Novice-Friendly Natural Language Generation Template Authoring Environment

Maria Fernanda Caropreso[1], Diana Inkpen[1], Shahzad Khan[2] and Fazel Keshtkar[1]

[1]School of Information Technology and Engineering, University of Ottawa
{caropres,diana,akeshtka}@site.uottawa.ca
[2]DISTIL Interactive
s.khan2@distilinteractive.com

Abstract. Natural Language Generation (NLG) systems can make data accessible in an easily digestible textual form; but using such systems requires sophisticated linguistic and sometimes even programming knowledge. We have designed and implemented an environment for creating and modifying NLG templates that requires no programming knowledge, and can operate with a minimum of linguistic knowledge. It allows specifying templates with any number of variables and dependencies between them. It internally uses SimpleNLG to provide the linguistic background knowledge. We test the performance of our system in the context of an interactive simulation game.

1. Introduction

Natural Language Generation (NLG) is the process of constructing outputs from non-linguistic inputs [1]. NLG is useful in systems in which textual interaction with the users is required, as for example Gaming, Robotics, and Automatic Help Desks.

However, the use of the available NLG systems is far from simple. The most complete systems often require extensive linguistic knowledge, as in the case of the KMLP system [2]. A simpler system, SimpleNLG [6], requires Java programming knowledge. This knowledge cannot be assumed for the content and subject matter experts who develop eLearning systems and serious games. However, these individuals do need to interact with the NLG system in order to make use of the message generation capability to support their product development efforts. It is then necessary to provide them with an environment that will allow them to have access in a simpler way to the features they need of a specific NLG system.

We present an environment that provides simple access to the use of SimpleNLG in order to generate sentences with variable parts or templates. We developed this NLG Template Authoring Environment guided by the need of templates required for generating content in the ISO 14001[1] game developed by DISTIL Interactive[2]. The goal of this project was to provide the game content designers with an accessible tool

[1] The ISO 14001 game's formal title is 'Business in Balance: Implementing an Environmental Management System'
[2] http://www.distilinteractive.com/

they could use to create and manipulate the NLG templates, and thus generate sentences that would support the narrative progression of the game.

In the rest of this paper we first introduce general concepts of NLG and some of the tools available. We then introduce Serious Games (or training games) and their need for NLG. With this we motivate the developing of our NLG Template Authoring Environment. We then describe its design, implementation, evaluation and capabilities to cover different aspects of the templates. We finish the paper presenting our conclusions and what is pending as future work.

2. Natural Language Generation and SimpleNLG

There are two widely adopted approaches to NLG, the 'deep-linguistic' and the 'template-based' [4]. The deep-linguistic approach attempts to build the sentences up from a wholly logical representation. An example of this type of system is KMPL [2]. The template-based NLG systems provide scaffolding in the form of templates that contain a predefined structure and some of the final text. A commonly quoted example is the Forecast Generator (FOG) system designed to generate weather reports [5].

SimpleNLG is an NLG system that allows the user to specify a sentence by giving its content words and their grammatical roles (such as subject or verb). It is implemented as a java library and it requires java programming knowledge to be used.

SimpleNLG allows the user to define flexible templates by using programming variables in the sentence specification. The variable parts of the templates could be filled with different values. When templates are used without an NLG system, they are called canned-text, and they have the disadvantage of not being very flexible, as only the predefined variables can change. When templates are defined using SimpleNLG, however, they keep all the functionality of the NLG system (for example, being able to modify the verb features or the output format, and making use of the grammatical knowledge), while also allowing for the variable values to change.

3. Serious Games and the Need for NLG

The term serious games refer to a sub-category of interactive simulation games in which the main objective is to train the player in a particular subject matter. The player is typically presented with challenging situations and is encouraged to practice different strategies at dealing with them, in a safe, virtual environment. Through tips and feedback provided during and at the end of the game, the player develops an understanding of the problem and what are the successful ways of confronting it [3].

The game ISO 14001 from DISTIL Interactive is an example of a serious game. The objective of this game is to train the player in the process of implementing an environmental management system (EMS) ISO 14001. The player controls the main character of the game, who manages the implementation of a standards-based process in a simulated fictional organization. All the other characters of the game are controlled by the computer. Feedback is provided as e-mail messages from other characters to the main character.

