Lecturer-Chatbot: An AI for Advising Struggling Students in Introductory Programming Extended Abstract SACLA 2019

Mohammed Ismail[1][0000−0003−2640−9800] and Abejide Ade-Ibijola[2][0000−0001−9507−0455]

Formal Structures, Algorithms, and Industrial Applications Research Cluster
Department of Applied Information Systems
School of Consumer Intelligence and Information Systems
University of Johannesburg, Auckland Park Bunting Road Campus
Johannesburg, South Africa
mohammedis271@gmail.com, abejideai@uj.ac.za www.abejide.org

Extended Abstract

Students often struggle to communicate with their peers or lecturers about some of the issues they face during their time at university, either it be academic or personal [1]. One of these issues is understanding Introductory Programming. Programming requires an understanding of how certain logical flows or algorithm work [2]. Teaching students takes a great deal of time, and at times students who lack confidence prefer not to ask questions in class. This results in students failing to understand the foundations of introductory programming. Some investigations have been conducted on students battling with the initial steps of figuring out the basic programming concepts [1].

For many first year students, adapting to the new university environment is a struggle and can take a considerable amount of time. Sometimes students may not attend their lectures and may generally drop out of university in the middle of their first year. Changes in the environment are not the only challenges students face. Financial aspects play a big role in a students academic life [3]. Some students may miss classes in order to work for funding for their tuition. Other students opt to take out bank loans with high interest rates in order to pay their tuition [4, 5, 6].

The reason for creating a Lecturer-Chatbot is to enable students to get advice based on topics that they may be afraid to ask from lecturers or peers. With this tool they will be able to talk to an Artificial Intelligence program that gives them advice on specific issues. This tool will allow students to get advice so that they are able to cope with some of the issues they face and resolve problems in a viable manner as well as mitigate the thoughts of dropping out.

Lecturer-Chatbot differs from many other chatbots because it is an Advisory Tool for struggling student compared to other popular chatbots like SIRI (A

virtual assistant for apple users) [7], Google Assistant (This is a virtual assistant used by android users) [7], BIXBY (This is Samsung's virtual assistant for all users who are using Samsung devices) and ALEXA (a virtual assistant developed by amazon) [7, 8, 9]. Lecturer-Chatbot is different in that it helps students struggling with academic issues specifically in introductory programming and other personal problems. A student can either choose to solve a problem, be entertained, or follow a more formal approach and ask relevant questions on how to overcome introductory programming learning problems [10, 11].

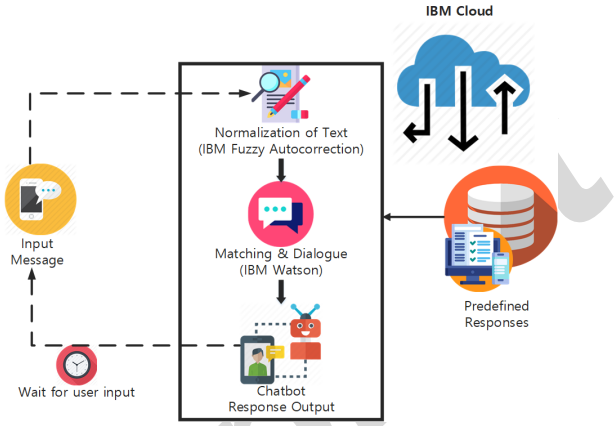

Fig. 1. The workings of Lecturer-Chatbot

In Figure 1, we can see a graphical representation of the work flow of Lecturer-Chatbot. A student having an issue chats with Lecturer-Chatbot, the chatbot will then cross-reference the question with its predefined responses, and then produce a response for the user.

To summarise the contributions of this paper, we have:

1. designed a working prototype of Lecturer-Chatbot mobile application that produces relevant responses.
2. introduced machine learning to produce the best responses. The application is able to identify typing mistakes and to learn new synonyms so that it may give the best response based on the present questions.
3. sent out a survey to students who used the Lecturer-Chatbot demo to evaluate its usefulness.

The following are the functions of Lecturer-Chatbot.

