Textual Reuse for Email Response

Luc Lamontagne[1,2] and Guy Lapalme[2]

[1]Département d'informatique et génie logiciel,
Université Laval, Québec, QC, Canada
luc.lamontagne@ift.ulaval.ca

[2]Département d'informatique et recherche opérationnelle,
Université de Montréal, Montréal, QC, Canada
lapalme@iro.umontreal.ca

Abstract. The case-based reasoning approach to email response consists of reusing past messages to synthesize new responses to incoming requests. This task presents various challenges due to the nature of the messages: Textual descriptions, multiple topics, heterogeneous content, variable text length and varying recurrence of the statements. In this paper, we address the problem of determining which portions of past cases are reusable. Our scheme consists of identifying parts of a past message and declaring them variable, optional or reusable. This formulation of case reuse corresponds, from an application point of view, to the dynamic creation of a response template from antecedent messages. We describe and compare two strategies for selecting the messages portions to be reused: Case grouping and condensation models. Our results indicate that the case grouping strategy is a better choice. We also describe some of our experiments for identifying variable parts, based on named entity extraction techniques.

1 Introduction

Contrary to structural case-based reasoning (CBR) approaches that offer numerous strategies for adapting structured cases, the reuse of textual solutions remains mainly an unexplored research topic in CBR. This situation can be explained by the nature of the work in textual CBR mostly dedicated to retrieval tasks [1], [2], and to its applications to tasks such as legal jurisprudence [3], [4], a domain that does not require the modification of solutions descriptions.

Nonetheless, many tasks requiring that new descriptions be written could benefit from a capacity to adapt the textual solutions content. An example of such a task is the response to email exchanges. Many organizations face the problem of managing the response to a large volume of incoming requests. Tools to support the writing of recurrent responses offer many advantages and could be easily integrated into current email client software. A response is defined as a sequence of statements satisfying the content of a given request. To be reused in a different context, a response requires some personalization and the adjustment of specific information.

In this paper, we study and evaluate an approach to reuse past solutions when the content is textual. The reuse process consists of two parts: Determining the portions from past responses that could be reused and identifying how to adapt these portions. Most of the reuse approaches in structural CBR consist of modifying the feature values of well-structured solutions. Furthermore, these features are determined in advance. In a textual setting such as email response, this scheme is difficult to implement because the solutions are unstructured and because the portions of the response to be modified cannot be determined a priori since they will differ depending on the new incoming request. Hence, a first step is to determine the basic units of text to process, their pertinence and their specificity.

In our application, the cases consisting of requests (problems) and responses (solutions) messages are short separate textual descriptions. Email messages present some particular characteristics that make them difficult to reuse. First, they are usually heterogeneous and contain multiple topics. Their writing and grammatical style can present some weaknesses, which makes syntactic approaches difficult to use. Contrary to texts written for official usage (e.g. news reports, legal documents), their content does not present any specific structure or rhetorical forms.

This paper is organized as follows: In section 2, we give a brief overview of our CBR approach to email response. We describe in section 3 the reuse approach methodology that we have developed. In sections 4 and 5 we present two strategies based on case grouping and case condensation. Section 6 contains some experimental results indicating that the case grouping strategy is superior to the condensation strategy. We propose some ideas for future work in section 7 and conclude in section 8.

2 Overview of the LUG Approach to Email Response

Our work was conducted as part of a project to study the applicability of natural language processing techniques to email response [5]. Various potential approaches were identified from current commercial systems and from the current NLP literature:

- *Static text*: Systems, such as autoresponders, send pre-written messages to respond to new requests. The system associates these messages to the presence of an email address in the header or keywords in the body of a message. Each message received by the response system can trigger rules to select and send predefined and completely specified responses. This approach offers little flexibility and requires that most situations be anticipated in advance.
- *Structured requests*: Another approach is the mandatory use of forms accessible from a web server in order to constrain the content of the requests. The different sections of the form bring the user to better describe the purpose of his/her request. The requests generated by these web sites consist of a mixture of keywords, attribute-value pairs and some predefined textual formulations. This structuring facilitates the processing of the requests but does not propose a way to formulate new responses.
- *Response templates:* Templates are patterns made available to the writer to help in the formulation of a new response. A template is a form that dictates the

structure of the response message. Some systems only insert a header to new messages. Others propose pre-specified responses with sections that can be modified or removed by the user. While they are inexpensive and rapid to deploy, these systems present some limitations since the number of possible situations must be determined in advance. Templates must be written for each of these situations which might not be feasible for evolving and complex domains. They require therefore constant human intervention for the creation or modification of the patterns.