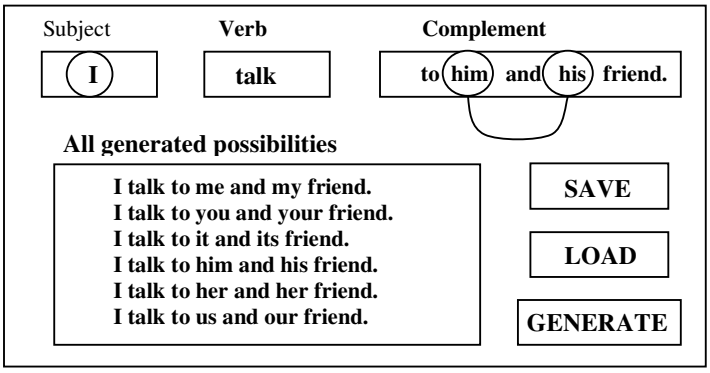
The amount of textual information required in serious games can be a burden on the game designers. It is then desirable to include templates that will statically provide the basic information, combined with variable parts that adapt the narrative to the circumstances. The templates were hard-coded in the game ISO 14001. In our current work, we propose the use of a more flexible way of generating templates for the dialog of the games. We present a NLG Template Authoring Environment that takes advantage of the grammatical knowledge of SimpleNLG in a simpler way. It does not require the user to have either advance linguistic or programming knowledge.

4. NLG Template Authoring Environment

The NLG Template Authoring Environment allows the user to input a model sentence template, to indicate its variables with their type and possible values, and to specify dependencies between variables. It then shows the user all the possible sentences that could be generated from the given template by calculating all the possible combinations of variable values that respect the specified dependencies. The user can then refine the template by changing either the given example or the specified variables and dependencies, in order to adjust the generated sentences to the needs of the game.

Figure 1 shows a graphical design for the NLG Template Authoring Environment and a simple example of a sentence with specified variables and dependencies. In this figure, the variables are indicated by circles and a dependency between variables is indicated by an arc connecting two circles.

Fig.1. Graphical Design for the NLG Template Authoring Environment



A prototype of the NLG Template Authoring Environment has been implemented in Java. It allows variables of different type, such as pronouns and some predefined noun subsets (e.i.: employee type). The SimpleNLG library was used to automatically generate correct sentences and provide the user with the possibility of exploring different attributes to the verb such as tense, form and mood.

In order to verify the correct functioning of the NLG Template Authoring Environment, we selected a set of sentence templates from the game ISO 14001. The templates were selected manually, while keeping in mind the need to cover different aspects, as for example the number and type of the variables and dependencies.

Templates that presented dependencies between variables were successfully generated with our system by declaring the variables and establishing the dependencies. In templates with variations to the verb the user has to be aware and specify them, as for example its tense and whether or not it uses modals such as "could" or "would". Another feature that the user needs to be aware of is the possibility of specifying dependencies between the main verb and a variable in the subject of the sentence.

5. Conclusions and Future Work

We have identified the need for a NLG Template Authoring Environment that allows game content designers without linguistic and programming background to experiment with and finally design language templates. We have designed a system that allows the user to specify an example sentence together with variables, its dependencies, and verb options that complete the template. This system shows the user all the possible sentences that could be generated with the specified template. It can be used to refine the template until it satisfies the user's needs. We have implemented a simple prototype that makes use of the SimpleNLG java library which provides us with correct sentences and the possibility of including many verb variations. We have evaluated our NLG Template Authoring Environment in a set of sentence templates from the game ISO 14001 covering different characteristics.

In this version, we enforced some limitations to make the prototype manageable. In particular, the current system is text only and does not allow for refinements. In the future we will provide a user-friendly intuitive graphical interface that will allow the user to iteratively make changes to the sentence, variables and dependencies definitions. In addition, in a future version a module will be added in order to allow users to create their own subset of nouns or adjectives to be added as variables' type.

Acknowledgements
This work is supported by the Ontario Centres of Excellence (OCE) and Precarn Incorporated.

References

[1] Bateman, J.A: Natural Language Generation: an introduction and open-ended review of the state of the art. (2002)
[2] Bateman, J.A: Enabling technology for multilingual natural language generation: the KPML development environment. Journal of Natural Language Engineering, 3(1):15-55. (1997)
[3] French, D., Hale, C., Johnson, C. and Farr, G.: Internet Based Learning: An introduction and framework for higher education and business. London, UK: Kogan Page. (1999)
[4] Gagné, R. M., & Briggs, L. J.: Principles of instructional design (4th Ed.). Fort Worth, TX: Harcourt Brace-Jovanovich. (1997)
[5] Goldberg, E., Driedger, N., and Kittredge, R.I.: Using Natural-Language Processing to Produce Weather Forecasts. IEEE Expert: Intelligent Systems and Their Applications. 9(2):45-53. (1994)
[6] Reiter, E.:. SimpleNlg package: http://www.csd.abdn.ac.uk/ereiter/simplnlg (2007).