



Automatic Generation of Personalised and Context-Dependent Textual Interventions During Neuro-rehabilitation

Timon Felske¹ · Sebastian Bader¹ · Thomas Kirste¹

Received: 1 October 2021 / Accepted: 24 May 2022 / Published online: 8 September 2022
© The Author(s) 2022

Abstract

In this paper we present our system that synthesises personalised and context dependent texts during robot guided exercises for neuro-rehabilitation. This system is used to generate texts for the communication between a care robot and patients. We present requirements that a system in such a medical domain has to meet. Afterwards the results of a systematic literature review are presented. We present our solution based on the RosaeNLG system. It supports different language levels and referring expressions in a real-time text generation system, so that generated texts can be adapted to the reader in the best possible way. We evaluate our system with respect to the requirements. The contribution of the paper is twofold: We present a set of requirements for Natural Language Generation (NLG) in medical domains and we show how to extend RosaeNLG with an external dialogue memory to handle complex referring expressions in medical real time settings.

Keywords NLG · Referring expressions · Multi modal interaction · Context dependencies · Linguistic references · RosaeNLG

1 Introduction

The number of stroke patients, especially in the elderly population is increasing significantly worldwide [8]. The disabilities after a stroke can be greatly reduced with intensive and frequent training [13]. Therefore more therapies and therapists are needed to support the neuro-rehabilitation of the patients. Due to the shortage of therapists, an optimal therapy can no longer be guaranteed for all patients. This leads to a reduced quality of life and slower recovery of the patients [8].

Intelligent assistance systems can be used to support therapists at their work and patients with their post-stroke recovery [8]. In this paper we present an implemented text synthesis module that is used as a component in our intelligent assistance system to support the neuro-rehabilitation.

An exemplary test setup of this system can be seen in Fig. 1. This text synthesis module generates the texts inside the speech bubble and is essential for the natural multi-modal interaction between robot and patient.

In the neuro-rehabilitation the therapists support their patients during the execution of several exercises. The exercises are designed to train the abilities a patient lost due to a stroke. Each exercise is first demonstrated by the therapists to show the correct execution. After that the patient repeats the exercise. The therapists assist and give feedback.

In our project, several therapy sessions were observed by experts and the structure of the interaction as well as the respective contents of a dialogue section were recorded in a *neuro-rehabilitation therapy manual (NTM)*.

Our assistant system implements that NTM using a Moore machine with states for each part of the manual. Each time a state is entered a predefined content is shown on an output device (tablets integrated into pepper robot, additional tablet on the table for better readability) and uttered via voice output of the robot. This multi-modal content consists of multiple components:

- Text, which is shown and read
- Image / Video, which is displayed
- Buttons / Input-field, in case of a question

✉ Sebastian Bader
sebastian.bader@uni-rostock.de

Timon Felske
timon.felske@uni-rostock.de

Thomas Kirste
thomas.kirste@uni-rostock.de

¹ Department of Computer Science, University of Rostock,
Albert-Einstein-Straße 22, 18059 Rostock, Germany

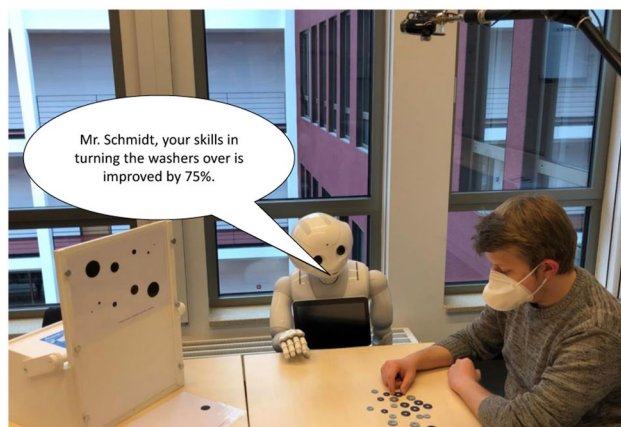


Fig. 1 Illustration of our intelligent robotic assistance system and the interaction between a Pepper robot and a patient

To personalize the texts and adapt them to the current context as well as possible, a text synthesis is required.

In order to implement this text synthesis, we determined requirements that the generated texts and the system itself must meet. Afterwards a literature analysis was carried out in order to identify existing systems and concepts. Since no system that meets our requirements was found, we developed and evaluated our own system.

