Mercure: Towards an Automatic E-mail Follow-up System

Guy Lapalme[1] and Leila Kosseim[2]

***Abstract*— This paper discusses the design and the approach we have developed in order to deal effectively with customer e-mails sent to a corporation. We first present the current state of the art and then make the point that natural language tools are needed in order to deal effectively with the rather informal style encountered in the e-mails. In our project, called Mercure, we have explored three complementary approaches: classification, case-based reasoning and question-answering.**

***Index Terms*— Customer relationship management, automatic e-mail response, e-mail response management, text classification, case-based reasoning, question-answering**

I. Context of the problem

THE number of free-form electronic documents available and needing to be processed has reached a level that makes the automatic manipulation of natural language a necessity. Manual manipulation is both time-consuming and expensive, making Natural Language Processing (NLP) techniques very attractive. E-mail messages make up a large portion of the free-form documents available today and as e-mail becomes more and more popular, an automated e-mail answering service will become as necessary as an automated telephone service is today.

This paper discusses the use of natural language processing for dealing with e-mail automatically. Our work was developed in the context of e-mails regarding investors relations sent to a specific corporation but we believe that the approach can be applied to any Customer Relationship Management (CRM) application.

Although it is difficult to find reliable figures on the quality of online customer service (because of commercial interests and the fact that these figures are most often given by companies selling CRM systems) the following situation described in [1] seems to be typical:

> A recent Jupiter study[1]of the top 125 web sites found that 55% of customers expect accurate responses to e-mail within 6 hours, yet only 20% of companies are meeting their expectations. Forty-two percent of the sites never responded to the e-mails, took more than five days to respond to the questions, or had no e-mail address listed on their site.

Given the fact that more than half of the people in the US and Canada now have an everyday access to e-mail, it is important for companies to make sure that their clients can use this medium for customer service inquiries. In the context of e-commerce, customers expect more access, continuous support and increased convenience and at the same time, they are less tolerant of poor response time, inaccurate answers or worse, non-responsiveness.

E-mail offers a number of advantages for customers compared to telephone calls: there are no tedious telephone menus and no waiting on the line for an available operator during business hours; with e-mail, the customer can formulate her request any time at her own pace and can continue her normal activities while waiting for the answer. The answer arrives in her usual mailbox and it can be kept for later reference. The customer no longer has to listen carefully to a verbal answer and take the risk of missing or forgetting critical information. However, because there is no immediate feedback between the operator and the customer, the later can never be certain that the request has been received. In addition, interaction between the operator and the customer is much more awkward and slow with e-mail than with a telephone call.

For an enterprise, using e-mail allows it to keep track of communications with its customers either for statistical or quality-control purposes. It is also possible to send more complete and complex instructions by e-mail and to include other media such as pictures, video or audio clips. In addition, it is cheaper to geographically or chronologically distribute e-mail answering to operators. On the other hand, e-mail is much less personal than direct contact with customers.

As described by Walker [25], e-mail should not be considered a substitute for all feedback from customers. In order to figure out *just when e-mail is really the right tool for the job* it is important to study this tool ogether with innovative ways to use it effectively.

II. Current approaches

The simplest level of e-mail answering systems is the so-called *auto-responder*[2]. These systems return a static document in response to an e-mail according to the presence of keywords in the subject or body of the message. As a variant, the user can fill a set of predefined fields in a web form to customize the response. An obvious drawback of these systems is that they do not analyze the content of free-form messages. The content of the text is reduced to a small set of keywords with no regards to the true meaning of the text.

[1] Jupiter Communications, “E-mail Customer Service: Taking control of Rising Customer Demand”, 2000.

[1] RALI, DIRO, Université de Montréal, CP 6128, Succ. Centre Ville, Montréal (Québec) Canada, H3C 3J7 lapalme@iro.umontreal.ca

[2] CLaC Laboratory, Concordia University, 1455 de Maisonneuve Blvd. West, Montréal (Québec) Canada, H3G 1M8
kosseim@cs.concordia.ca

[2] also known as *AR, infobots, mailbots* or *e-mail-on-demand*

More sophisticated types of e-mail responders are included in e-mail management systems, and can provide pre-written response templates for frequently asked questions. Slots are usually filled in with information extracted manually from the incoming mail, although some systems seem to perform the extraction automatically [19].

