# Processing open text input in a scripted communication scenario – Extended abstract –

Raja Lala[1], Johan Jeuring[1,2], F.P.M. Heemskerk[1], Marcell van Geest[1], Jordy van Dortmont[1], Gabriel Gutu-Robu[3], Stefan Ruseti[3], Dascalu Mihai[3], Beatrice Alex[4,5], and Richard Tobin[4]

[1]Utrecht University Computer Science Department, Netherlands
[2]Faculty of Management, Science and Technology, Open University Netherlands
[3]University Politehnica of Bucharest, Computer Science Department, Romania
[4]University of Edinburgh, School of Informatics, United Kingdom
[5]The Alan Turing Institute, London, United Kingdom

**Abstract.** Serious games often employ pre-scripted dialogues and interactions with a player; in contrast to free user input that enables deeper immersion. In this paper we explore possibilities for interactive natural language dialogue in a serious game by combining Natural Language Processing (NLP) techniques with dialogue management. Our game learning environment has a communication scenario editor in which a domain expert develops a structured, scripted scenario as a sequence of potential interactions. A communication scenario is context-specific and often follows a protocol - for instance, delivering bad news to a patient. Currently, a player navigates through a simulation and converses with a virtual character by choosing a statement option from one of the pre-scripted player statements, at each step in the simulation. We develop a scenario-specific corpus method (SSCM) to process open responses (i.e. natural language inputs) in our learning environment. We conduct an experiment to collect data for comparing SSCM against multiple NLP methods, and another experiment to investigate if framing can improve processing open-text input using SSCM in a communication simulation.

## 1 Introduction

Many universities and vocational programs train students in communication skills. Communication skills are best learned through practice, in role-play or with a simulated patient [1]. In a digital learning environment for training communication skills, a student often performs a conversation with a virtual character, and the learning environment assesses the performance of each student against the conversation's learning goals. Serious games often employ pre-scripted dialogues and interactions with a player; in contrast to free user inputs that enable deeper immersion.

Our game learning environment Communicate [5] provides a communication scenario editor in which a domain expert develops a structured, scripted scenario

as a sequence of potential interactions. A scenario is context-specific and often follows a communication protocol - for instance, delivering bad news to a patient. Communicate provides expressive features to a scenario author and decouples scenario development from the implementation of a communication simulation. An author typically encodes a learning goal for a scenario e.g. assertiveness as a parameter. A player statement usually has an incremental value on a parameter and triggers an emotional effect e.g. 'Happy' in a VC. A structured scenario represents the expert knowledge of a communication skills teacher for a particular protocol in a domain.

A scenario simulation in Communicate [5] presents statement choices to a player at a step of a scenario. A player navigates through a simulation and converses with a virtual character by choosing a statement option from one of the pre-scripted player statements, at each step in the simulation. In this respect, a scenario currently resembles a sequence of multiple choice questions. Communicate has a good take-up; more than twenty teachers/teaching assistants use it as part of communication skills education at different faculties (medicine, veterinary science, pharmacy, psychology etc.) of Utrecht University. Other users include the city-municipality, some social services organisations, a few hospitals and a national-level government organisation.

## 2 Research questions and experiments

Martinez [6] describes how test item formats vary in cognitive load and in the range of sampled cognition processes. Multiple-choice items often elicit low-level cognitive processing, whereas constructed-response items more often require complex thinking. Test item formats pose trade-offs in the dimensions of cognitive features, psychometric characteristics, and costs of administration and scoring. However, there is no format appropriate for all purposes and for all occasions.

Hammer et al. [4] assert that the most appropriate value assigned to a word in the sentiment lexicon depends on the domain. They advocate that a sentiment lexicon needs to be specialised for each particular domain.

We explore possibilities for interactive natural language dialogue in a serious game by combining Natural Language Processing (NLP) techniques with scripted dialogue management. Our contribution is to use information present within a communication scenario to process open-text player-input. We use a scenario as a basis to develop a scenario-specific corpus and we match a player open-text input to a pre-scripted statement choice at a step in a scenario using this scenario-specific corpus. Our research question is: 'How does the scenario-specific corpus method (SSCM) compare to some other Natural Language Processing (NLP) techniques, when matching user open-text inputs to predefined answers?'

We extended Communicate to perform an experiment in spring-summer 2018. The focus of this experiment is to gather data to compare SSCM versus other NLP methods.

At our University, final year bachelor computer science students work in a team project and develop a software product for a real customer. In spring-summer 2018 there are a total of eighty two students assigned in eight teams of ten to twelve students each. Seventy eight students gave consent to use their data for research. The age of the students ranges between twenty and twenty-eight years.

