

# Python-Bot: A Chatbot for Teaching Python Programming

Chinedu Wilfred Okonkwo *Member, IAENG*, and Abejide Ade-Ibijola *Member, IAENG*

**Abstract**—The applications of Artificial Intelligence (AI) in Education is growing continuously. One of the recent AI applications introduced to Education is the Chatbot technology, used to support teaching and administrative tasks. A long standing problem in Computer Science Education (CSE) is how to improve program comprehension for novice programmers. So far, CSE has not seen any Chatbot for supporting the teaching of programming. This paper proposed a Chatbot named Python-Bot that helps novice programmers to understand Python's basic syntactic structures and semantics. Python-Bot was implemented on an existing Chatbot API (called SnatchBot) and an evaluation was performed to prove the effectiveness of Python-Bot. The results suggest that Python-Bot can support novice programmers in the comprehension of programs written in Python. Python-Bot can be found at: <https://tinyurl.com/pythonbotchatbot2020>.

**Index Terms**—Chatbot, Python Programming, Novice Programmers, Computer Science Education, Program Comprehension, SnatchBot.

## I. INTRODUCTION

NOVICE programmers struggle to comprehend computer programming [1, 2], having to learn various set of skills simultaneously [3] including programming constructs (such as arrays, loops, conditional statements, etc.) [4, 5], and debugging process [6]. One of the challenging issues contributing to this problem is inadequate support provided to the students [7, 8]. Studies have shown that the introductory programming courses have a high drop-out rate [7, 9], due to lack of adequate learning aids that can support the students [10, 11]. A large number of students are admitted into different institutions annually, resulting in an increase in the ratio of students per teacher [12, 13]. This decreases students' learning opportunities [7, 14], as a result, there is need to improve the teaching of this subject by any innovation available [15, 16], in this case — *Chatbots*.

Programming is a challenging subject to teach [17, 18], and it is one of the required skills in the Fourth Industrial Revolution (4IR) [19, 20]. Most students dislike programming subjects believing that these subjects are hard to learn [21]. Altadmri and Brown [2] revealed that students have issues in writing and designing a clear program. Thus, technological innovations, powered by

This work is based on research supported by the National Research Foundation (NRF) of South Africa (Grant Number: 119041).

Chinedu Wilfred Okonkwo is a Postdoctoral Research Fellow in the Research Cluster on Formal Structures, Algorithms, and Industrial Applications, Department of Applied Information Systems, School of Consumer Intelligence and Information Systems, at the University of Johannesburg, Johannesburg, South Africa. e-mail: [chineduo@uj.ac.za](mailto:chineduo@uj.ac.za)

Abejide Ade-Ibijola is an Associate Professor in the Research Cluster on Formal Structures, Algorithms, and Industrial Applications, Department of Applied Information Systems, School of Consumer Intelligence and Information Systems, at the University of Johannesburg, Johannesburg, South Africa. website: <https://www.abejide.org>, e-mail: [abejideai@uj.ac.za](mailto:abejideai@uj.ac.za).

Artificial Intelligence (AI) are needed as interventions in addressing this problem [22, 23].

One way to approach this is the adoption of technological tools that aid the teaching and learning of programming [1]. There exist a number of Chatbots in Education, but to the best of our knowledge, none has been designed to teach Python programming. Some existing Chatbots in Education include Chatbots for: students' support [10], asking and answering queries [24], mobile language learning [25], answering university related questions [26], finding paths on a university campus [27], and so on. The introduction of Chatbots in Education can be seen as a significant advancement and an innovative solution for improving learning of difficult subjects [10, 11]. Using Chatbots can change the pace at which students can learn without being compelled to [28]. The use of Chatbots in learning enables students to be more agile, customised, engaged and inspired [10, 29].

This work presents a chatbot for teaching programming, called Python-Bot. This chatbot is designed to help students in learning Python programming. Python-Bot explains programming concepts to students, provides supports for booking appointments with lecturer or tutors, and supplies the students with a bank of pre-defined programming problems. SnatchBot, a platform for creating chatbots, was used to create Python-Bot, and the new chatbot is deployed as a mobile web application. SnatchBot provides the user with an Interface unit, a Knowledge unit (for processing user messages), and a Message bank (containing predefined answers). The knowledge unit comprises of the query and Artificial Intelligence Markup language (AIML) interfaces. AIML enables the Chatbot to check the input of the user with the predefined messages in the message bank in order to provide a matching response to the user. This interaction can be deployed to the social media network through the Integration unit. The operation flow of SnatchBot is shown in Figure 1.

The following are the contributions of this work. We have:

- 1) developed a chatbot for novice programmers to support the learning of Python programming, and
- 2) conducted an evaluation of how helpful the chatbot is to students.