Some commercial systems such as Kana [17], RightNow [23] or XM-MailMinder [26] are aimed at optimizing the work flow of a call-center by keeping track of customer e-mails, helping representatives to answer by means of partially filled templates and providing productivity statistics on the answering process. However, to our knowledge, these systems do not use any NLP technology outside spell-checking and regular expression matching. Some systems also perform text classification (using learning techniques from annotated corpora or regular expressions) to categorize the incoming message into general pre-defined classes (e.g. requests, congratulations, complaints, . . .). The e-mail can then be routed to the appropriate department or representative or, with specific categories, can even be answered automatically or deleted in the case of *spam*.

An early work on the automatic generation of appropriate answers to customer requests was performed by Coch [9], [10] who developed a system to generate answers to complaint letters from clients of La Redoute (a large French mail-order corporation). As letters were not in electronic format, the reading, the extraction and the decision was performed manually, but the production of a well-formed response was done automatically. Through a formal blind evaluation, Coch demonstrated that the best responses (according to specific criteria) are still the human-generated ones, but that the use of a hybrid template-based Natural Language Generation (NLG) system produced acceptable responses at a much faster rate.

III. MERCURE

Bell Canada Enterprises (BCE) is a large Canadian corporation offering communication and entertainment services such as telephone, internet and television to private and commercial customers. To keep its competitive edge, its customer service must be efficient and cost-effective. In order to achieve this, BCE asked the Bell University Laboratories (BUL) to study the problem of e-mail follow-up in cooperation with the RALI (Recherche Applique en Linguistique Informatique[3]) laboratory. This has resulted in Mercure[4], a 4 year study, also funded by a Cooperative Research and Development grant from the National Science and Engineering Research Council (NSERC) of Canada.

After a preliminary study on a corpus of e-mails dealing with printer related problems [18], we focused on customer e-mails sent to a specific department at BCE: the investors relations department. This department receives and answers e-mails of current and potential investors sent to the address `investors.relations@bce.ca`. The e-mails are often requests for annual reports, press releases, but sometimes contain more complex financial questions such as values of stocks on specific dates, buying and selling plans, explanations about current events of the company; and also regard more routine issues such as address changes, lost of certificate, etc. Although the e-mail service is limited to administrative matters and that no judicial responsibility can be attributed to late or even false answers, timely and exact responses are essential for keeping good relations with investors.

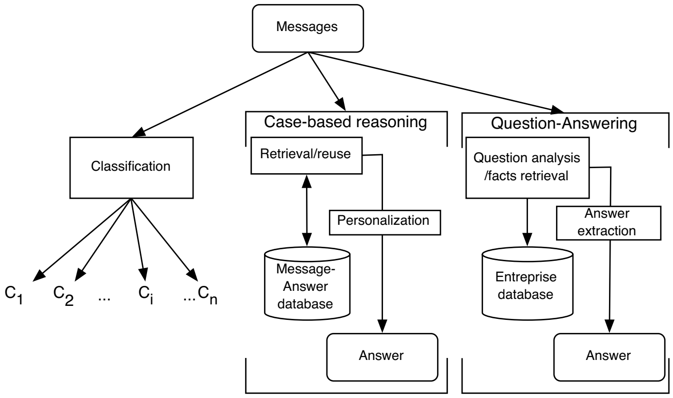

Fig. 6. Modules of the Mercure Project

In order to understand how e-mail is currently dealt with within BCE, we studied a preliminary corpus of more than 1000 e-mails sent to the investors relation department. The analysis showed that the e-mail varied considerably with regards to the level of difficulty required to analyze them: some e-mails were short and asked for a factual answer often found directly in a corporate documentation, while others were quite long and answering them required deeper research and information gathering from various sources. Because of this, we believed that a single technique could not suffice to deal with all e-mails, and we decided to try three complementary techniques in parallel and then to determine which one seems more appropriate given specific e-mail characteristics. Eventually, a combination of these techniques could be used in a real implementation. Figure III shows the three techniques explored in Mercure: text classification, case-based reasoning and question-answering. The following subsections will describe each technique in greater detail.

