The Role of Natural Language Generation During Adaptation in Textual CBR

Pablo Gervás, Raquel Hervás, and Juan Antonio Recio-García

Departamento de Ingeniería del Software e Inteligencia Artificial, Universidad Complutense de Madrid
c/ Prof. José García Santesmases s/n, 28040 Madrid, Spain
pgervas@sip.ucm.es, raquelhb@fdi.ucm.es, jareciog@fdi.ucm.es

Abstract. The adaptation stage of a CBR cycle implies somehow constructing a modified version of an original case to match a new situation. When CBR operates over textual cases, adapting a case to a new situation may involve producing new text, or adapting the original text by making so many modifications to its content as to merit a complete redrafting. This constitutes an important obstacle on the road to providing textual CBR systems with adaptation functionality. Natural language can be extremely complex. Fortunately, it is not always necessary to be able to model its full complexity to achieve interesting results for specific tasks. There are many NLP solutions available - such as information extraction - which by means of shallow processing allow access to approximate versions of the information embodied in a text. Such solutions provide easy means for implementing interesting retrieval process for textual CBR. However, the problem of case adaptation usually requires a more advanced level of modelling of the complexity of language. This paper outlines some of the difficulties involved, and puts forward Natural Language Generation (NLG) as a possible candidate for addressing them.

Keywords. Case Based Reasoning, Textual CBR, Adaptation, Natural Language Generation

1 Introduction

The process of adaptation in Textual CBR requires a flexible and complete representation of the cases that can be reused to generate a new text. A basic approach to this representation of the texts may include both semantic and lexical information. Semantic information is used to adapt the case to a new situation and lexical information is required to generate the adapted text. Such information can be obtained by applying Information Extraction (IE) and Natural Language Processing (NLP) techniques to obtain the semantic and lexical information [4,11]. A number of existing solutions to this problem are described in section 2.1.

This approach has been shown to provide acceptable solutions in situations where very little adaptation of the available cases is needed. However, as soon

as significant changes are required during adaptation, it results in texts of a relatively poor quality, easily identifiable as machine-produced. Several factors contribute to this problem. Text involves several layers of linguistic information in addition to its semantics and its lexicon. An important ingredient is syntax. A simple approach would be to account for the syntactic information of a text by using templates - predefined strings corresponding to generic linguistic constructions, with convenient slots to be filled in for a particular use - to render the semantic information, using the lexical information to fill in the gaps. Another option is to reuse the structure of the original case, restricting adaptation to introducing new lexical items - corresponding to the desired semantics - in place of the old ones. Such solutions might have worked if it were not for two additional levels of information involved in the linguistic form of a text: morphology and pragmatics. *Morphology* deals with how words are inflected depending on the role they play within a sentence. Nouns may have singular or plural forms, pronouns change depending on number, person, and case - whether they act as subjects or objects of the verb -, verbs change depending on tense, mood. *Pragmatics* models the way in which humans optimise text to ensure that no unnecessary information is mentioned. It is concerned with problems such as ensuring that pronominal references allow easy identification of the intended referent, or identifying which portions of the available information may be omitted because they can be inferred from the context. When simplistic approaches attempt to reconstruct text without taking these aspects into account, the quality of the output text tends to be poor. This quality might have been improved if the basic rules in these two fields had been taken into account. This is due to the fact that a slight change in the context for a given fragment of text may trigger a chain of changes in the form of many of its words. By reusing only semantic and lexical information, a system runs the risk of producing sentences like "Jane buys Jane's car in 2006", where the desired output may have been "Jane bought her car last year". Humans innately identify the problems in these examples, usually without being aware of particular rules to explain them. But unless a system is provided with explicit rules, it will stumble on this kind of problem when trying to produce text.

This is where natural language generation may be of help. As a model of the way humans go about the task of generating text, natural language generation covers most of these aspects in working implementations. A brief overview of the relevant ideas in the field is provided in section 2.2. The description given focuses on the particular areas of natural language generation that may be relevant to the task in hand. The rest of the paper puts forward the idea that a combination of existing approaches to textual CBR, and existing solutions for natural language generation, might together cover enough of the problems involved to provide acceptable solutions. Section 2 presents briefly the elements that may be combined, and introduces examples of similar combinations of CBR and natural language generation that have proved fruitful in the past for other problems. Section 3 presents a proposal for a hybrid system that uses classic information extraction textual CBR processes for case retrieval and natural language gener-

ation techniques during case adaptation. Section 4 discusses the advantages and disadvantages of such a proposal, and outlines conclusions and future work.