2 Requirements

In close cooperation with our medical partners (neuro therapists and psychologists), we identified the following requirements for a text generation system in E-BRAiN:

- R1. Texts have to be user and content dependent.
- R2. Because this system is used in Germany, the text synthesis has to generate German¹ texts.
- R3. The content to be conveyed and the document structure is already given in a specified guide.
- R4. Texts have to be generated for different language levels, depending on the cognitive abilities of the patient.
- R5. The texts should be as natural as possible. The patient should not be able to distinguish between texts synthesised by our system and a text written by a human.
- R6. Full offline capability, due to the usage in a clinical environment.
- R7. Text synthesis in real time to be adapted to the current context.

¹ We use English translations for all examples presented here, to make the paper readable for non-German speakers.

Those requirements were identified during informal expert interviews with our medical project partners (N=5, medical doctors and nurses from neurology, psychology, and psychotherapy). They are quite specific for the addressed E-BRAiN project, and arose from the very determinate form of the interaction between therapist and patient.

3 State of the Art

We performed a systematic literature review to identify existing systems and concepts for text generation in a medical context. During the review, 22 papers were selected that met our search criteria. Table 1 shows a summary of the best tools we found with respect to our requirements.

Suter et al. have developed a system to simplify existing German texts into simple language [12]. To implement this, a rule-based procedure was used. It was found that the system is able to simplify texts. But, the system lacks the ability to simplify single difficult words or phrases, and in those situations a simplification by hand was found to be better.

Also other systems and concepts like the Penman system [7] for the English language and the extension by KOMET [5] for the German language were considered, but we classified them as unusable in our case, since no implementations are available. The same applies to the VIE-LANG system [2], which uses the semantic network SEMNET to synthesise texts as response to dialogue inputs.

Puzikov and Gurevych [9] used a template-based model and compared it with a neural model in order to take part in the E2E NLG Challenge. In this comparison, the template-based model prevailed due to its reliability and absence of errors. The template-based approach has also proven to be successful in other areas. For example, ontologies can be automatically translated into texts [1].

RosaeNLG [11], a node.js library based on the template engine Pug, implements a template-based approach. In addition to the German language, it also supports English, French, Italian and Spanish. Furthermore, it is completely offline-capable and already provides relevant functions for the natural language generation, such as the correct selection of an article, the correct conjugation of verbs, etc.

4 Implementation

Our system is built on top of RosaeNLG, extended with an explicit external dialog state. After introducing the general architecture, we will describe the following sub-problems to be solved in order to make the texts as natural as possible: synonymic alternatives, context dependencies, linguistic references, and language complexity levels.

Table 1 Comparison of existing approaches wrt. our requirements

System	R1	R2	R3	R4	R5	R6	R7
Suter ea.	o	+	+	+	o	+	+
Mann	+	-	-	?	o	?	o
Henschel	+	+	-	?	o	?	o
Buchberger ea.	+	+	o	-	o	?	+
Puzikov ea.	+	+	+	-	+	+	+
RosaeNLG	+	+	+	+	+	+	+

4.1 System Architecture

As mentioned above, the general flow of the therapies is fixed by the NTM. Therefore, we only have to perform the context-adaptation, micro-planning and the surface realisation. We implemented the dialog structure using a Moore machine, with the automaton states corresponding to states of the NTM and transitions for each possible user interaction (texts, images, ...) is associated, which are shown to the user while entering the state. We will refer to those multi-modal interactions as *communicative act* below. Fig. 3 shows a small part of such an automaton.

4.2 Synonymic Alternatives

To support diversity of the generated texts, we implemented different versions of a given communicative act as synonymic alternatives. This is already supported by RosaeNLG using, for instance, the built-in *synz* mechanism. Using this we can enable our system to generate texts in different forms, for example, switching between simple alternatives like “Hello” and “Welcome”, but also choose between larger synonymic statements like “You didn’t answer me” and “I did not get an answer from you”.

4.3 Context Dependencies

In our domain the generated texts have to be customised with respect to the following context information:

- (R1) Demographic information (name, age, ...)
- (R2) Medical information wrt. impairment (affected side, severity, ...)
- (R3) Cognitive abilities (language understanding, vision, ...)
- (R4) Performance during the exercises.

While the first context items (R1-3) are static for a given person, the item (R4) depends on information collected during the user interaction.