- *Free-text generation*: One may consider using text generation approaches combined with techniques for the understanding of incoming requests. Unfortunately, such systems would be far too complex since they must rely on the linguistic generation of the messages and make use of NLP techniques to manage communicational, semantic, syntactic and lexical aspects. Considerable effort would be required for the construction of grammars. At the present time, few resources are available to implement such systems and the efficiency of these approaches would depend on a good understanding of the incoming requests, another difficult problem to tackle.

Our CBR approach to email response consists mainly of two steps [6] :

- A past case is selected from the case base as a basis to build a new response (i.e. the retrieval phase). The case base contains {request, response} pairs corresponding to the problems and solutions of our application.
- Modifications to the solution part of the case (the response) are proposed as a function of the new incoming request, to help adjust the content (the reuse phase).

In order to support the retrieval phase, we exploit word associations between the requests and their responses. This scheme takes advantage of the homogeneity of responses and helps improve the precision of the system. We refer the reader to [7] for additional details on the use of word co-occurrences and translation models to implement this approach.

After selecting a case, our CBR module proposes to the user a response description annotated as follows:

- some regions indicate the portions of text deemed optional that could be pruned by the writer.
- some regions indicate that some specific information may be modified by the user to take into account the context of the new request.

An example of how a response is annotated is presented in Figure 1. The final decision regarding the modification or the withdrawal of the textual passages is the writer's responsibility. Hence, the purpose of this scheme is not to automatically reshape a structured text but rather to guide the user in identifying the portions that should be modified. From a CBR point of view, the system is responsible for the reuse phase and leaves the case revision up to the user.

By producing a text containing reuse annotations, we borrow from the approaches based on response templates. However, our approach has the following important features that make it more attractive. The positions of the gaps to be filled out, i.e., the response annotations, are chosen dynamically and depend on the reuse potential of the text with respect to the request. Hence, each new response created by the writer

can become a new template to be reused for the processing of subsequent requests. This means that the patterns do not have to be created manually. Furthermore, their integration in the case base increases the number of future situations that can be addressed while avoiding substantial modifications of the system.

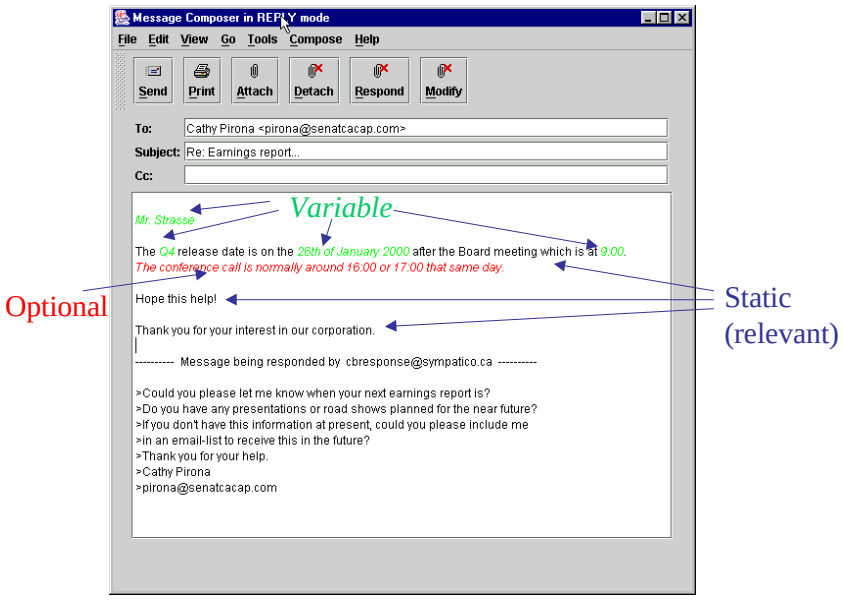

Fig. 1. Recommendations on the reuse of a past response

The insertion of « *optional* » and *?variable* annotations corresponds to a text generalization. As illustrated in part (a) of Fig. 2, a light generalization will present a smaller subset of the passages to be modified, hence making the selection of the passages clearer to the writer. On the other hand, a generalization of the courtesy sentences makes it more difficult to select the reusable portions of the text (Fig 2(b)). This example illustrates the need to avoid aggressive strategies for annotating the text passages.