The rest of this article is structured as follows. Section II presents background and related work. Section III describes the design of Python-Bot and Section IV explains the implementation and results. Section V discusses the evaluation and Section VI summarises the scope of Python-Bot. Section VII is the conclusion and future work.

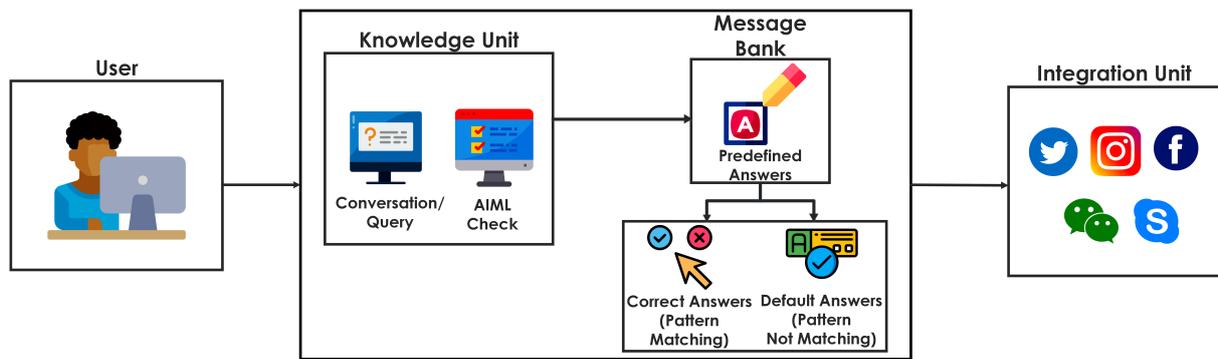


Fig. 1: The flow of SnatchBot

## II. BACKGROUND AND RELATED WORK

In this section, the background of Chatbots with much emphasis on the applications thereof in Education is presented. Chatbot system is an AI technology being used in various aspects of life activities including Healthcare [30], Finance [31], Social interactions [32] and Education [33, 11]. Several organisations have developed APIs or platforms for creating Chatbots, such as: IBM Watson [34], Microsoft Cortana, Apple Siri [10, 35] and Facebook M [36].

### A. What is a Chatbot?

Chatbot system is a smart conversational agent or dialog system that promotes interactive learning [37]. An intelligent agent able to communicate with a user in order to answer a number of questions and provide the correct response [10]. Modern dialog systems usually adopt a hybrid design combining handcrafted states and procedures with computational machine learning algorithms [38, 39, 40]. The architecture involves a collection of interrelated subsystems that are closely linked to the development of the microservice architecture [41, 42]; that is, small interconnected components that communicate through a lightweight protocol over the Internet [43]. These systems depend on an initial Natural Language Understanding (NLU) module to determine intentions and extract the appropriate data from the user queries they take as input [44]. The need for Educational Chatbot systems has increased with the extensive use of computers and mobile devices with the ability to interact, and the willingness of their manufacturers to deliver natural language interfaces. More so, Knill *et al.* [45] found in the Education domain that Chatbot technology amplifies the learning system.

### B. Chatbot as a Teaching and Learning Tool

Technology offers many learning opportunities, and the internet has tremendous potential [46]. The introduction of AI in Education is changing the learning process [10]. Students learn and absorb information through AI tools like Chatbot [33, 47]. Today's teachers are able to offer the learners with a customized learning environment thanks to AI. Researchers have developed a number of learning tools to improve the Education system [48]. Chatbots, as conversational agents, act as game changers in the innovative learning world [10, 49], and are built to enhance the interaction and collaboration among the students and their courses [50, 51]. Several studies have emphasized

that Chatbots are useful for learning [10, 33, 52]. Clarizia *et al* [10] stated that the Chatbot system is one of the key innovative solutions that fills the technological and educational gap.

Knill *et al* [45] noticed that with a Chatbot conversation, teachers could observe the type of questions that students ask, the subject's problem areas, and measure student learning abilities. As the development of Chatbot technology advances, there are more cases of use in almost every conceivable industry. Chatbots have recently started to be introduced in Education, and the expected effects are extensive. Traditionally, the adaptation of new technologies to Education is a little slower, but the innovation trend in the 4IR era is creating rapid changes. Educators and administrators who understand Chatbots' importance will discover an easy and economical way to engage more students and simplify teaching and learning processes. Chatbots plays vital role in Education. Some of these roles include providing the students with an interactive learning environment, support to teachers, spelling and grammatical checking [10]. Chatbots also serve as a means of social learning [53], providing personal advice [54] used as online assessments [55], replying to students' queries, and checking their assignments [24].