We implemented context-dependent text generation using the built-in functionality of RosaeNLG, in particular using

```

| Try to imagine that the hand
if (lang_level == "normal")
| , you see in the mirror on the table,
else
| in the mirror
| is your
| #[+agreeAdj(para_side, 'Hand')] hand.
    
```

Fig. 2 A small RosaeNLG code sample for context-dependent text generation. Texts in typewriter font are output. The construct `+agreeAdj` generates the correct reference to the impaired hand

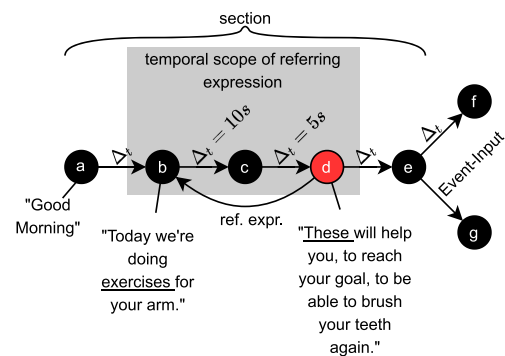


Fig. 3 A small fragment of the automaton. 7 states are shown with corresponding *state_ids* (a–g). The referring objects are underlined

If-Then, `agreeAdj` and the `value-mixin` statements as shown in Fig. 2.

4.4 Linguistic References

Each state of the automaton corresponds to one communicative action. Sub-sequences of states, corresponding to sections of the NTM, are meant to convey a larger piece of information. To make the interaction as natural as possible, we would like to use linguistic references within a single communicative action and also across the different states within a section. RosaeNLG allows the usage of anaphoric references while generating the texts, as long as the text is generated during a single invocation of the system, but has no mechanism to refer to texts generated in a previous invocation.

Using a linguistic (anaphoric) reference to refer to some information just presented makes the texts sound very natural. But if the time between introducing a concept and referring to it is too long, the user might not recall what is meant by anaphoric references like “it” and “this”. Unfortunately, we have no control over the amount of time spent in a state as this is controlled by the user. It might, for instance, be that the user needs a break for some reason.

Our dialogue follows the states of the underlying automaton, as shown in Fig. 3. The states are linked to each other by temporal or event-triggered transitions. Referring expressions over multiple states should be used whenever possible, but they should only be used if they are in a limited temporal scope. This is the case if the state is linked to other states, describing the same context and if a time span Δ_i is below a defined amount of seconds. In Fig. 3, the temporal scope of referring expressions is illustrated.

RosaeNLG allows anaphoric references within a single call. In our case, however, the text for a given state of the automaton can only be generated while entering the particular state, therefore, we could not use the mechanism provided by RosaeNLG. To solve this problem we introduced an external dialogue memory. With this memory it is possible to generate the texts of several states jointly, split them into chunks corresponding to the states and to save them individually. One *section* consists of multiple *state_ids* and implements the idea of the temporal scope of referring expressions. With this technique we are able to use the same referring expressions in multiple *state_ids*.

In case our system receives a request to generate a text of a given *state_id*, it checks to which *section* the *state_id* belongs. The texts of successor *state_ids* of the *section* are synthesised and stored in our external dialogue memory. This external memory is implemented as a cache with an adaptive timeout. The entries will be removed as soon as the time span after introducing some concept exceeds a user dependent time out.

After that time span the concept that is referenced, must be explicitly renamed. If the time out was set to 3 seconds in Fig. 3, the following text is generated in state d: “*The exercises will help you, to reach your goal, to be able to brush your teeth again.*”.

4.5 Language Complexity Levels

To adapt the texts to the current patients abilities, we introduced the concept of language levels (R4) into our system. Before starting the therapy, various tests are carried out in which the physical and cognitive abilities of a patient are determined. Based on these values, a language level is chosen for each patient. Currently, texts can be generated in normal and simplified German, but other levels are possible as well. Simple language is often used to communicate

with people with limited cognitive skills. For this purpose, all texts were simplified and checked with online tools for simplified language [6] and [10]. The language complexity level is added as a dynamic context information that can be adjusted and evaluated as sketched in Section 5.3.

This concept can also be used to include other types of language levels. For example, people with left brain impairment have limited processing of logical and mathematical statements. The other way around, people with impairment of the right hemisphere can process these statements very well [3]. So, depending on the patient, logical/mathematical or creative texts can be used to communicate with him/her.

4.6 Debugging Functionality

As texts in medical settings must be correct we integrated a debugging functionality into our system. We extended RosaeNLG and our dialog engine such that we are able to generate all possible paths through the dialog including all possible verbalisations of the texts. All texts including further multimedial content is exported into a PDF with a single page for each state. Transitions between states are integrated as hyperlinks between the corresponding pages. This allows medical experts to check the correctness and dynamics of the dialogues.