Dear *?PERSON_NAME*
« The year ended on ?DATE »
The release date for the next earnings report is on *?DATE*.
Please, do not hesitate to contact us for any other questions.
Sincerely…

(a)

« Dear ?PERSON_NAME »
« The year ended on ?DATE »
The release date for the next earnings report is on *?DATE*.
« Please, do not hesitate to contact us for any other questions. »
« Sincerely… »

(b)

Fig. 2. Generalization of a past response: (a) generalization of some passages, and (b) generalization of courtesy sentences

3 The Reuse Scheme

The sequence of statements in a solution (a response) is meant to satisfy the sequence of statements of a problem (a request). When the context of the problem is modified, some of the statements become irrelevant while some others become erroneous. While a complete restructuring of the solutions can not be considered with current NLP techniques, some approaches can help to:

- Preserve the relevance of cases with respect to the context of a new problem ;
- Ensure that the descriptions are adequately specified.

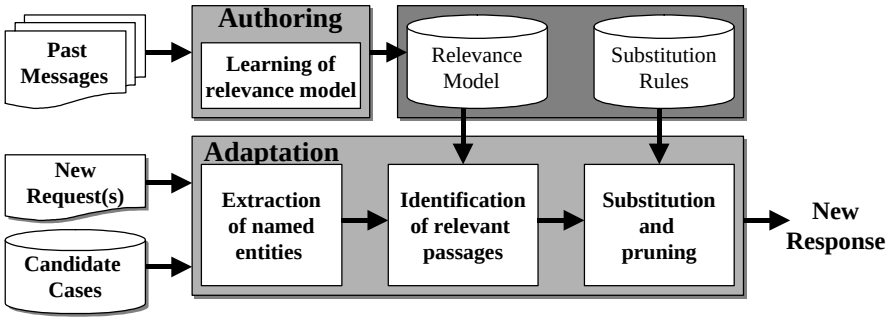

Fig. 3. Steps of the reuse process of textual solutions

Given a new request and a past response selected during the retrieval phase, we implement the reuse of textual cases as a three step process (Figure 3):

1. *Identification of optional passages:* This step consists of determining the textual portions that are applicable to the context of the new request. We start by segmenting the past responses in passages, more specifically in individual sentences. Then, the sentences are evaluated and the best subset is selected. The pertinence is established with respect to the content of the new request. This evaluation is the basis for rendering the relevant passages static (i.e. recommending them to the user) and making the rest of the solution description optional (i.e. inviting the user to review these passages). The description of this approach is presented in Section 4 of this paper.
2. *Modification of the specific content:* Among the relevant sentences, the next step is to identify which portions of text are subject to being modified. At this point of the reuse scheme, we identify the different portions that might present discrepancies with the context of the request. These portions usually refer to information such as individuals, locations, and addresses that will vary according to the context or temporal references. This information often takes the form of named entities and their pertinence is established as a function of the new request. The usage of information extraction techniques [8] for this step is presented in Section 5.
3. *Pruning and substitution:* The withdrawal of irrelevant portions and the substitution of portions to be specified are made in this step. Although mainly manual, we discuss some of these aspects in Sections 4 and 5.

The process presented above has many advantages. By reusing valid responses, no syntactic processing is required and we are able to control text uniformity, quality and fluency. By inheriting the content of past messages, we avoid the efforts devoted to content planning. By limiting the variable portions to factual information contained in named entities, we avoid surface generation problems (morphology, punctuation, genre, and agreement…). Although past responses usually contain few sentences (usually less than 10), it is still faster to find them automatically (a few milliseconds) than asking the user to select them manually and to cut and paste them in the new response message.

4 Identification of Optional Portions

The identification of optional portions makes it possible to reorganize the content of an antecedent response by presenting the superfluous parts. By declaring sentences to be optional (or static), we ensure that the response content will adequately cover the content of the new request.

Passage granularity in terms of individual terms, syntactic groups, sub-sequences of words, etc., will vary according to the application domain. In our application, the relevance of the statements in a solution relies on the sentence as the basic unit. This favours the coherence and the intelligibility of the subset resulting from the pruning process. We assume that a statement corresponds to a sentence and that this statement pertains to a single theme. Nevertheless, this choice is not a critical issue for the application of the techniques we propose.

In order to find the sentences of the response that best cover the new request, we execute the three following tasks:

- Segmentation: We break the past responses into individual sentences. The software we used in our experimentation (*lmtag*) provides a tagging of the beginning of the sentences and paragraphs;
- Evaluation of relevance : We estimate the relevance of each individual sentence with respect to the content of the request;
- Selection: We choose the sentences that seem the most promising and present them to the user as static (i.e. no highlighting). The others are presented as optional (highlighted using various colors).