A. Classification

Classification of documents is a well known problem, but only recently has it been possible to use computers to separate texts into predefined categories according to their contents. The result of classification can be seen as a summary representation of the topic of a set of similar documents in order to ease the finding of related documents. Assigning a document to a certain class is not always a clear cut decision as a document may differ considerably from the others or could be assigned to more than one class. Text classification is typically performed using standard machine learning techniques and information retrieval term weighting schemes. Word distribution is a good feature for discriminating among categories and to classify a new document to its most appropriate category. Although much work has been done on the classification of

[3] Applied Research in Computational Linguistics

[4] French name for Mercury, the roman god who was messenger of the other gods.

TABLE I
RESULTING CLASSIFICATION OF THE 818 SINGLE-PURPOSE MESSAGES OF OUR CORPUS.

Category	%	Description
dividend r.p.	5%	dividend reinvestment plan
stock split	5%	BCE stock split
dividend	5%	other questions about dividends
mailing list	7%	asking to be added or removed from a distribution list
report	17%	asking for annual or trimestrial reports
share price	29%	value of BCE stock
general	32%	other

newspaper articles through techniques such as *K nearest neighbors* [13], *naive Bayes* [15], decision trees such as CART [4] and ID3 [16]. Fewer projects have addressed the problem of e-mail classification [8], [11]. A notable exception is the classification of *spam*, which has attracted some interest in this problem and has even spurred an open-source project [14].

In the context of BCE, a seemingly simple problem is dealing with the intricacies of the contents of e-mail such as headers, citations, attachments, HTML parts, etc. that, in some cases, *hide* the text content and creates noise for the classifier. After removing this *noise*, Dubois [12] managed to extract the content of the e-mails in order to build a corpus of 1568 message and follow-up pairs sent between June 1999 and November 2000 to `investors.relations@bce.ca`. These e-mails were used by Dubois to study many types of classifiers (k nearest neighbors with k=10,20,30,40,50, naive Bayes network and Ripper) on different number of classes (5,10 and 22), with or without preprocessing (numeral and stop word removal or stemming, truncating words or not) and using different separation of corpus between training and validation sets. About 150 configurations have been tested with a success rate of about 50%. The main cause for errors was the noise brought by the fact that some messages dealt with more than one subject or were part of a multi-message exchange. So it was decided to work with only single-topic e-mails. With similar configurations as in the previous case and combination of them (210 in total), results raised to 90% for 5 categories, 80% for 10 categories and 67% for 22 categories. After studying the confusion matrices for all these cases, Dubois finally choosed the 6 categories (plus one *general*) shown in table III-A. With these categories, a success rate of about 80% was obtained on a 144 e- mail test set for March 2002, a period not contained in the learning set.

These results are adequate in the context of Mercure because e-mail of some of these classes (*dividend r.p.* and *mailing list*) are already being forwarded to people outside of BCE. Messages of the *report* category are answered by simply mailing the desired report.

B. Case-Based Reasoning

The second approach we are investigating is the application of textual case-based reasoning (CBR) techniques to generate responses to incoming email messages. This CBR module exploits a corpus of email messages comprising requests from investors and their corresponding responses from financial analysts. Case-based reasoning is similar in spirit to the way humans reuse (and adapt) previous e-mails for answering new requests. The design of a CBR email response system relies on a corpus of previously answered messages, a resource that is representative of the domain of discourse and of the various problems tackled during email exchanges. The *search and adapt* reasoning scheme then offers a natural mapping to the two phases of email response, i.e. the analysis of incoming requests and the synthesis of relevant responses. Presented from a client perspective, the CBR module attempts to reuse messages in the SENT mailbox of the analyst's email software to suggest responses to new messages incoming in the INBOX. Our processing is divided into three main phases (retrieval of cases, reuse of cases and personalization of the answer). Each step is now described below and has been implemented in a prototype Java-based mail client.