### C. Related Work

We could not find a Chatbot specifically designed to teach Python programming, however, related works (similar bots) in this area are as follows:

**Questions and Answers** Clarizia *et al* [10] created a Chatbot prototype. The model uses Natural Language Processing (NLP) method to detect students' questions and provide the accurate answers to the student. Sharod *et al* [24] developed a Chatbot system for answering queries. The system can automatically respond to a query from a user on behalf of an individual, for learning purposes. Ranoliya *et al* [26] designed a Chatbot that uses the Artificial Intelligence Markup Language and Latent Semantic Analysis to provide correct responses to questions based on the FAQ dataset.

**Language Concepts** Pham *et al* [25] built an English learner Chatbot. The system is set up on mobile devices for communication with users. The Chatbot will automatically alert students to learn and propose solutions to multiple-choice questions. It can also assist users

learn new language concepts. Zhao *et al* [56] created a learning tool called APIHelper that helps novice Android programmers learn how to use the APIs. Zaw *et al* [57] developed a Java programming learning tool that provides different kinds of exercise problems for study coverage at different levels.

**Administrative and Services** In a study to discuss learning environment, Hein *et al* [58], developed a Chatbot that automatically responds to a student question on behalf of the academic staff on the delivery of the Education system. The Chatbot will play the role of a digital assistant, providing tertiary institutions with ways to enhance their present services, develop new innovative products, and minimize labour costs. Some of the AI techniques used in building the system include text classification and named entity recognition. Röhrig and Heß [59] have developed a mobile manipulator platform that can enable human-robot collaboration services, and mobile manipulation can be used in research and education [60]. Information technology enables the creation and functioning of collaborative applications [61].

**Evaluation Platform** Introducing Computer Science concepts to high school students in an innovative way, Nicol *et al* [12] presented a Chatbot designed platform using pattern matching, finite state automata, and state-of-the-art lemmatization methods to offer students with automated formative evaluation. To improve analytical thought, Mor *et al* [62] suggested a Chatbot model and architecture based on skill training and evaluation. The Chatbot provide a conversational interface to simulate a lab experience and setting based on activities.

#### D. Chatbot Development Platforms

A Chatbot Development Platform is a tool/application that can be used to create a Chatbot. These platforms support software developers by building a stream, offering Machine Learning capabilities and integration with APIs to add more features to your Bot. Several platforms can be used to build Chatbots quickly and easily including Microsoft's Azure Bot Framework, Google's Dialogflow, Amazon's Alexa, Facebook's Bots for Messenger, ManyChat, GupShup, Chatfuel, Botsify [25], and SnatchBot. The SnatchBot platform allows developers to create interactions from Chatbot that address a range of business issues. Python-Bot was developed using the Snatchbot platform concepts.

#### E. The Gap

From the literature, tools are needed to support students during their learning process. Various types of Chatbot have been designed and created for different activities in Education. No attempt has been made to create a Chatbot for teaching Python programming. If such a Chatbot is developed, it should help instructors save time in responding to student queries as well as help students learn how to program in Python. This paper proposed a Chatbot for teaching Python programming.

#### F. Motivation

The software engine that drives the 4IR is AI [63, 64]. Technology advancement through the Machine Learning

system enables digital revolution [65, 66]. As a consequence, machines no longer capture explicit knowledge (where humans can explain a sequence of relatively logical steps) but are now creating a tacit understanding the intuitive, know-how integrated into the human mind. The introduction of AI technology in Education will bring significant improvements, allow fresh possibilities to teach and learn anywhere at any time, and provide fresh instruments and techniques for improving learning results and supporting innovative teaching and learning [67]. The skills required to build and operate AI technologies are based on the knowledge of programming. Unfortunately, Computer Programming courses were seen by students as difficult to learn and shy away from them. There is therefore a need to simplify the teaching and learning methods of Computer Programming in order to motivate students' interests in programming. Python-Bot is designed to provide an easy-to-use Python programming learning platform for students.

### III. CHATBOT DESIGN

In this section, the system design of Python-Bot with an example of a working algorithm are presented. The development of Python-Bot follows SnatchBot platform principles (mainly drag-and-drop, code-free design). SnatchBot provides an in-built editor that can be used to develop a simple or complex conversation with action buttons and translations. Also, SnatchBot allows you to create many interactions in relation with activities of the Chatbot. During our development, we created *interactions* that are used to describe the subjects. For example, in building an interaction between the Chatbot and the user to introduce themselves, we created an interaction called Introduction. In a similar way, many other *interactions* were created including Name, Rename, Goodbye, Contents, Concepts, Exercises, etc. The *subjects* are the predefined response patterns from the Chatbot including messages, videos, graph, etc. With SnatchBot, we created a design consisting of *interactions* and *subjects*. Figure 2 shows some part of the building scheme of Python-Bot.