To identify the static/optional sentences, we must first establish a correspondence between the statements found in the solutions and problems. Relationships between words can contribute to establish some correspondence between a request and a response. However, relationships are weak or absent from *accessory* sentences such as greetings, courtesy forms and general information. While these sentences are not essential, they play an important role in the narrative form of the solution and they should ideally be preserved when the do not contradict the context of the request.

We study and compare two strategies for the evaluation and selection phases:

– We evaluate each sentence individually and we select those that obtain a satisfactory support from the content of the request. To evaluate a sentence coming from a past solution, we identify the cases that confirm or reject the correspondence between a target sentence and a request. The similarity between

the various cases in the case base indicates whether the sentence should be selected or not. We present this approach in Section 4.1.

– The second strategy is to select a subset of the sentences that best covers the content of the request. This processing of the relevance at the sentences group level corresponds to a reduction of the text. This type of summary is frequently referred to as *query-biased* [9]. We present this strategy in Section 4.2.

Our goal is to preserve the sentences of the response that obtain a sufficient support from the request. To determine this support, the case base is used to model the knowledge necessary to apply both strategies. The base contains different examples that establish a correspondence between problem and solution descriptions.

4.1 Case Grouping Strategy

Our first strategy is to determine whether each individual sentence should be kept in the solution proposed to the user. For each sentence $sent_j$ of an antecedent solution we are reusing, we identify from the case base of the CBR module the cases $Cases_{support}$ that comprise one or more statements similar to $sent_j$ and the cases $Cases_{reject}$ which do not contain it. This corresponds to determining, given a new problem *P* (a request) and some pairs <*problem*, *solution*> from the case base, whether the solution recommended to the user should contain the target sentence $sent_j$ (Figure 4).

```
Case_Grouping_Select(sent_j, P, CB)
        cases_support := Supporting_Cases(sent_j, CB)
        cases_reject := CB - cases_support
        R ←  Similarity( Centroid(cases_support), P ) >
              Similarity( Centroid(cases_reject), P )
        return R

Supporting_Cases(sent_j, CB)
        R := {}
        for each case c of CB
            s :=  solution(c);
            if Contains(sol, sent_j)
                R := R + problem(c);
        return R

where Contains is implemented as an Overlap metric, Similarity
is a cosine function, and Centroid as a weighted sum of term
vectors.
```

Fig. 4. Recommendation algorithm for including a sentence in the reused solution using a case grouping strategy

As illustrated in Figure 5, we partition our case base into two groups used to determine the content of the problems supporting the usage of a specific sentence in the response. We then create a distribution of the requests that characterizes the sets $Cases_{support}$ and $Cases_{reject}$. By interpolating between these distributions and the new request, we determine the membership of the target sentence to the solution.

The membership of a target sentence to a solution (i.e., the predicate *Contains*) is estimated according to the similarity between the target sentence and each of the sentences of a solution. In our work, we evaluate the similarity between sentences of solutions by an *Overlap* metric, i.e.

$$Contains_{Overlap}\left(sent_{target}, solution\right) = \max_{sent_i \in solution}\left(\frac{\left|sent_{target} \cap sent_i\right|}{\min\left(\left|sent_{target}\right|, \left|sent_i\right|\right)}\right)$$

This metric estimates the proportion of words that the sentences have in common. A statistical similarity metric gives good results for our domain solutions since these are highly homogeneous. We have observed in our corpus that the users tend to cut and paste portions of past responses, resulting in few variations among similar statements. Other metrics based either on domain or linguistic resources could be useful for messages from application domains presenting less uniformity in the statements.

All the cases with a value $Contains_{Overlap}$ superior to a given threshold are associated to the set $Cases_{support}$ while others are associated to the group $Cases_{reject}$. Some experiments helped in choosing empirically a threshold value for the similarity between sentences. Since they do not depend on the content of the new request, these sets can be authored during the construction of the CBR module.

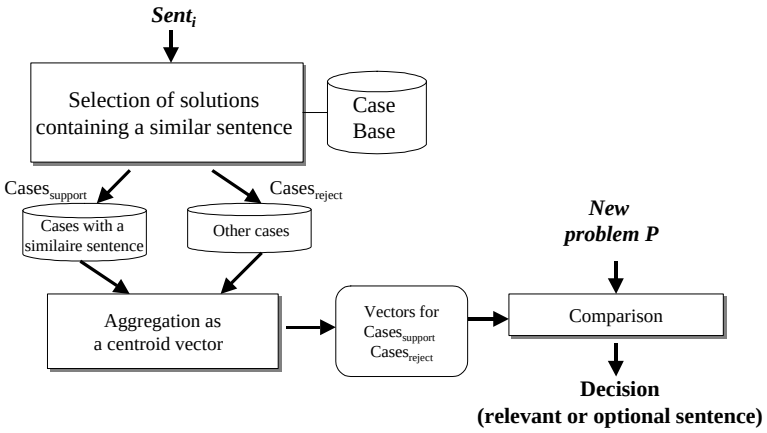

Fig. 5. Partitioning of the case base and similarity of a new problem with the partitions

