Enhancing Free-text Interactions in a Communication Skills Learning Environment

Raja Lala, Utrecht University, r.lala@uu.nl
Marcell van Geest, Utrecht University, marcell@marcell.nl
Stefan Ruseti, University Politehnica of Bucharest, stefan.ruseti@cs.pub.ro
Johan Jeuring, Utrecht University, J.T.Jeuring@uu.nl
Mihai Dascalu, University Politehnica of Bucharest, mihai.dascalu@cs.pub.ro
Jordy van Dortmont, Utrecht University, jordy@dialoguetrainer.com
Gabriel Gutu-Robu, University Politehnica of Bucharest, gabriel.gutu@cs.pub.ro
Michiel Hulsbergen, DialogueTrainer, michiel@dialoguetrainer.com

Abstract: Learning environments frequently use gamification to enhance user interactions. Virtual characters with whom players engage in simulated conversations often employ pre-scripted dialogues; however, free user inputs enable deeper immersion and higher-order cognition. In our learning environment, experts developed a scripted scenario as a sequence of potential actions, and we explore possibilities for enhancing interactions by enabling users to type free inputs that are matched to the pre-scripted statements using Natural Language Processing techniques. In this paper, we introduce a clustering mechanism that provides recommendations for fine-tuning the pre-scripted answers in order to better match user inputs.

Introduction

Digital learning environments are frequently used in education, in our particular case for training communication skills. Learners discuss with virtual characters, while the environment can determine whether they reached the goals established for the conversation (Jeuring et al., 2015). A communication scenario in Dutch was built for this purpose consisting of a sequence of replies between a virtual character and the learner. The script provided answer options and alternative branches corresponding to the previous choices that were made by the user. The aim of this paper is to improve users' learning experiences by enabling free-text inputs with the help of Natural Language Processing (NLP) techniques. After testing multiple NLP similarity matching methods on a dataset consisting of open input text answers annotated with matching options, clustering was performed on the students' answers and the predefined ones in order to improve the scenario.

Wang and Petrina (2013) performed discourse analysis on conversations between a student learning a new language and a digital tutor. It was observed that chatbots should reply to learners' input, but they should also provide feedback on errors. Our aim is to provide support for better matching user inputs and enhancing the feedback received from the virtual character.

Method

Various Natural Language Processing (NLP) techniques were used in our experiment to match free text inputs provided by players in Dutch to a set of scripted options, namely: a) FuzzySet, an open source library (https://glench.github.io/fuzzyset.js) based on syntax; b) SpaCy (https://spacy.io), an advanced NLP framework integrating syntactic dependency parsing and part-of-speech tagging; c) string kernels, a method that compares texts without the need of a large training corpus (Lodhi, Saunders, Shawe-Taylor, Cristianini, & Watkins, 2002); and d) Scenario-Specific Corpus Method (SSCM) in which experts annotate emotions for each statement of the virtual character (Lala et al., 2018). The final similarity score used in follow-up analyses was calculated as the average between spaCy similarity and the average of string kernels.

However, a large number of input answers from our dataset did not match any pre-scripted statement. This shows that our scenario is incomplete in terms of pre-scripted options and influenced the accuracy of matching algorithms. An additional clustering experiment was performed on the list of input answers and predefined options put together for each stage in the scenario. Affinity Propagation (https://scikit-learn.org/stable/modules/clustering.html#affinity-propagation) was chosen as it relies on a distance function between points (i.e., inverse of semantic similarity between answers in our case) and there is no requirement for an a priori number of clusters. Similar answers were grouped together with the aim of fixing two common problems: 1) cases in which two predefined options were in the same cluster – one could be removed since these options are too similar compared to the variety of answers provided by students; 2) cases when multiple answers are grouped in a cluster which does not contain a predefined option – a new option related to them should be added in order to enhance the chances of adequately matching new user inputs to pre-scripted answers.

Results

In our experiment, users were shown the available options after typing their input and they had to indicate which one was the closest to their intention, or select "No response matches" if no pre-scripted answer was adequate. The dataset consisted of 126 statements, out of which 59 (47%) had a match (Lala et al., 2018). Thus, a threshold had to be set, below which the answer was considered not to match any option. FuzzySet had a built-in threshold value. For spaCy and string kernels, a threshold was set statistically as the sum between the average and the standard deviation of all scores. For SSCM, the threshold was empirically established to .65. Table 1 provides the accuracy scores for each method. The best global accuracy was obtained by combining spaCy similarity with string kernels which was chosen as the distance function for clustering.

Table 1: Accuracy of matching methods

Method	Accuracy for matched answers (%)	Accuracy for unmatched answers (%)	Global accuracy (%)
FuzzySet	27%	82%	56%
spaCy	27%	84%	57%
String kernels	31%	**96%**	65%
SSCM (using Cosine similarity)	**54%**	48%	51%
spaCy and string kernels	37%	93%	**67%**

Allowing users to play the scenarios by writing texts in natural language lead to statements that were considerably different to the pre-scripted ones. This can emphasize an incomplete scenario, but it also has critical impact on the accuracy of NLP matching algorithms. If several similar answers are not matched to a pre-scripted answer, the scenario could be improved by adding an option similar to them. Thus, by analyzing the generated clusters, our algorithm provided improvement suggestions. These suggestions covered cases when a statement choice can be added every time a group of similar input answers do not have a matching candidate, and cases when the options were not different enough to be split across clusters and should be further adapted. As the semantic matching algorithms are far from being perfect, the clustering results and suggestions need to be analyzed by experts, who decide whether the scenario should be modified. The clustering algorithm can be applied repeatedly, thus enabling experts to visualize the effects of each modification.

Conclusions

This paper presents an experiment of improving a communication learning environment by replacing a multiple-choice selection with free input texts that are matched using NLP techniques. A dataset of annotated Dutch statements was created, and a clustering algorithm was applied to provide suggestions for improving the learning scenario. Users engaged in innovative discussions with the chatbot, while striving for a coherent conversation following pre-scripted actions. Nevertheless, the limitations of NLP matching methods in terms of accuracy must be emphasized, together with the need of a larger training corpus.

References

Jeuring, J., Grosfeld, F., Heeren, B., Hulsbergen, M., IJntema, R., Jonker, V., Mastenbroek, N., van der Smagt, M., Wijmans, F., Wolters, M., & van Zeijts, H. (2015). Communicate! --- A Serious Game for Communication Skills ---. *Design for teaching and learning in a networked world, 9307*, 513-517.

Lala, R., Jeuring, J., Heemskerk, R., van Geest, M. , van Dortmont, J., Gutu-Robu, G., Ruseti, S., Dascalu, M., Alex, B., & Tobin, R. (2018). Processing Open Text Inputs in a Scripted Communication Scenario. In *SEMDIAL 2018* (pp. 211–214). Aix-en-Provence, France.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learning Research, 2*(Feb), 419–444.

Wang, Y. F., & Petrina, S. (2013). Using learning analytics to understand the design of an intelligent language tutor–Chatbot lucy. *Editorial Preface, 4*(11), 124–131.

Acknowledgments

This activity has received funding from the European Institute of Innovation and Technology (EIT). This body of the European Union receives support from the European Union's Horizon 2020 research and innovation programme. This research was also partially supported by the 644187 EC H2020 RAGE project and by the FP7 ICT STREP project LTfLL (http://www.ltfll-project.org/).