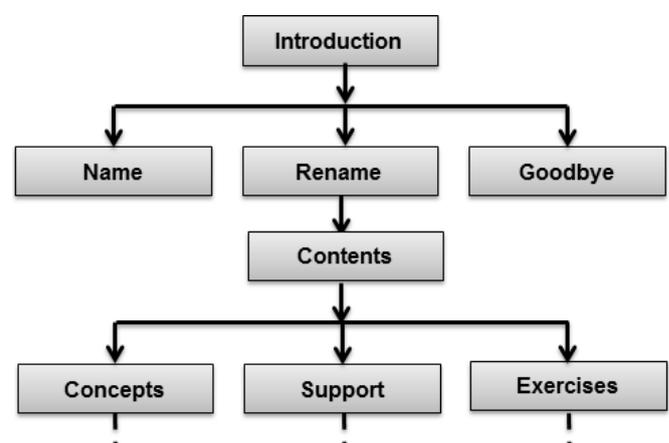


Fig. 2: The building Scheme of Python-Bot

In Python-Bot, we have 50 interactions. With SnatchBot's capabilities, interactions are linked together locally and globally for efficient operations. Local

connections can only work on a specific interaction, while global connections can work on all interactions. All connections are in the form of a logical statement:  $\text{if } x \text{ then } y \text{ else } K$ . The algorithm below shows the main activity in the information decision section of the Python-Bot.

---

**Algorithm 1:** Python-Bot Information Decision
 

---

```

Input: Topic (this is the topic deduced from the text
entered by the user);
if Topic Matches Message_Bank then
  return predefined response
  ( $\exists$ :  $Topic \in Response$ ) // returns predefined
  responses such that the response is in the
  collection of responses for the topic;
else
  return default answer (select a topic);
end

```

---

The Python-Bot's functional decision in response to user queries is shown in Figure 3.

#### IV. IMPLEMENTATION AND RESULTS

Python-Bot was developed with the SnatchBot API, using its predefined tools that is driven by various NLP algorithms. SnatchBot is a website for Chatbot creation. It helps you to handle end-to-end bot management from creating and reviewing flows to publishing and reporting performance. It communicates with many social networking and communication sites, including Facebook, Slack, Skype, Viber, Twitter, etc. Another interesting feature of SnatchBot is the Channel. The SnatchBot Channel is made up of an integrated Application Programming Interface (API). This allows Chatbots to be deployed on websites and social networks. Through the SnatchBot Channel, Python-Bot is placed on a website. Students may use their smartphones or laptops to connect to the website to chat with the Chatbot, making Python-Bot accessible to novice programmers for learning how to program in Python.

Python-Bot starts an operation with the introduction of the Bot and the user. It describes itself and seeks the student's willingness to engage in conversation, as well as recognition of the student's name. The Bot receives the student's query, prepares it and answers the student by sending him/her predefined textual insights into the subject. Also, it provides some examples of the implementation of Python algorithms and outputs showing the willingness of the user to participate in a learning conversation and the ability of the Bot to provide the correct answers. A sample discussion with Python-Bot is shown in Figures 4a and 4b.

SnatchBot provides a Natural Language Processing (NLP) capability that enables Python-Bot to understand and function appropriately on the context of a user's response or query. This is done by breaking a sentence into basic parts including entity and intent. While the entity represents the object, the purpose of the sentence is covered by the

intent. For the NLP module to achieve particular goals, we had to define the types of entity and intent it would recognise. In other words, we used some predefined NLP models in SnatchBot for each of the entity and intent that we want Python-Bot to identify. The models use Machine Learning to make decisions in any new sentence they encounter, using the samples provided as the basis. In this work, two NLP models were used to provide support to learners, including model that detects the date (entity) and one that detects the time (intent) for appointment. During a conversation with Python-Bot, a student may need the attention of a tutor or lecturer requiring a meeting.

These models help to generate the appropriate date and time for a meeting based on the predefined dates and times available. In addition, an email message will automatically be sent to the tutor/lecturer to notify him/her of the meeting. If the meeting is confirmed by the tutor/lecturer, the student can have a face-to-face discussion with the tutor/lecturer. Figures 5a and 5b demonstrates the Python-Bot's NLP capability of creating a meeting appointment to provide the learners with physical support.

#### V. SCOPE OF PYTHON-BOT

This section summarises the scope of Python-Bot's operations covering main Chatbot functionalities; including how it introduces itself to students, shares programming concepts and provides support capability.

**Introduction** Python-Bot begins by first introducing itself and asking the user to introduce himself or herself.

This enables the Chatbot to keep record of the user name and conversation period (starting time, ending time and duration of the conversation).

**Programming Concepts** The Python-Bot provides detail explanation of Python programming concepts including Data Types, Expressions, Control Statements, Strings, Text Files, Lists, Dictionaries, Tuples, Problem Solving, and Algorithms.