The two groups obtained by the partitioning of the case base characterize the problems that favour or reject the usage of a sentence similar to $sent_{target}$. By estimating the proximity of the new problem P with the cases of the two groups, we can interpolate the correspondence between P and $sent_{target}$. To estimate their proximity, we represent each of the groups by a structure that merges the vectorial representations of the problem descriptions after the problems terms have been lemmatized and filtered according to the vocabulary of the CBR module. We compute the centroid of each group from the term frequency vectors of the requests. The similarity between the request P and the centroid of each group is determined by a cosine of the two vectors. We select the sentence (i.e. make it static) if the similarity of the request with $Cases_{support}$ is the greatest, i.e.

$similarity(new\_request, Cases_{support}) > similarity(new\_request, Cases_{reject})$

If this inequality is not verified or if the case base does not contain a solution with a similar statement, then the sentence is deemed optional.

4.2 Condensation Strategy

The second strategy that we study is not based on the evaluation of each individual sentence but on the global quality of a subset of sentences selected from a reused solution. The presence of irrelevant passages is mostly due to the occurrence of multiple themes in the requests and solutions. The identification of these passages corresponds to the production of a subset of antecedent responses that covers most of the context of the new request. In natural language processing, this is often referred to as a *query-biased* or *user-centered* summarization process. More specifically, it corresponds to the production of a condensed text based on the terms of the request. In this variation, a request indicates the focus of the user (what is being looked for) and the portions of text that are found in the summary should be in agreement with the statements of the request.

As illustrated in Figure 6, the resulting solution S_c can be produced by the deletion, from the original solution S, of sentences (or text portions like noun phrases) that can be associated (or *aligned*) to the new request Q.

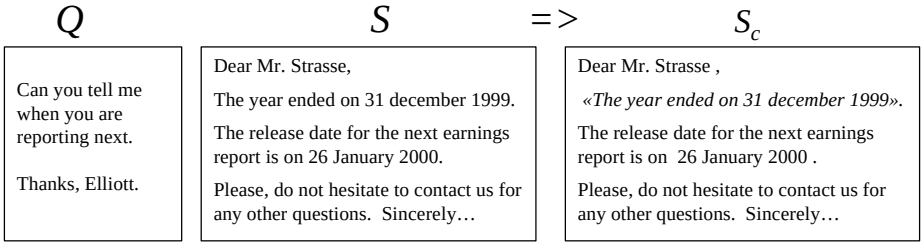

Fig. 6. Identification of relevant passages by a condensation process

As proposed by Mittal and Berger [9], this matching process tries to determine a subset S_c that covers most of the request Q. In terms of probability, we are trying to find a condensed response S' that maximize the following probability estimate:

$$S_c = f(Q,S) = \arg\max_{S'} P(S'|S,Q)$$

Using Bayes rule, this expression can be approximated as follows:

$$\begin{aligned} S_c &= \arg\max_{S'} P(S'|S,Q) \\ &= \arg\max_{S'} P(Q|S',S)P(S'|S) \\ &\sim \arg\max_{S'} P(Q|S')P(S'|S) \end{aligned}$$

Hence, this formulation suggests that the text being recommended to the user (i.e. the static text) is a compromise between the integrity of the past response and a subset of the response that best fits the new request.

The expression *P(S'|S)* can be modeled as a random withdrawal of terms from the original request *Q*. Some probability distributions (for instance multinomial or hypergeometric) allow the evaluation of the resulting condensate. In our work, we model the distribution *P(S'|S)* by a multinomial distribution.

$$P(S'|S) = \frac{\prod_{i \in S} tf_i! \times \left(tf_i / N \right)^{c(i \in S')}}{N!}$$

where tf_i is the frequency of the term *i* in response, *c* is the number of occurrences of term *i* in the condensate *S'*, and *N* is the number of terms in the response. Since the responses are relatively short and since most of the terms appear only once in each description, we can approximate this distribution by:

$$P(S'|S) = \frac{1}{|S|! \times |S|^{|S'|}}$$

Therefore a severe reduction will diminish the fidelity of the recommended solution. This indirectly quantifies the textual support provided to the user.