2 Relevant Previous Work

This section outlines three basic ingredients on which the proposal in this paper is built: the processes used to extract conceptual information from text in existing textual CBR systems, the functionalities covered by a classic natural language generator, and existing efforts to link CBR and NLG.

2.1 Textual CBR: Accessing the Information in Text

In the Textual CBR literature there are several approaches to extract and organize the information contained in the cases (see [17] for a review). But, commonly, these systems are built ad-hoc and it is not easy to reuse them in different domains. To solve this problem, there are generic IE/NLP libraries that can be used to extract the information contained in the cases. These libraries (or frameworks) are divided into several layers or steps that gradually process the text: stemmers, part-of-speech taggers, sentence detectors, synonym detection using Wordnet, ... A common factor in these libraries is the last step where rules are used to extract the information into a structured representation. Some well known examples are the GATE library[1] or the OpenNLP package[2] that include several implementations for each step. These steps were also defined and organized by Lenz [10] in a layered model suitable for Textual-CBR.

There are two important features to analyze in these architectures: the rules used to extract the information and the way it is represented. The IE process is domain dependent so it is commonly implemented through specific rules for each application. GATE defines its own rules language and Lenz's model leaves this task unspecified. [13] describes a restaurant recommender system that follows the Lenz approach and uses regular expressions to define rules that obtain a slot-based representation of the cases. At this point we arrive to the other important feature: the organization and representation of the extracted information.

GATE uses an approach based on tables that contain the described piece of text and the information associated to that segment. OpenNLP creates an XML tree over the text adding the information into the tags of this XML description. Evolving these representations we find the approaches developed in the Semantic Web research field based on ontologies. There is a clear analogy between Textual-CBR and the Semantic Web where both fields try to obtain a representation of the texts (or web pages) that allows to reason with them. In TCBR we try to reuse this information to solve a new problem and in the Semantic Web they perform "semantic retrievals" over the web [16,7,8].

Ontologies allow to represent both semantic and lexical information about a text and bring many advantages to the Textual-CBR process. They can be

[1] http://gate.ac.uk/
[2] http://opennlp.sourceforge.net/

reused through applications easing the domain knowledge acquisition of the CBR applications [14]. Also, there are several reasoners available that can be applied in the adaptation step. And finally, it is easy to modify the IE rules used by GATE or the Lenz model to tag a text with a given ontology.

2.2 NLG: Modeling the Way Text is Put Together

Natural Language Generation (NLG) is a subfield of Artificial Intelligence and Computational Linguistics that covers the design and construction of systems that produce text in human languages. Natural language generation is currently accepted to operate as a pipeline of stages [15], each enriching[3] a conceptual input with the linguistic information required for realizing it as text in a given context. The initial stages of the pipeline, known as *content planning*, are concerned with filtering and grouping into messages - elementary units of information that can be expressed as sentences - the content that is to be rendered as text.

Of particular interest in the present context is the task of sentence planning. *Sentence planning* is in charge of deciding how to refer to the concepts appearing in these messages (*referring expression generation*), which specific words to use to express them (*lexicalisation*), which information may be omitted because it can be inferred from the context and how the resulting linguistic elements may be grouped together to achieve natural and compact text (these last two tasks are known collectively as *aggregation*). Sentence planning basically models issues largely relating to pragmatics.

The final stage of a natural language generator is called *surface realization*, and it embodies all decisions related to the grammar and morphology of a particular language. This stage is known to be knowledge-hungry, requiring explicit formulation of linguistic details for the intended output language, but largely domain independent. This has made possible the existence of a number of reusable modules for dealing with this task, such as KPML [1] or FUF [5].

With respect to the formats employed for knowledge representation, NLG has been caught up in the recent trend towards standardization. Although similar levels of abstract semantic organization are now being sought in many natural language systems, they are often built anew for each project, are to an unnecessary extent domain or theory specific, are required to fulfill an ill-determined set of functionalities, and lack criteria for their design. The Penman Upper Model [3] was a linguistically motivated ontology developed at the Information Sciences Institute in the late 1980s for mediating between domain knowledge and a natural language generation system. It is a hierarchy of concepts that captures semantic distinctions necessary for generating natural language. The Generalized Upper Model [2] is a descendent of the Penman Upper Model. It is a general task and domain independent 'linguistically motivated ontology' intended for organizing information for expression in natural language. The categories of the ontology

[3] The actual operations involved may also consist of trimming the input, or masking certain parts of it so that they do not appear explicitly in the final text.

enforce a consistent modelling style on any domain which is also guaranteedly appropriate for flexible expression in natural language.