Python-Bot can also discuss algorithms that are free of programming language syntax. Examples of the some content shared by Python-Bot to assist novice programmers in understanding the grammatical structure of if statements is:

```

IF (logical condition(s)) THEN
  Statement block 1
ELSE
  Statement block 2
END IF

```

Python-Bot proceeds to explain to the user the process of writing or constructing an algorithm. For instance, to write an algorithm to find the average of two numbers. It displays an explains these steps:

---

Algorithm: Find the Average of Two Numbers

---

```

Step 1: Add the two numbers and save this
       result in a variable.
Step 2: Divide this new number by the
       number or constant 'two',

```

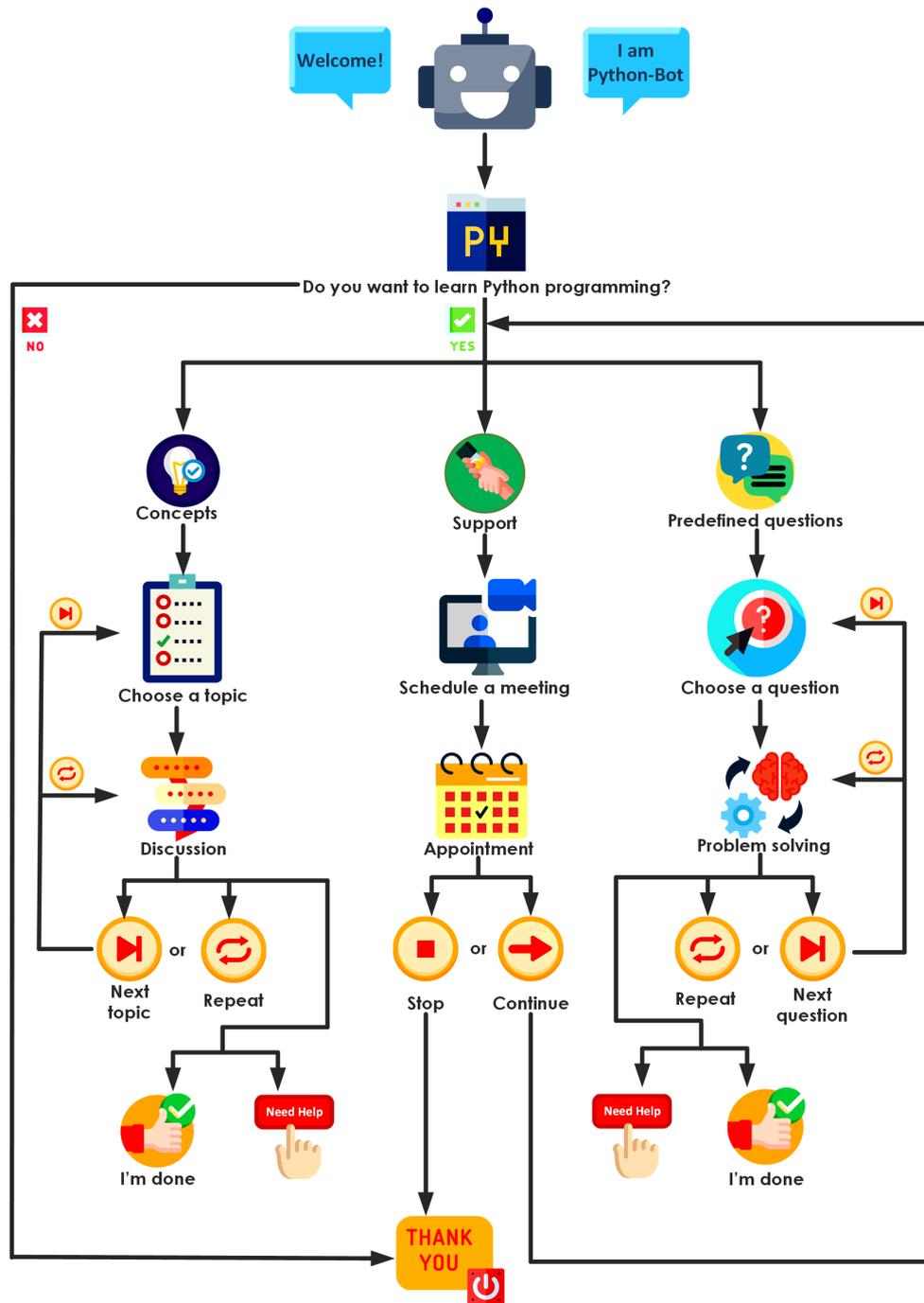


Fig. 3: The Flow of Python-Bot

and save this result in a variable.