The expression *P(Q|S')* corresponds to the probability that a new request *Q* is at the origin of a response *S'*. We modeled this distribution as an IBM1 model [10] obtained during our work on the retrieval phase. We exploit the case base of the CBR module to learn the distribution of the model. Some parameters of the model are obtained during the training. To assign a probability to missing values (or values deemed insignificant by the learning process), we smooth the distribution using a backoff formulation, i.e.

$$p(q_i | s_j) = \begin{cases} t(q_i | s_j) & \text{the value of the transfer probability} \\ a p_{CB}(q_i) & \text{otherwise} \end{cases}$$

where *t* is the transfer model obtained during the training and p_{CB} is the distribution of terms in the case base (i.e. in our corpus of email messages).

5 Identification of the Variable Portions

For our application, we have observed that most modifications necessary to reuse antecedent messages rely on specific information like phone numbers, company names, dates, and so on. These information items refer to named entities and can be obtained using information extraction techniques. Hence, an adequate extraction of these entities combined with a modeling of their role will capture most of the substitution information. This part of our application is domain dependent and was constructed manually based on an analysis of our corpus.

The three steps to process the variable portions are the extraction of named entities, the assignment of roles and the substitution decisions.

a) Named entity extraction: we have identified the following domain entities that offer a potential for reuse:

- Dates: Specific dates (e.g. *Jan/01*, *Tuesday May 12*, *tomorrow*, *coming year*), time periods (e.g. *last ten years*, *past 6 months*, *first quarter*) and temporal references (e.g. 16h00, 9:00AM, 2:00EST).
- Persons: Combinations of names, initials and titles (e.g. *Mr. P. J. Smith*).
- Organizations: Proper names not designating individuals. Many messages contain keywords such as *Capital*, *Corporation*, *Associates*, *Bank*, *Trust*, *Inc*, *Department*,
- Addresses: Some URLs, email addresses and civic numbers.
- Locations: Names & acronyms of countries, states, regions and cities (e.g. *Canada*, *Boston*, *Thames Valley*, *USA*, *UK*, *NY*).
- Quantities: *Currency*, real, integers, fractions, percentages.
- Phone numbers: Most of these are in North American format.

Most of these entities were obtained using Gate [11], an information extraction system with predefined rules for extracting named entities. We also used regular expressions to capture some information specific to our domain. We are therefore able to obtain solutions annotated according to the preceding categories.

b) Role assignment: The entity categories give an estimation of the portions with potential for substitution. However, their role in the application domain (in our case, investor relations) must first be defined prior to taking a decision on their reuse value. To define the role of an entity, we take into account its category, its type (e.g. a date of type *time*), the character strings it might contains, and the context defined by the words either preceding or following it. For instance, the role "*conference_time*" is defined as a date of type *time* preceded either by the words *conference* or *call*.

c) Entities substitution: The investor relations domain presents low predictability on how to recommend substitution values for the named entities. We considered three substitution cases:

- A role is never modified: Some roles are invariant for the domain (e.g. the name of the main corporation) and some others can not be determined based on the context of the problem (names of locations). Also, some roles occur only in the problem descriptions (e.g. names of newspapers, personal URLs, employers names).
- The value of a role can be extracted from the request: The substitution value can be obtained from entities present in the content of the problem. For instance, the name of the investor that submitted the request or the fiscal year pertaining to the discussion could follow this substitution pattern.
- The value of the role can be modified if declared in the CBR module: For these roles, a value can not be located or inferred from the context of the problem. This role remains invariant for a given period of time and a recommendation could be made if its value is declared in a lookup table or by some other persistent means. Most of the entities of our application domain are of this type (e.g. financial factors, dates, temporal references, names of documents, web site addresses…).

By restricting the selection of substitution values with respect to the role of the entities, the efficiency of our approach relies mostly on the capacity of the CBR

module to extract the named entities and to assign an adequate role to them. We evaluate these two capacities in the following section.

6 Some Experimental Results

For this experimentation, we used a corpus pertaining to the Investor Relations domain (i.e. the assistance provided by enterprises to their individual and corporate investors). The messages cover a variety of topics such as requests for documents, financial results, stock market behaviour and corporate events. We worked with 102 messages after having removed the headers and signatures. The length of the textual body parts of these messages varies from a few to over 200 words with an average of 87 words. The courtesy and accessory sentences were kept in order to evaluate their influence on the reuse process.

a) Results for the selection of optional portions: Our first experimentation was to evaluate, with the two strategies proposed in Section 4, the pertinence of each response of our corpus with respect to their corresponding requests. We performed a leave-one-in evaluation (i.e. we left the target problem in the case base) and estimated the accuracy by the proportion of sentences declared relevant by the algorithm. We obtained an accuracy of 89% for the case grouping strategy and 77% for the condensation strategy. These results are superior to a random strategy (i.e. an average accuracy of 50%). Since the target cases were present in the case base, we assume that these results are upper bounds for system performance. At this point, we note a major difference between the two strategies. The condensation strategy tends to drop most of the accessory sentences while case grouping tends to preserve them. The condensation approach is hence more conservative.