1. Lecturer-Chatbot is able to have general conversations with the user. The user can ask Lecturer-Chatbot for a joke.

2. Lecturer-Chatbot can advise students on academic issues. Lecturer-Chatbot will ask the user various questions to analyse if the student is attending class or not, it will then give the students programming example and solutions, or give them a YouTube link to learn Introductory Programming online.
3. Lecturer-Chatbot can advise students on personal issues. Lecturer-Chatbot will ask the user various questions to understand the emotions of the user and then give them advise accordingly. Lecturer-Chatbot will then advise the student to head to the psycad[1] in order to get professional help thereafter.
4. Lecturer-Chatbot will advise students on what steps to take when they are facing a certain situation. Lecturer-Chatbot may either tell them to visit the psycad or visit the lecturer with the module they are struggling with.
5. Lecturer-Chatbot can also do daily tasks. The user can request for a doctor and it will call the nearest doctor from the user. Lecturer-Chatbot will first get the location of the user and then contact the nearest medical facility from the user.

In Figure 2, we show the various branches of the bot's inference. Each node represents the state the Lecturer-Chatbot can be in.

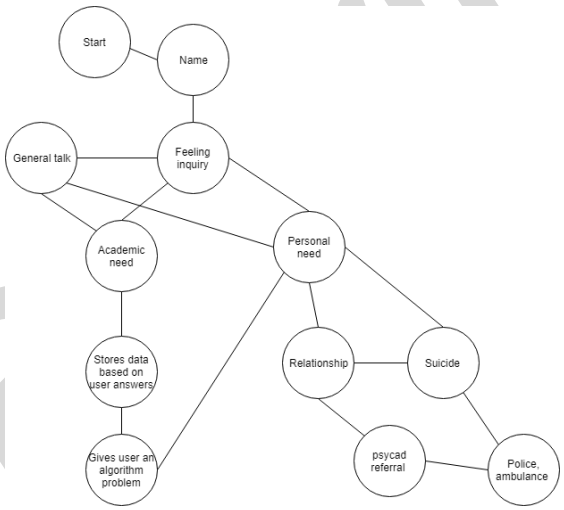

Fig. 2. Various States of Lecturer-Chatbot

In this paper, we discussed the difficulties that students face in Introductory Programming. We have also developed an interactive AI chatbot (called Lecturer-Chatbot) that students can interact with on some of the academic issues they face. Lecturer-Chatbot can provide many practice algorithms and their solutions to students. Additionally, Lecturer-Chatbot provides personal advises

[1] Centre for Psychological Services and Career Development

to students. An evaluation of Lecturer-Chatbot showed that there is a need for such a tool in aiding students through their university life. 60% of respondents said programming was difficult, which is a cause for concern. So by implementing a tool like Lecturer-Chatbot we may be able to solve this problem.

Keywords: Novice programmers, Introductory programming, Chatbot, Automated Advisory, Artificial Intelligence in education.

References

[1] Neil CC Brown and Amjad Altadmri. Novice java programming mistakes: large-scale data vs. educator beliefs. *ACM Transactions on Computing Education (TOCE)*, 17(2):7, 2017.

[2] José Figueiredo and Francisco José García-Peñalvo. Building skills in introductory programming. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 46–50. ACM, 2018.

[3] Ruiqi Deng and Brent W Ritchie. International university studentsâ^{TM} travel risk perceptions: an exploratory study. *Current Issues in Tourism*, 21(4):455–476, 2018.

[4] Masud Jariah, AR Husniyah, P Laily, and Sonya Britt. Financial behavior and problems among university students: Need for financial education. *Journal of Personal Finance*, 3(1):82–96, 2004.

[5] Roy L Moore and George P Moschis. The role of family communication in consumer learning. *Journal of Communication*, 31(4):42–51, 1981.

[6] Susanne Friese and Hal Koenig. Shopping for trouble. *Advancing the Consumer Interest*, pages 24–29, 1993.

[7] Gustavo López, Luis Quesada, and Luis A Guerrero. Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *International Conference on Applied Human Factors and Ergonomics*, pages 241–250. Springer, 2017.

[8] Phillip A Laplante. *What every engineer should know about software engineering*. CRC Press, 2007.

[9] Ariel Rosenfeld, Abejide Ade-Ibijola, and Sigrid Ewert. Regex Parser II: Teaching regular expression fundamentals via educational gaming. In *Annual Conference of the Southern African Computer Lecturers' Association*, pages 99–112. Springer, 2017.

[10] Carlos I Chesnevar, Maria P González, and Ana G Maguitman. Didactic strategies for promoting significant learning in formal languages and automata theory. *ACM SIGCSE Bulletin*, 36(3):7–11, 2004.

[11] M Vijayalaskhmi and KG Karibasappa. Activity based teaching learning in formal languages and automata theory-an experience. In *2012 IEEE International Conference on Engineering Education: Innovative Practices and Future Trends (AICERA)*, pages 1–5. IEEE, 2012.