm 2 are themselves color coded using the same identification sys-

[1] We have also experimented with a symmetric version of Levenshtein distance, but we prefer the n-gram overlap score due to its linear run time complexity.

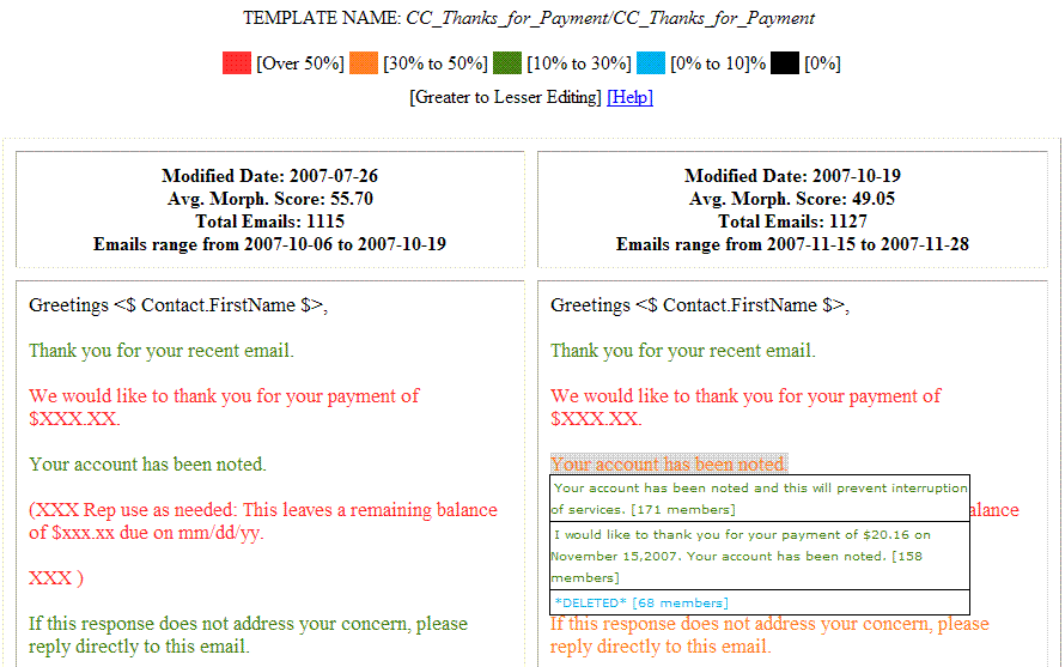

Figure 2: Example of two versions of a template and the edit score (Avg. Morph. Score) and centroids associated with each sentence of the template.

tem. A typical sentence that occurred in over 50% of the emails is colored red. A typical sentence that occurred in 30% to 50% of the emails was orange and so on.

In seeing the two versions side by side, the analyst can visually inspect the agents' edits on the current version of a template relative to the previous version. If the previous version of the template is a 'hotter' document (with more red sentences), it means that the changes made to the template by the author had led to less editing by agents thus speeding up the process of creating a customer response. If the current template looks hotter, it suggests that the changes made to the template were increasing the agents' edits and probably the email handling time.

5 Automatic Extraction of Potential Templates

The goal of the template author is to minimize the number of edits done to a template and thus indirectly lowering the handling time for an email. In the preceding section, we discussed a tool that aids the template authors to identify sentences where changes are most often made by agents to a template. This information could be used by the template authors to create a new version of the template that achieve the goal.

In this section, we investigate a technique that automatically identifies a possible template with the potential of directly minimizing the average handling time for an email. We use the set of responses created by the agents using a given template and select one of the responses to be generalized and stored as a new template. The response to be converted into a template is chosen so as to directly minimize the average handling time. In essence, we seek to partition the set of responses R generated from template T into two clusters R_1 and R_2. These clusters have centroids T (current template) and T' (new template) such that constraint shown in 1 holds.

$$(\forall_{r \in R_1} AHT(T, r) < AHT(T', r)) \wedge (\forall_{r \in R_2} AHT(T', r) < AHT(T, r)) \quad (1)$$

Now, the quantity $AHT(T, r)$ is logged as part of the email management system and corresponds to the time taken to respond to a customer's email.[2]

[2]Although typically this time includes the time to look up a

Cluster	Number of members	Centroid (Template/Response)
1	1799	GREETINGSPHR, Thank you for your recent email. On behalf of the company, I would like to extend my sincere apology for the problems you encountered when (XXX over key with appropriate response XXX). It is our goal to provide excellent customer service, and I am sorry that we did not meet that objective. Your input is very valuable, and we will take your feedback into consideration. Regards, Agent.FirstName
2	206	GREETINGSPHR, Thank you for letting me know that you've been unable to send an online order to upgrade your NAMEDENTITY service. Please accept my apologies for any problems this issue may have caused you. You're a highly valued customer. I understand your concerns and I'll be happy to address them. I am investigating this issue. I have already made a personal commitment to email you tomorrow, with the resolution. Thank you for your patience and for choosing the company. We appreciate your business and continued loyalty. Sincerely, Agent.FirstName

Table 1: Result of clustering responses using the AHT model as the distance metric.

However, we do not have access to $AHT(T', r)$ for any $T' \neq T$. We propose a model to estimate this in the next section.