To obtain a more representative estimation of the performance of the reuse module, we selected a sample of 50 pairs of <request, response> messages obtained through our retrieval module. For these pairs, we manually determined the subset of sentences that should be selected by the CBR system. This was made possible since we have responses provided by financial analysts for each of the requests. However, we found it difficult to determine whether accessory sentences (for instance, the courtesy sentences) should be included or not in the reused message. In order to take this into account, we produced two sets of results where these sentences are either required or not. The results are presented in Tables **1** and **2**.

Table 1. Selection of relevant portions with accessory sentences

Strategy	Precision	Recall	*Accuracy*
Case grouping	84.1%	68.0%	70.8%
Condensation	78.4%	39.6%	50.2%

Table 2. Selection of relevant portions without accessory sentences

Strategy	Precision	Recall	*Accuracy*
Case grouping	77.7%	76.0%	71.8%
Condensation	78.5%	53.1%	62.2%

The case grouping strategy selects most of the sentences pertaining to the request. The results in both tables indicate that it preserves most of the accessory sentences. Some sentences are rejected because they are too widely spread in the case base which makes it difficult to decide whether their usage is appropriate.

The condensation approach presents a totally different behaviour. Many words contained in the sentences are infrequent and can not be associated with some other words of the requests. Moreover, almost all of the accessory sentences are rejected, since they have no statistical associations with request words. This explains why recall figures increase when accessory sentences are not taken into account (Table 2).

b) Results for the selection of the variable portions: We retained 130 sentences from our corpus of responses that contains over 250 named entities. We conducted the entity extraction and role assignment on these sentences. The results we obtained are presented in the Table 3

Table 3. Results for the extraction of entities and the assignment of roles

Entity	*Entity extraction*		*Role Assignment*
	Precision	*Recall*	*Accuracy*
Date	91.7%	85.6%	82.9%
Time	100%	100%	61.1%
Location	71.4%	93.5%	66.6%
Person	100.0%	80.0%	81.8%
Quantity	92.2%	95.6%	68.7%
Organization[1]	97.2%	83.3%	94.4%
Phone number	95.4	90.9%	81.8%

We note that the extraction of most of the categories give good results (precision and recall columns of the table). For instance, the few errors for dates are references to financial quarters (e.g. *Q4*, *4th quarter*). Also some company names like *Bell Canada* are annotated by Gate as a combination of a name and a location. Such errors can easily be removed by augmenting the lexicon and extraction patterns of the system.

We manually constructed a rule base for the subset of our original corpus and we assigned roles to the entities of the 130 sentences of our test corpus. The entities were initially assigned to their true category. The results indicate that the global accuracy of the role assignment is approximately 76.7%. We estimate that such a result is satisfying given the simplicity of the rules that we constructed. For some entities, it is sometimes difficult to establish their role based on a single sentence. Other times, coreference limits sentence interpretation. For instance, various meanings can be assigned to the temporal reference “It will be at 17:00”. However, most of the errors were due to role descriptions that we had not anticipated while constructing our rule base.

[1] The term “BCE” account for more than half of the organization entities occuring in our test corpus. If we remove these, we obtain a precision of 95.5% and a recall of 68.3%.

7 Related Work

In order to position our work with respect to adaptation techniques used in structural CBR, we remark that our scheme offers both substitutional and transformational components for the reuse of antecedent solutions. Recommending which specific passages should be modified corresponds to parametric variations found in substitutional approaches. Moreover, the identification of optional passages leads to the pruning of some statements, which corresponds to a transformation of the response structure. Since our reuse scheme relies on a single message, our approach is not compositional nor do we consider a complete reformulation of the solutions as performed by generative approaches. Because the user of the response system supervises pruning and substitution of the passages, our approach addresses the problem of case reuse and leaves the revision of solutions to the user.

Some substitutional methods to acquire knowledge for the adaptation of structured cases were proposed [12], [13], [14]. Our approach differs from these methods since it is a transformational algorithm that learn term distributions and translations models instead of rules or cases. Furthermore, our adaptation process is driven by the solutions to be reused (i.e. the pertinence of their sentences and their named entities) while other approaches rely on a comparison of the features of problems descriptions.