Step 3: Provide this number (result from step 2) to the rest of the program (or print it for the user).

```
>>> example.append(10)
>>> example
{[1, 2, 10]}
```

Extend: To extend one list with another list

Other examples of conversations that Python-Bot can have includes how to append items to lists, extend lists and display these lists in Python; using the examples:

```
>>> example = [1, 2, 10]
>>> example.extend([11, 12, 13])
>>> example
{[1, 2, 10, 11, 12, 13]}
```

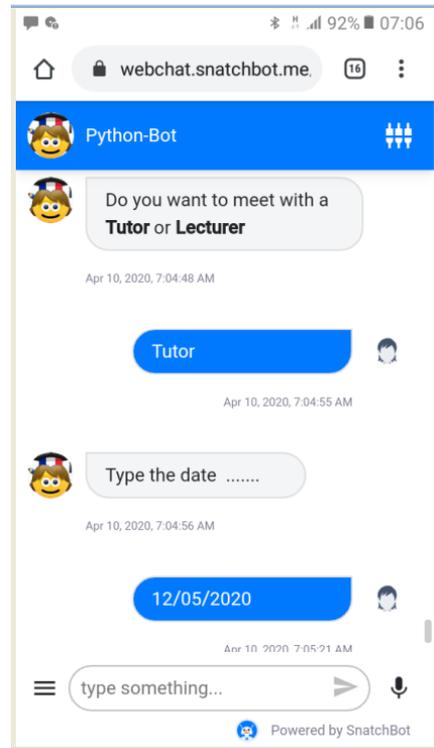
Append: To add an element into an existing list

```
>>> example = [1, 2]
```

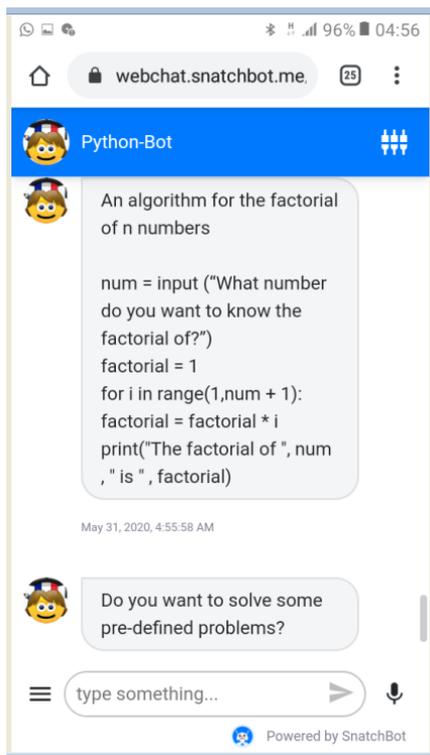
**Support Capability** Python-Bot uses the NLP capability models to provide support to the learners by enabling



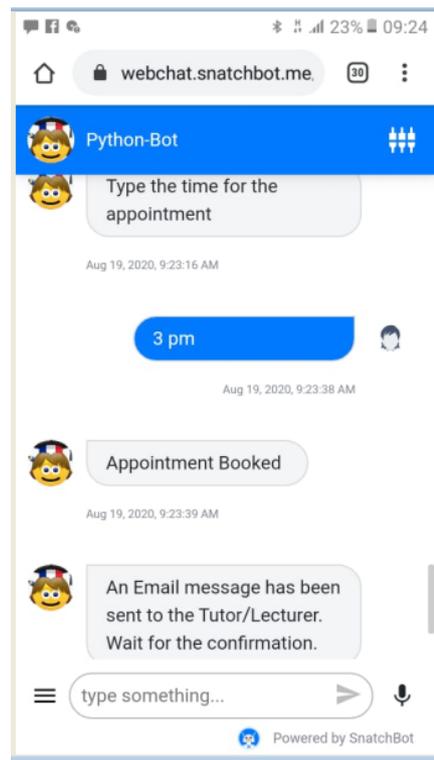
(a) Starting a conversation with Python-Bot



(a) Processing of date for a meeting



(b) Examples of Python code examples displayed to the user



(b) Processing of time for a meeting

Fig. 4: Screenshots of starting a conversation with Python-Bot

Fig. 5: Python-Bot NLP capability

them to set up a meeting with the tutor or lecturer. Accordingly, the Python-Bot will generate the date and time of the meeting with respect to the user's response to Python-Bot queries.

## VI. EVALUATION AND APPLICATION OF PYTHON-BOT

This section presents the outcomes from an online survey evaluation of the students' opinion of the use of Python-Bot and its possible usefulness.

### A. Evaluation of Python-Bot

A survey was conducted at the University of Johannesburg, South Africa, and the target audience includes the first-year students taking the Python programming introductory course as well as those who have completed the course before. This demographic was selected because they are considered to be novice programmers. The purpose of the experimental campaign was to determine:

- 1) how easy do they find the design and use of Python-Bot, and
- 2) if using Python-Bot improves students' program comprehension.