2.3 CBR and NLG: A Symbiosis Already Proved Fruitful

The combination of CBR and NLG has already been employed to solve problems that evaded more conventional solutions. A classic problem in natural language generation is the "generation gap" described by Meteer [12], a discrepancy between what can be expressed in the text plan and what the particular realization solution can actually convert into text. This is particularly apparent in template-based generators, which have recently achieved widespread acceptance. Template-based solutions for natural language generation rely on reusing fragments of text extracted from typical texts in a given domain, having applied to them a process which identifies the part of them which is common to all uses, and leaving certain gaps to be filled with details corresponding to a new use. However, the fact that templates are made up of words that are not accessible to the system makes the system blind to the possible ways of combining them.

Following this trend of using templates for textual CBR solutions, Lamontagne and Lapalme [9] presented an approach to email response by reusing past messages to synthesize new responses to incoming requests. The reuse process consisted of two parts: determining the portions from past responses that could be reused and identifying how to adapt these portions. However, Lamontagne and Lapalme did not rely on natural language generation solutions, but only in the structures of the texts from the case base.

CBR has also been fruitfully applied to address this problem in [6] by considering sentences in a text as instances of already solved lexicalization problems, where particular linguistic constructions have been used to convey certain conceptual relations that hold between a set of concepts - the referents mentioned in the sentence. This solution involves the use of a vocabulary for actions or verbs stored in the form of cases, where each case stores not only the corresponding template but also additional information concerning the type of case, the elements involved in the action, and the role that those elements play in the action. A case is not an abstract instance of a verb or action, but rather a concrete instance in which specific characters, places and objects appear. These elements are stored in the module's knowledge base. This allows the establishment of relations between them when it comes to retrieving or reusing the cases.

An example of a case and the associated template is given below. It shows the structure of the case in the form of attribute value pairs, which corresponds to the query for which a lexicalization is sought, and the associated solution for the case in the form of a text template, with slots to be filled by the lexical realizations assigned to the values of the attributes (*witch* and *Hansel*).

```
TYPE:   LEX:     ACTOR:  OBJECT:
FIGHT   attack   witch   Hansel
```

_ *attacked* _

This solution uses CBR to improve the functionality of an existing NLG system in two different ways. On the one hand, it reduces significantly the need to have explicit templates for all possible situations, because the system can adapt - and/or combine - existing templates to express new ideas. On the other hand, it provides an easy way to restrict the lexicalization options for output text to those that are similar to a given corpus, that which was used to build the case base. These two improvements are particularly relevant for the present proposal, as is discussed in section 3.3.

3 Combining CBR and Natural Language Generation

It seems apparent that most existing textual CBR systems take text as input but actually operate on some internal representation of the cases which is either slot-based or structured in some way. This particular representation must satisfy the constraints imposed on it by the need to succesfully carry out two basic tasks involved in the reasoning process: the task of computing similarity between case and query, and the task of carrying out valid modifications to the content of a retrieved case to match a particular query. An important part of the functionality of existing textual CBR systems is concerned with the extraction of such a structured representation from the text. Another important task is the reasoning involved in constructing a valid solution from the retrieved case or set of cases. To address this problem, representations based on ontologies have been used. These representations fulfill the requirements of being structured and the task of defining similarity measures can be supported with actual inference if a logic-based ontology language is employed. This allows complex reasoning during adaptation.

The problem for including an adaptation phase in textual CBR systems lies in how to revert back into text the structure that results after it has been transformed into a possible solution to the input query. Natural language generation provides a possible way to solve this final step. Natural language generation systems are especially designed to take as input any kind of conceptual representation, and to produce natural and coherent text from it. As a first approximation, it would be enough to connect a simple pipeline, carrying out only sentence planning and surface realization, to the output of the textual CBR system. Such a module would be in charge of converting the structured representation corresponding to an individual message - the information that can be expressed as a single sentence - into a valid sentence. Additionally, if more than one such sentence is required to provide the final answer, the sentence planning stage may carry out the task of ensuring that each such sentence follows on naturally from the previous ones, using correct pronominal references or omitting information already available in the context.

A full description of all the tasks involved is beyond the scope of this paper, which aims simply to put forward the combination as a potential solution. But providing some detail for at least one task may help to clarify the advantages.

3.1 One Stage Described in Detail: Referring Expression Generation

The appropriate use of referring expressions to compete with human-generated texts involves a certain difficulty. According to Reiter and Dale [15], a referring expression must communicate enough information to identify univocally the intended referent within the context of the current discourse, but always avoiding unnecessary or redundant modifiers. When looking for a reference for a specific concept in the text, it is possible to decide between using a pronoun, the plain name of the concept, its proper noun (if any), a description using its attributes, a description using its relations with other concepts, etc. The range of choice depends directly on the available knowledge.