1) Retrieval of cases: This phase compares a new message with the ones previously received, in order to find a similar one and reuse its answer. During our initial experimentation, the similarity between messages was established based on the comparison of a tf.idf (term frequency × inverse document frequency) vectorial representation of the message content. Using a cosine function to compute global similarity provides a precision of approximately 57.9%. This is similar to the results of comparable experiments with FAQs [7]. However, the nature of our cases can be exploited to improve some aspects of the retrieval phase. As the selection of wrong answers requires additional manipulation by the user of the system, it is important to optimize the ranking of the most relevant(s) case(s) to ensure the production of a relevant response.

For improving the performance of the retrieval phase, we first considered the classical word relationships but it required an exact correspondence of words (or key-phrases or ngrams). To overcome this constraint, some authors [6], [7] have made use of existing linguistic resources (e.g. thesaurus) to establish the semantic similarity of different words that have related meanings. This approach does not transpose well to our problem as, to our knowledge, no domain specific resources are available.

Since textual responses provided by a limited number of analysts are more similar (based on word distribution) than requests sent by many different investors, we conjectured that similarity should be more easily established when the textual responses are also taken into account during the retrieval phase. We combined both of the above possibilities into a single scheme. A textual case can be seen as the linguistic *conversion* of a textual problem into a corresponding textual solution. The case base then corresponds to a mapping from a *request* language (problem) to a *response* language (solution). The finding of associations, captured as co-occurrences, provides indications that the occurrence of problem words increases the likelihood of the presence of some other words in the solution. To obtain the co-occurences, we collect the count of all pairs of words coming respectively from the requests and their corresponding responses, and we select the most significant ones based on the mutual information metric [21].

The approach we are currently using of inserting the associations in the retrieval phase is inspired from query expansion techniques. The incoming problem description (the investor's request) is expanded into a vector of response terms provided by the lists of co-occurrences. Similarity of the cases then corresponds to the weighted sum of both problem and solution vector cosine. Experimentation [20] conducted on 102 test requests indicates that the expansion scheme slightly improves the overall precision (62.0% vs. 57.9%) of the retrieval phase and preserves the rank of the first pertinent solution in the similarity list (2.01 vs. 1.96). The most significant improvement has been observed for the test messages where the response is not directly addressing the request (e.g. redirection to a generic web site address following the request of specific documents or financial information). For this category of message, the precision is almost doubled (80.1% vs. 51.0%) and the average rank is reduced to a very good level (1.33 vs. 2.38). For the other messages, the precision is mostly preserved but we observed some degradation for the routine messages as the expansion scheme introduces some noise in the internal representation of the textual cases. This result is however interesting as responses are built from a limited number of the most highly ranked cases (usually the first one). And, most importantly, we expect that the selection of a judicious trade-off between request and solution similarities will bring further improvement.

2) Reuse of previous cases: Our application presents strong incentives to implement some adaptations of previous responses. While complete reformulation of past textual responses for diverse situations is beyond the capability of current CBR and NLP techniques, some of these techniques can nevertheless help to personalize past messages and preserve the relevance of cases with the context of the new incoming request. In the CBR literature, case adaptation (i.e. case reuse) has exclusively been conducted for structural cases and mostly corresponds to modifying the values of pre-selected solution features. In a textual setting like our email response domain, such a scheme is rather difficult to implement, as the textual solutions are not structured. Therefore, prior to the modification of the content of the messages, we need to determine what portions of the responses are good candidates for modification. Given a new message and some past solutions selected during the retrieval phase, we have implemented the reuse of textual cases as a three-step process:

1) identification of passages for determining the text portions that are applicable in the context of the new incoming request. Statistical distributions, captured as word alignments [5], can be used for this task;
2) message personalization that determines what text portions are to be modified;
3) pruning and substitution for removal of irrelevant passages and the substitutions of the portions to be personalized. In NLP, this corresponds to a query-relevant summarization process [3], more specifically to the condensation of a text based on the terms of a request.

3) Personalization of the messages: Personalization of messages refers to the capacity to detect some factual information in the messages and to substitute them in the responses. This includes, for instance, names of companies, individuals, financial factors, dates and time references. These expressions correspond to named entities and can be identified using information extraction techniques (IE). IE techniques identify, using either rule patterns or statistical models, information from textual documents to be converted into a template-based representation. As we did during the first phase of the project, we make use of extraction patterns and lexicons (lists of company names, titles, acronyms and frequent financial terms).