5.1 Modeling Average Handling Time

We model AHT as a linear combination of several factors which we believe would influence the handling time for an email. These factors include the length in words of the customer's input email ($inplen$), the length in words of the template ($templatelen$), the length in words of the response ($resplen$), the total number of edits between the template and the response ($edit$), the normalized edit score ($nedit$), the number of individual events of the edit distance – substitution (sub), insertion (ins), deletion (del) and identity (id), the number of block (contiguous) substitution ($blksub$), block insertion ($blkins$) and block deletion ($blkdel$). Using these independent variables, we fit a linear regression model using the AHT values for 6175 responses created from one particular template (say G). The result of the regression fit is shown in Equation 2 and the data and error statistics are shown in Table 2. It must be noted that the coefficients for the variables are not necessarily reflective of the importance of the variables, since they compensate for the different ranges in variable values. We have also tried several different regression fits with fewer variables, but find that this fit gives us the best correlation with the data.

$$\begin{aligned} \hat{AHT} &= 0.5314 * inplen - 2.7648 * templatelen \\ &\quad +1.9982 * resplen - 0.5822 * edit \\ &\quad +2900.5242 * nedit \\ &\quad +4.7499 * id - 1.6647 * del \\ &\quad -1.6021 * ins + 26.6704 * blksub \\ &\quad -15.239 * blkins + 24.3931 * blkdel \\ &\quad -261.6627 \end{aligned} \tag{2}$$

Mean AHT	675.74 seconds
Median AHT	543 seconds
Mode AHT	366 seconds
Standard Deviation	487.72 seconds
Correlation coefficient	0.3822
Mean absolute error	320.2 seconds
Root mean squared error	450.64 seconds
Total Number of Instances	6175

Table 2: Data statistics and the goodness of the regression model for 6175 AHT data points.

Based on the goodness statistics of the regression fit, it is clear the AHT model could be improved further. However, we acknowledge that AHT does not depend solely on the editing of a template to a

the customer's account etc., we assume that time is quite similar for all responses created from the same template.

response but involves several other components including the user interface, the complexity of customer's email, the database retrieval to access the customer's account and so forth.

Nevertheless, we use this model to cluster a new set of 2005 responses originating from the same template (G), as shown in Equation 1. Using the k-median clustering as described earlier, we partition the responses into two clusters. We restrict the first cluster centroid to be the template and search for the best centroid for the second cluster. The results are shown in Table 1. The centroid for cluster 1 with 1799 members is the template itself while the centroid for cluster 2 with 206 members is a response that could be suitably generalized to serve as a template. The overall AHT for the 2005 responses using the template was 989.2 seconds, while the average AHT for the members of cluster 1 and 2 was 971.9 seconds and 1140 seconds, indicating that the template had to be edited considerably to create the members of cluster 2.

6 Discussion

For the purposes of this paper, it is assumed that AHT is the same as or correlates well with the time to compose a response for an email. However, in most cases the email care agent might have to perform several verification, validation, and problem resolution phases by consulting the specifics of a customer account before formulating and composing a response. The time taken for each of these phases typically varies depending on the customer's account and the problem category. Nevertheless, we assume that the times for these phases is mostly a constant for a given problem category, and hence the results presented in this paper need to be interpreted on a per problem category basis.

A second limitation of the approach presented in this paper is that the metric used to measure the similarity between strings (n-gram overlap) is only a crude approximation of an ideal semantic similarity metric. There are however other similarity metrics (e.g. BLEU (Papineni et al., 2002)) which could be used equally well. The purpose of this paper is to illustrate the possibility of analysis of responses using one particular instantiation of the similarity metric.

In spite of the several directions that this work can be improved, the system and algorithms described in this paper have been deployed in an operational customer care center. The qualitative feedback we have received are extremely positive and analysts have greatly improved the efficiency of the operation using this tool.

7 Conclusions

In this paper, we have presented two approaches that help template authors in designing effective templates for email customer care agents. In the first approach, we have presented details of a graphical tool that provides vital feedback to the template authors on how their templates are being modified by agents when creating responses. The template authors can accommodate this information when designing the next version of the template. We also presented a novel technique for identifying responses that can potentially serve as templates and reduce AHT. Towards this end, we discussed a method to model AHT based on the characteristics of the customer's email, the template text and the response text.

8 Acknowledgments

We would like to thanks Mazin Gilbert, Junlan Feng, Narendra Gupta and Wenling Hsu for the discussions during the course of this work. We also thank the members who generously offered to their support to provide us with data used in this study without which this work would not have been possible. We thank the anonymous reviewers for their useful suggestions in improving the quality of this paper.

References

S.K. Card, J. Mackinlay, and B. Shneiderman. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.

J. Coch. 1996. Evaluating and comparing three text-production techniques. In *Proceedings of Coling-96*, pages 249–254, Copenhagen, Denmark.

R.O. Duda, P.E. Hart, and D.G. Stork. 2001. *Pattern Classification*. Wiley, New York.

G. Foster, P. Isabelle, and P. Plamondon. 1997. Target text mediated interactive machine translation. *Machine Translation*, 12(1):175–194.

G. Foster, P. Langlais, and G. Lampalme. 2002. User-friendly text prediction for translators. In *EMNLP-02*, pages 46–51, Philadelphia, USA.

Genesys. 2008. http://www.genesys.com. Genesys Corporation.

Kana. 2008. http://www.kana.com. Kana Corporation.
G. Lapalme and L. Kosseim. 2003. Mercure: Towards an automatic e-mail follow-up system. *IEEE Computational Intelligence Bulletin*, 2(1):14–18.
V.I. Levenshtein. 1966. Binary codes capable of correcting deletions, insertion and reversals. *Soviet Physics Doklady*, 10:707–710.
C.D. Martnez-Hinarejos, A. Juan, and F. Casacuberta. 2003. Generalized k-medians clustering for strings. *Lecture Notes in Computer Science*, 2652/2003:502–509.
F.J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *EACL-03*.
K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association of Computational Linguistics*, pages 313–318, Philadelphia, PA, July.
I. Zukerman and Y. Marom. 2007. Evaluation of a large-scale email response system. In *Proceedings of IJCAI07*, Hyderabad, India.