When dealing with automatic generation of referring expressions, there are some elements that are required - or at least desirable - in some kind of knowledge base containing the information about the discourse domain. Every entity appearing in the text should be characterised in terms of a collection of attributes and their values, being one of them its type. Following this trend, the knowledge base may organise some attribute values and types as a subsumption hierarchy. These assumptions are clearly satisfied if a description logic ontology is used for this purpose. Entities would correspond to instances of concepts of the ontology, the attribute corresponding to the type would be the concept of which they are immediate instances, and the taxonomical structure of the ontology of concepts would provide the subsumption hierarchy.

The input of the reference generation module is a structured discourse containing messages for all the information that is to be conveyed, and set in the order in which they should be told. In messages obtained in this way, referents appear in terms of the identifiers of some element in the domain ontology. Different mentions of the same element in different messages about it all share a pointer to a single copy of that element. Before text can be generated for those messages, these mentions of referents must be replaced by specific referential expressions or *references*.

The process of selecting which information to use when referring to an element should take into account basic principles of economy of discourse: try not to use more information than is necessary at each stage to successfully identify the correct element. The process of telling apart a given element at a particular mention must ensure that it avoids possible confusion with other elements presented nearby. To achieve this, a partial version of the context is built incrementally in the knowledge base as the discourse is processed. The system can query this partial context to ascertain whether the information under consideration is enough to not confuse the reader. For example, in a text describing a room with a chair, a comprehensible reference for the chair would be simply "the chair" or "it" if the previous sentence in the discourse is about this chair ("the chair is red", for example). However, if there are more than one chair, the reference must include information to distinguish between the two chairs in order to be understandable. For example, if they are painted with different colours, the reference "the red chair" would not lead to confusion and would be correct.

3.2 Synchronising Knowledge Representations

The connection between a textual CBR system and a natural language generation system should not be difficult to make. The stages of a natural language generation pipeline concerned with interpreting conceptual input are known to be highly domain-dependent, and they usually need to be rewritten for each particular application. Connecting a natural language generation module to the output stage of a textual CBR adaptation stage could be as easy as writing the appropriate code for converting the internal structured representation of the CBR system to whatever internal representation format the NLG system uses. Moreover, natural language generation is following the general trend of relying more and more on ontologies for knowledge representation. The first and second authors of this paper are currently working on the development of a natural language generation system that takes input in the form of OWL ontologies, and relies on underlying domain models also represented as OWL ontologies. This would greatly simplify the task of connecting with a textual CBR system whose internal representation relied on OWL ontologies - such as the one mentioned in [13] based on the jCOLIBRI framework.

3.3 Corpus-Based Lexicalization

The choice of which lexical items are used to refer to each concept in the solution would be made by a lexicalization stage. These modules usually rely on a dictionary or look up table which lists assignments between concepts and lexical items. This type of solution requires explicit construction of domain-specific resources which capture the particular style of lexical choice desired in any given application. An important disadvantage may be that changing to a different case base may imply a drastic redevelopment of the corresponding lexicon. By using a CBR lexicalization stage of the kind described in [6] this difficulty may be significantly reduced. A change in the case base to employ a different corpus of texts would simply involve substituting the lexicalization case base for another one built using the new corpus. This would also ensure that whatever output is generated is composed not only relying on the semantic content of the cases in the case base, but also lexicalised *in the style of* the cases, in linguistic terms.

3.4 Extensions of the Concept of Context

The sketch given above would correspond to a solution that applied natural language generation techniques to convert into text the output of a textual CBR system, but specifically constrained to dealing with such output in terms of individual sentences. Such a solution would be appropriate if the output expected from the textual CBR system were sentence-sized fragments of text. In most cases, the texts required as output would concern not isolated sentences but larger units such as paragraphs describing a particular object - as might be the case in the restaurant recommender system described in [13] - or full reports as might be the result of the specific workshop challenge. Under these slightly

different circumstances, the addition of a sentence planning module might not be enough to guarantee a natural and coherent output text. Human texts usually have a high level structure that organises the material based on complex relations that hold between the concepts involved - such as causality, or chronological order, but also rhetorical relations such as explanation, elaboration... -, and based on the expectations of the reader when faced with particular genres of text. This type of consideration is also addressed by natural language generation systems, in the initial stages known as content planning. The extension of the present proposal to deal with these issues would also be feasible, but it may imply the implementation of domain specific modules for content planning, and possibly the additional encoding of specific content-planning knowledge for the particular domain involved.