We developed a scenario called *Samenwerken* (Collaborate) in Communicate to train a student in collaboration skills. We adapted Communicate to gather data in this experiment: a student gets an open-text input box in which she writes her response instead of choosing from the multiple choices at each step. A student inputs her text, after which Communicate displays the available scripted statement options at this step. There is also an option *No response matches* displayed at each step. A student indicates which statement is closest to her open-text input, or chooses *No response matches* in case no scripted statement matches her input. If a student chooses *No response matches*, Communicate thereafter asks her to select one of the scripted statement options to continue the simulation.

Two independent experts annotate the play-throughs from the students in this experiment. We compared a match between a student and the two annotators. For statements where a match is present, we run a two-way random effects model of ICC (Intraclass Correlation Coefficient) and Cronbachs alpha. We reason that the agreement between a student and the two annotators represents the upper-bound an NLP match-method can achieve as a match-method cannot exceed human comprehension. To compare SSCM, we use open-source NLP methods namely: a) fuzzy string matching (`glench.github.io/fuzzyset.js`), b) cosine similarity between word stems, c) semantic distance measures exposed by the ReaderBench (RB) framework [2], d) semantic similarity computed using spaCy (`https://spacy.io`). We also investigate comparing an input to a cluster of strings.

In our second experiment in fall-winter 2018, we use SSCM to handle open-text at run-time of a scenario simulation. The focus of this experiment is more pedagogical.

Entman. [3] describes framing as selection and salience; select an aspect of a perception and highlight that aspect in a communicating text, to promote a particular interpretation. Van Lehn et al. [7] study standard behaviour in Intelligent Tutoring Systems (ITS) and find that giving hints and feedback at a step level of an ITS improves student learning.

We investigate if framing (highlight and hint) in a dialogue can improve processing open-text input in our game learning environment Communicate. Our research questions are: 'a) what is the effect of highlighting on a student's choice; and b) do hints influence a student following a scripted scenario?'

We assign half the students in fall-winter 2018 software projects to an experiment and a control group. We modify Communicate to use SSCM to match a player's open-text input to available pre-scripted statement choices at a step of a dialogue scenario. A match-method has a threshold-value below which there

is no match. SSCM takes an open input text and returns a match-score per scripted statement at a step of the simulation. If all returned match-scores for a input statement are below the threshold value for SSCM, Communicate provides a hint to a player in the experiment group. To the control group students, Communicate says, "I could not understand". In both cases, Communicate prompts a player to input a new text.

If at least one statement option has a match-score above the threshold-value, we display all options, highlight the best match and ask a player to choose an option closest to her input. To measure the effect of highlighting, we conduct a 2nd round with the students a few weeks after the 1st round; to ensure that a student does not automatically remember the choices. Communicate presents a student her play-through from the 1st round. At each step, Communicate displays the statement a student entered in the 1st round, along with the statement options available at that step of the scenario, and an option *No response matches*. A student chooses an option closest to her input from the 1st round or *No response matches*.

## References


1. Marianne Berkhof, H. Jolanda van Rijssen, Antonius J.M. Schellart, Johannes R. Anema, and Allard J. van der Beek. Effective training strategies for teaching communication skills to physicians: An overview of systematic reviews. *Patient Education and Counseling*, 84(2):152–162, 2011.
2. Mihai Dascalu, Wim Westera, Stefan Ruseti, Stefan Trausan-Matu, and Hub Kurvers. Readerbench learns dutch: Building a comprehensive automated essay scoring system for dutch. In *18th Int. Conf. on Artificial Intelligence in Education (AIED 2017)*, pages 52–63. Springer, 2017.
3. Robert M Entman. Framing: Toward clarification of a fractured paradigm. *Journal of communication*, 43(4):51–58, 1993.
4. Hugo Hammer, Anis Yazidi, Aleksander Bai, and Paal Engelstad. Building domain specific sentiment lexicons combining information from many sentiment lexicons and a domain specific corpus. In *IFIP International Conference on Computer Science and its Applications*, pages 205–216. Springer, 2015.
5. Johan Jeuring, Frans Grosfeld, Bastiaan Heeren, Michiel Hulsbergen, Richta IJntema, Vincent Jonker, Nicole Mastenbroek, Maarten van der Smagt, Frank Wijmans, Majanne Wolters, and Henk van Zeijts. Communicate! — a serious game for communication skills —. In *Proceedings EC-TEL 2015: Design for Teaching and Learning in a Networked World: 10th European Conference on Technology Enhanced Learning*, volume 9307 of *LNCS*, pages 513–517. Springer, 2015.
6. Michael E Martinez. Cognition and the question of test item format. *Educational Psychologist*, 34(4):207–218, 1999.
7. Kurt Vanlehn. The behavior of tutoring systems. *International journal of artificial intelligence in education*, 16(3):227–265, 2006.