Substitutions of these entities are partly conducted using a rule-based approach. Replacement of individual names and companies is based on the roles of the messages entities. The role is determined by the type of patterns used during extraction, mostly based on the part-of-speech and the terms preceding/following the entities. For instance, expressions like *"Sincerely, John Smith"*, *"to purchase Nortel shares"*, *"registered with Montreal Trust"*, could provide indications of the message sender, subsidiary company and financial institution respectively. However, as the Investor Relations domain does not offer much predictability, the elicitation of domain rules for numeric information (dates, price, factors?) remains difficult and such substitutions rely mostly on the user.

C. Question-Answering

Many of the e-mails sent to corporations are asking for information and can be considered as questions from customers to which representatives should answer in the best possible way. The third technique used is based on Question-Answering (QA) technology: the task of finding an exact answer to a natural language question [24] in a large set of documents. The question type is determined by the presence of trigger phrases (e.g. *where*, *how many*, *how much*), which indicate the type of the answer required (e.g. *location, number, money*). Information retrieval is typically performed to identify a subset of the documents and a set of passages that may contain the answer. Named entities are then extracted from these passages and semantically tagged and the string containing the best scoring entity is retained as the answer. Within Mercure, we have developed Quantum [22], a *traditional* QA system with which we participated in the QA-track of TREC and that will be used as a basis for our work in e-mail answering.

QA differs from e-mail answering in several aspects. Generally speaking, e-mail answering involves *analyzing* a longer text and *formulating* a linguistically-motivated answer, while QA takes a short and explicit question as input and focuses on *locating* the answer. Issues in discourse analysis and generation must therefore be addressed in e-mail answering, but not in QA. In addition, questions, at least in systems participating to the TREC evaluations, are restricted to specific types such as *who, why, where, ...* but pertain to an unrestricted discourse domain. On the other hand, in e-mail answering, the questions are of unrestricted type, but the discourse domain is typically restricted. E-mail answering thus involves finding passages from the textual knowledge base that best relate to the incoming message and sending the passages as is to the user. This is the avenue currently being pursued by Luc Blanger[2] in his Ph.D. thesis.

IV. Transfer to the industry

In order to make sure that the technology we developed in our lab could be transferred to the operational context of BCE, we installed a mirror mail server with the same hardware and software configuration as the one used by BCE. We also made arrangements to receive a copy of all e-mails sent to investors relations at BCE and this enabled us to build a dynamic corpus of e-mails which was used for testing: these new e-mails deal with the same domains as the ones used for developing the system. A version of the classifier has been installed in the BCE mail server but administrative delays and change of personnel did not allow a complete integration into the answering process. The CBR and Question-Answering modules are being developed separately and will eventually be integrated into the mail server.

V. Conclusion and future work

In this paper we have described the research conducted within the Mercure project, aimed at the automatic follow-up of e-mail messages. The work was performed specifically with a corpus of e-mails from the investors relations department of Bell Canada Enterprises. As the e-mails were not homogeneous in their textual characteristics, we explored three complementary approaches: text classification, case-based reasoning and question-answering. Our experience with e-mail classification was not very fruitful. As the classes considered were very much related, the standard word distribution approach showed insufficient discrimination power. However, it would be interesting to compare our results with human classification to have an upper bound measure of what we can hope to achieve. This would allow us to evaluate whether the approach needs to be modified or if the task is simply too difficult. The 2 other approaches are still under development. The case-based reasoning module seems promising and the research performed so far seems to show that an important number of messages can be answered using this technique. Finally, the question-answering approach still needs more work, especially to identify the question in the texts.

Once the case-based reasoning and the question-answering modules are in place, we plan to evaluate each approach on different sets of e-mails so as to measure how appropriate each approach is as a function of specific e-mail characteristics such as e-mail length, category, etc. This will allow us to combine the three approaches either by running then in parallel and combining their result, or by using one approach and revert to another if the previous one is unable to produce an appropriate answer with enough confidence.

Acknowledgments

The work described in this article is a joint work with the following graduate students: Stéphane Beauregard, Luc Plamondon, Luc Lamontagne, Luc Bélanger and Julien Dubois. We are grateful to our colleagues from RALI for their relevant comments. This work was supported by a grant from Bell University Laboratories and a Cooperative Research and Development grant from NSERC.