4 Conclusions and Future Work

By construction, natural language generation systems are designed to address appropriately all the difficulties faced by textual CBR systems when trying to include an adaptation stage - as listed in section 1. Sections 3.1 and 3.3 have sketched brief descriptions of how the resources already available to a textual CBR system might be benefitially employed in providing complex solutions to some of the problems at little development cost. All concerns regarding grammar and morphology would largely be addressed by reusing an existing surface realization module.

It is worth mentioning that in this proposal the actual adaptation process would not be carried out by the natural language generation system. The textual CBR system would need to have an implementation of the adaptation process, possibly along the lines of using multiple cases for adaptation or applying knowledge-intensive CBR to support the process with complex inference. The natural language generation process would be concerned exclusively with converting the result of the textual CBR system - expressed in the same structured representation used for its internal processing - back to a text that might be judged reasonably natural and coherent by a human reader.

The present paper is not intended as a description of an existing system, not even a system under development, but rather as the proposal of a possible research line, which shows significant promise and the authors consider worth exploring. This is done in the spirit of the workshop challenge, and we hope it will give rise to interesting comments from the community. The authors do have the intention of carrying this proposal further unless significant obstacles arise in the process.

Acknowledgements

This research is funded by the Spanish Ministry of Education and Science (TIN2006-14433-C02-01 and TIN2006-15140-C03-02 projects), Complutense Uni-

versity of Madrid and the G.D. of Universities and Research of the Community of Madrid (UCM-CAM-910494 research group grant).

References

1. J. A. Bateman. Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering*, 3(1):15–55, 1997.
2. J. A. Bateman, R. Henschel, and F. Rinaldi. Generalized Upper Model 2.0: documentation, 1995.
3. J. A. Bateman, R. T. Kasper, J. D. Moore, and R. A. Whitney. A General Organization of Knowledge for Natural Language Processing: the PENMAN upper model, 1990.
4. S. Brüninghaus and K. D. Ashley. The Role of Information Extraction for Textual CBR. In *ICCBR '01: Proceedings of the 4th International Conference on Case-Based Reasoning*, pages 74–89, London, UK, 2001. Springer-Verlag.
5. M. Elhadad. FUF: The universal unifier. user manual, version 5.2. Technical Report CUCS-038-91, Columbia University, 1993.
6. R. Hervás and P. Gervás. Case-based reasoning for knowledge-intensive template selection during text generation. In *Proc. of the 8th European Conference on Case-Based Reasoning*. Springer-Verlag, 2006.
7. E. H. Hovy. Toward Large-Scale Shallow Semantics for Higher-Quality NLP. In *ESWC*, page 2, 2006.
8. L. Khan, D. McLeod, and E. Hovy. Retrieval effectiveness of an ontology-based model for information selection. *The VLDB Journal*, 13(1):71–85, 2004.
9. L. Lamontagne and G. Lapalme. Textual reuse for email response. In *Proc. of the 7th European Conference on Case-Based Reasoning (ECCBR 04)*, pages 242–256. Springer-Verlag, LNAI 3155.
10. M. Lenz. Defining knowledge layers for textual case-based reasoning. In *EWCBR*, pages 298–309, 1998.
11. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.
12. M. W. Meteer. *The generation gap: the problem of expressibility in text planning*. PhD thesis, Amherst, MA, USA, 1990.
13. J. A. Recio-García, B. Díaz-Agudo, M. A. Gómez-Martín, and N. Wiratunga. Extending jCOLIBRI for Textual CBR. In *Proceedings of the Sixth edition of the International Conference on Case-Based Reasoning*, 2005.
14. J. A. Recio-García, B. Díaz-Agudo, P. A. González-Calero, and A. Sánchez-Ruiz-Granados. Ontology based CBR with jCOLIBRI. In R. Ellis, T. Allen, and A. Tuson, editors, *Applications and Innovations in Intelligent Systems XIV. Proceedings of AI-2006, the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 149–162, Cambridge, United Kingdom, December 2006. Springer.
15. E. Reiter and R. Dale. *Building Natural Language Generation Systems*. Cambridge University Press, 2000.
16. D. Vallet, M. Fernández, and P. Castells. An ontology-based information retrieval model. In *ESWC*, pages 455–470, 2005.
17. R. O. Weber, K. D. Ashley, and S. Brüninghaus. Textual case-based reasoning. *The Knowledge Engineering Review*, 20(03):255–260, 2006.