8 Conclusion

In this paper, we presented a CBR approach to reuse antecedent responses to respond to new requests. We proposed two strategies to select relevant portions of antecedent messages. We observed that case condensation is a conservative selection strategy. We recommend using a case grouping strategy that offers better performance in terms of precision and recall. Case condensation could also be a useful alternative for applications built from a large case base. We also explored the use of techniques for named entity extraction in order to determine the variable parts of a response. The efficiency of this step relies mainly on the availability of tools to locate the entities. The results we obtained indicate that the identification of roles, once the entities are extracted, is rather simple to implement with rules based on regular expressions.

To our knowledge, our work represents a first attempt for textual case adaptation and it brings up numerous directions in which this research could be pursued. We believe that the idea of dynamically created templates is a metaphor sufficiently generic to be applied to other contexts than email response. It preserves the narrative form of the solutions and overcomes the limitations of the generative approaches that, in a textual setting, are difficult to achieve. We have chosen, for this work, to concentrate our efforts on the reuse of a single case. However, a compositional approach, which takes into account multiple cases, would offer a better covering of the various themes occurring in a request. The reuse of multiple cases could be based on voting schemes to select messages portions. Another issue related to the multi-case reuse is the identification of variable passages that could be conducted by comparing the statements of the solutions and lead to the selection of passages based on syntactic and/or semantic features. This would overcome the main limitations of

our work where the roles of the domain entities are manually defined. Finally, the case grouping strategy could be extended so that the two case groups providing positive or negative support may be used to learn categorization rules.

References

1. Lamontagne L., Lapalme G., 2002. Raisonnement à base de cas textuels – état de l'art et perspectives, *Revue d'Intelligence Artificielle*, Hermes, Paris, vol. 16, no. 3, pp. 339-366.
2. Lenz M., 1998; Textual CBR, in Lenz M., Bartsch-Spörl B., Burkhard H.-D., Wess S. (Eds.), *Case-Based Reasoning Technology - From Foundations to Applications*, Lecture Notes in Artificial Intelligence 1400, Springer Verlag.
3. Weber R., Martins A., Barcia R, 1998. On legal texts and cases, *Textual Case-Based Reasoning: Papers from the AAAI-98 Workshop*, Rapport technique WS-98-12, AAAI Press, pp. 40-50.
4. Brüninghaus S., Ashley K. D., 1999. Bootstrapping Case Base Development with Annotated Case Summaries, *Proceedings of the Third International Conference on Case-Based Reasoning (ICCBR-99)*, Lecture Note in Computer Science 1650, Springer Verlag, pp. 59-73.
5. Lapalme G., Kosseim L., 2003. Mercure : Toward an automatic e-mail follow-up system, *IEEE Computational Intelligence Bulletin*, vol. 2, no. 1, p. 14-18.
6. Lamontagne, L. ; Lapalme, G. ; 2003 "Applying Case-Based Reasoning to Email Response", Proceedings of ICEIS-03, Angers, France, pp. 115-123.
7. Lamontagne L., Langlais, P., Lapalme, G., 2003, Using Statistical Models for the Retrieval of Fully-Textual Cases, in Russell. I, Haller, S. (Editors), Proceedings of FLAIRS-2003, AAAI Press, Ste-Augustine, Florida, pp.124-128.
8. Cowie, J., Lehnert, W., 1996. Information Extraction, Communications of the ACM, vol. 39 (1), pp. 80-91.
9. Mittal, V., Berger, A., 2000. Query-relevant summarization using FAQs,. In Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL). Hong Kong.
10. Brown, P. F.; Della Pietra, S. A.; Della Pietra, V. J.; and Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation, Computational Linguistics, vol. 19, no. 2, pp. 263-311.
11. Cunningham H., Maynard D., Bontcheva K., Tablan V., 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, pp. 168-175.
12. Hanney K., Keane M., 1997. The Adaption Knowledge Bottleneck: How to Ease it by Learning from Cases, *Proceedings of the Second International Conference on Case-Based Reasonining (ICCBR-97)*, Springer Verlag, pp. 359-370.
13. Jarmulak J., Craw S., Rowe R., 2001. Using Case-Base Data to Learn Adaptation Knowledge for Design, *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, Morgan Kaufmann, pp. 1011-1016.
14. Leake D. B., Kinley, A., and Wilson D., 1996. Acquiring Case Adaptation Knowledge: A Hybrid Approach, *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, AAAI Press, Menlo Park, CA, pp. 684-689.