A total of 205 responses were received. Approximately 73% (150) of the respondents were students who have done Python programming course. Students were asked about their perception of:

**Computer Programming Courses** This tested students' general perceptions of Computer Programming courses in order to determine their feelings regarding learning the courses. 60.5% believed that the courses are difficult, 27.3% find the courses to be easy and 12.2% were indifferent. This is shown in Figure 6a.

**Use of Python-Bot** This tested how user-friendly Python-Bot is. 73.7% believed that the tool is easy to use, 15.6% found it difficult to use and 10.7% were indifferent. See Figure 6b.

**Correct Suggestions** This tested the Python-Bot responses to determine the degree of accuracy. Overwhelming 99% of the students agreed that Python-Bot suggestions were accurate and 2% disagreed. See Figure 6c.

**Making Learning Easy** The effect of using the Python-Bot on the programming skills of the students was checked. 81.4% of the students believed that using Python-Bot would make learning Python programming easier, 4.5% did not believe and 14.1% were not precise of their opinion. See Figure 6d.

The results showed that a large number of students saw Computer Programming courses as difficult to learn, which means that students need the support of learning aids. The design of Python-Bot was user-friendly, making it easy for students to use. By providing accurate suggestions, the use of Python-Bot makes learning Python programming easier for students and thus improves their programming skills.

### B. Application of Python-Bot

As a result of COVID-19 pandemic in the early months of 2020, the schools are closed and the students were in their respective homes. There are approximately 160 first year students in the Department of Applied Information Systems at the University of Johannesburg, South Africa, taking the Python introductory programming course for the first time. Python-Bot was used to administer the teaching and learning of the course content. With the help of Python-Bot support capability, students were able schedule appointments to have discussions with a tutor or

the lecturer for assistance when needed. From the students' perception, the use of Python-Bot was helpful in learning the subject.

## VII. CONCLUSION AND FUTURE WORK

### A. Conclusion

In this paper, we presented a Chatbot system called Python-Bot that assists novice programmers in learning how to program in Python. Python-Bot was created using Snatch-Bot API. The Chatbot can be deployed on the website and on social networking platforms. The evaluation results show that a vast majority of students currently taking a Python programming course and those who did the course before agreeing that the tool is user-friendly, simplifies programming logic and enhances their Python programming skills. Furthermore, this tool was used to support the teaching of introductory programming to students during the COVID-19 pandemic.

### B. Future Work

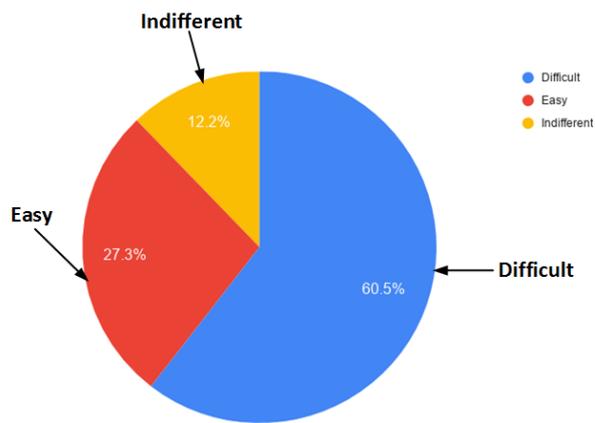
There are several programming language courses in the field of computer science. In the future, this tool will be extended to include the teaching and learning of these subjects. For many other undergraduate courses, the creation of Chatbot systems can help in assisting the online learning platform.

## ACKNOWLEDGEMENT

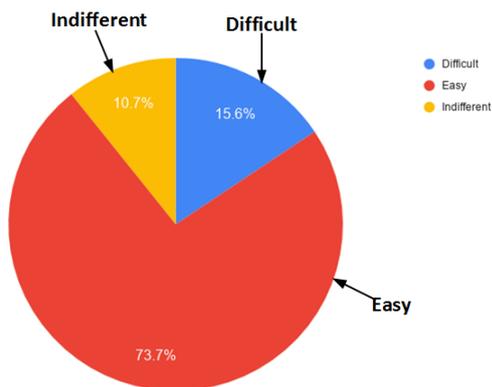
Any Opinion, findings and conclusions or recommendations expressed in this material are those of authors and therefore the NRF does not accept liability in regard thereto. We also acknowledge the contribution made by Nikita Patel (Research Assistant in the Formal Structures Research Cluster) in drawing the rich picture in this paper.