References

[1] Banter Inc. Natural language engines for advanced customer interaction, 2001. `http://www.realmarket.com/required062801.html`.

[2] Luc Bélanger. Le traitement automatisé des courriels pour les services aux investisseurs: une approche par la question-réponse. Technical report, Département d'informatique et RO - Université de Montréal, 2003.

[3] A. Berger and V. Mittal. Query-relevant summarization using FAQs. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 294–301, Hong-Kong, 2000.

[4] L. Breiman, J.H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth Int. Group, 1984.

[5] P. Brown, J. Cocke, S. Della Pietra, V. Della Pietra, R. Jelinek, F.and Mercer, and P. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.

[6] S. Brüninghaus and K. Ashley. Bootstrapping case base development with annotated case summaries. In *Proceedings of ICCBR-99, Lecture Notes in Computer Science 1650*, pages 59–73. Springer Verlag, 1999.

[7] R. Burke, K. Hammond, V. Kulyukin, S. Lytinen, N. Tomuro, and S. Schoenberg. Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System. *AI Magazine*, 18(2):57–66, 1997.

[8] S. Busemann, S. Schmeier, and R. Arens. Message Classification in the Call Center. In *Proceedings of ANLP-2000*, pages 159–165, Seattle, 2000.

[9] J. Coch. Evaluating and comparing three text-production techniques. In *Proceedings of COLING-96*, Copenhagen, Dannemark, 1996.

[10] J. Coch and J. Magnoler. Quality tests for a mail generation system. In *Proceedings of Linguistic Engineering*, Montpellier, France, 1995.

[11] W. Cohen. Learning rules that classify e-mail. In *Proceeding of the 1996 AAAI Spring Symposium on Machine Learning in Information Access*, 1996.

[12] Julien Dubois. Classification automatique de courrier électronique. Master's thesis, Université de Montréal, 2002.

[13] R. O. Duda and P.E. Hart. *Pattern classification and scene analysis*. John Wiley & Sons, 1973.

[14] John Graham-Cumming. Popfile - automatic email classification, 2003.

[15] T. Joachims. A probabilistic analysis of the rocchio algorithm with tfidf for text categorisation. In D.H. Fisher, editor, *Proceedings of the 14th International Conference on Machine Learning*, pages 143–151, San Francisco, 1997. Morgan Kaufmann.

[16] Quinlan J.R. Induction on decision trees. *Machine Learning*, 1(1):81–106, 1986.

[17] www.kana.com. 1999.

[18] Leila Kosseim, Stéphane Beauregard, and Guy Lapalme. Using information extraction and natural language generation to answer e-mail. *Data & Knowledge Engineering*, 38:85–100, 2001.

[19] Y. Lallement and M. Fox. Interact: A Staged Approach to Customer Service Automation. In H. Hamilton and Q. Yang, editors, *Canadian AI 2000*, LNAI 1822, pages 164–175, Berlin, 2000. Springler-Verlag.

[20] Luc Lamontagne, Philippe Langlais, and Guy Lapalme. Using statistical word associations for the retrieval of strongly-textual cases. In *Florida Artificial Intelligence Research Science (FLAIRS) 2003*, page 7 pages, St-Augustine, Florida, 2003.

[21] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[22] Luc Plamondon, Leila Kosseim, and Guy Lapalme. The QUANTUM question answering system at trec-11. In E.M. Vorhees and D.K. Harman, editors, *The Eleventh Text REtrieval Conference (TREC-2002)*, pages 670–677, Gaithersburg, MD, November 2002. NIST.

[23] RightNow Technologies. Revelation Knowledge Engine, 2002. `http://www.rightnow.com`.

[24] *Proceedings of the Tenth Text REtrieval Conference (TREC-X)*, Gaithersburg, Maryland, 2001.

[25] David Walker. Automation woes widen the email expectations gap. `http://www.shorewalker.com/pages/email_expectations-1.html`.

[26] XtraMind Technologies GmbH. XM-MailMinder, 2002. `http://www.xtramind.com`.