5 Evaluation

To evaluate our system we tested three different aspects: speed of the generation, correct context adaptation, and naturalness of the generated texts. In total,

To measure the speed we generated all texts for one of our implemented therapies (total 348 states) and measured the time between request and response (including network overhead). The mean time span was 11.67 ms, with a minimum of 4.97 ms, and maximum of 99 ms), which meets requirement R7.

To evaluate the correct adaption to a given patient, we generated a PDF-file that contains some of the texts (in total about 100 sentences) and asked a domain expert to verify them wrt. language quality and correct context adaption. To check the reliability of this method, we introduced some mistakes on purpose. For example, instructions for individual exercises were mixed up, as well as incorrect context information such as the side (left or right) of the patient to be treated. The expert only marked our forced mistakes, but no other problems. This shows that the system adapted all texts correctly to the patient and requirements R1–4 and R5 are satisfied. To further evaluate the correctness and naturalness of the generated texts, we used an online-tool [4] that checks the German spelling and grammar. To verify the simple language we used the tools mentioned above. The

checkers detected some minor improvements (e.g., small spelling mistakes in medical terms and swapped letters) which were integrated into the system.

6 Conclusion

In this paper, we describe a tool which is able to generate user and content adapted texts for predefined content in a medical domain. After identifying necessary requirements, we performed a systematic literature review and designed a system based on RosaeNLG.

To meet all requirements for our application domain, we extended RosaeNLG with an external dialog memory. We evaluated the performance and correctness of the system. In total, we have implemented 4 different neuro-rehabilitation therapies, which will be evaluated in a clinical trial in the upcoming months.

Funding Open Access funding enabled and organized by Projekt DEAL. The research project “*E-BRAiN - Evidence-based Robot Assistance in Neurorehabilitation*” is supported by the European Social Fund, reference: ESF/14-BM-A55-0002/19, and the Ministry of Education, Science and Culture of Mecklenburg-Vorpommern, Germany.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Bense H, Schade U (2015) Ontologien als Schlüsseltechnologie für die automatische erzeugung natürlichsprachlicher texte. X.Media Press, Springer, Berlin, pp 227–245. https://doi.org/10.1007/978-3-642-54886-4_16
2. Buchberger E, Horacek H (1988) VIE-GEN a generator for German texts. Natural language generation systems. Springer, New York, pp 166–204. https://doi.org/10.1007/978-1-4612-3846-1_5
3. Corballis MC (2014) Left brain, right brain: facts and fantasies. PLoS Biol 12(1):e1001767. <https://doi.org/10.1371/journal.pbio.1001767>
4. Cornelsen Verlag GmbH (2022) Duden Mentor. <https://mentor.duden.de>. Accessed 25 Jan 2022
5. Henschel R (1994) Merging the English and the German upper model. In: Arbeitspapiere der GMD Vol. 848, St. Augustin, GMD. <https://d-nb.info/941979938>
6. Hep Hep Hurra e.V. (2021) Hurraki Language Tool Prüfung auf Leichte Sprache. <https://www.hurraki.de/pruefung/pruefung.htm>. Accessed 25 Jan 2022
7. Mann WC (1983) An overview of the penman text generation system. In: AAAI-83 Proceedings
8. Matarić MJ, Eriksson J, Feil-Seifer DJ, Winstein CJ (2007) Socially assistive robotics for post-stroke rehabilitation. J Neuroeng Rehabil. <https://doi.org/10.1186/1743-0003-4-5>
9. Puzikov Y, Gurevych I (2018) E2E NLG challenge: neural models vs templates. Proc Int Conf Natl Lang Gener Assoc Comput Linguist. <https://doi.org/10.18653/v1/w18-6557>
10. Sächsisches Staatsministerium für Soziales und Gesellschaftlichen Zusammenhalt (2022) Online-Prüfer: Texte auf leichte sprache prüfen lassen. <https://www.leichte.sprache.sachsen.de/online-pruefer.html>. Accessed 25 Jan 2022
11. Stoecklé L (2021) Rosaenlg - a natural language generation library for node.js (version 2.1.7). <https://rosaeNLG.org/rosaeNLG/2.1.7/index.html>. Accessed 25 Jan 2021
12. Suter J, Ebling S, Volk M (2016) Rule-based automatic text simplification for German. Conf Natl Lang Process (KONVENS 2016). <https://doi.org/10.5167/UZH-128601>
13. Teasell RW, Kalra L (2005) What's new in stroke rehabilitation. Stroke 36(2):215–217. <https://doi.org/10.1161/01.str.0000153061.02375.04>