## REFERENCES

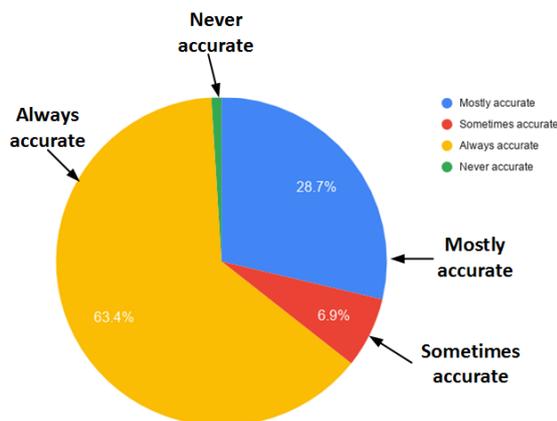
- [1] Abejide Ade-Ibijola. Syntactic generation of practice novice programs in python. *Communications in Computer and Information Science (CCIS)*, Springer, pages 158–172, 2018.
- [2] Amjad Altadmri and Neil CC Brown. 37 million compilations: Investigating novice programming mistakes in large-scale student data. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 522–527. ACM, 2015.
- [3] Tiantian Wang, Xiaohong Su, Peijun Ma, Yuying Wang, and Kuanquan Wang. Ability-training-oriented automated assessment in introductory programming course. *Computers & Education*, 56(1):220–226, 2011.
- [4] Amos Baker, Jinghua Zhang, and Elvira R Caldwell. Reinforcing array and loop concepts through a game-like module. In *2012 17th International Conference on Computer Games (CGAMES)*, pages 175–179. IEEE, 2012.
- [5] Jinghua Zhang, Mustafa Atay, Elvira R Caldwell, and Elva J Jones. Visualizing loops using a game-like instructional module. In *2013 IEEE 13th International Conference on Advanced Learning Technologies*, pages 448–450. IEEE, 2013.



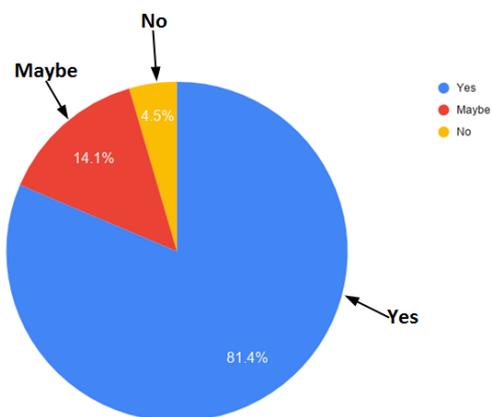
(a) Computer Programming is difficult



(b) Using Python-Bot is easy



(c) Python-Bot makes correct suggestions



(d) Python-Bot makes learning easy

Fig. 6: Evaluation results

[6] Basma S Alqadi and Jonathan I Maletic. An empirical study of debugging patterns among novices programmers. In *Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education*, pages 15–20. ACM, 2017.

[7] Sean B Eom and Nicholas Ashill. The determinants of students’ perceived learning outcomes and satisfaction in university online education: An update. *Decision Sciences Journal of Innovative Education*, 14(2):185–215, 2016.

[8] Carmen Richardson and Punya Mishra. Learning environments that support student creativity: Developing the scale. *Thinking Skills and Creativity*, 27:45–54, 2018.

[9] Sohail Iqbal Malik. Role of ADRI model in teaching and assessing novice programmers. Technical report, Deakin University, 2016.

[10] Fabio Clarizia, Francesco Colace, Marco Lombardi, Francesco Pascale, and Domenico Santaniello. Chatbot: An education support system for student. In *International Symposium on Cyberspace Safety and Security*, pages 291–302. Springer, 2018.

[11] M Naveen Kumar, PC Linga Chandar, A Venkatesh Prasad, and K Sumangali. Android based educational chatbot for visually impaired people. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, pages 1–4. IEEE, 2016.

[12] David J Nicol and Debra Macfarlane-Dick. Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in higher education*, 31(2):199–218, 2006.

[13] Oddny Judith Solheim, Mari Rege, and Erin McTigue. Study protocol: “two teachers”: A randomized controlled trial investigating individual and complementary effects of teacher-student ratio in literacy instruction and professional development for teachers. *International Journal of Educational Research*, 86:122–130, 2017.

[14] Kate S Hone and Ghada R El Said. Exploring the factors affecting MOOC retention: A survey study. *Computers & Education*, 98:157–168, 2016.

[15] Ana Marta Aleixo, Susana Leal, and Ulisses Miranda Azeiteiro. Conceptualization of sustainable higher education institutions, roles, barriers, and challenges for sustainability: An exploratory study in portugal. *Journal of Cleaner Production*, 172:1664–1673, 2018.

[16] Sue Cherrington, Anne Macaskill, Rhian Salmon, Suzanne Boniface, Sydney Shep, and Jonny Flutey. Developing a pan-university professional learning community. *International Journal for Academic Development*, 23(4):298–311, 2018.

[17] Vishv Mohan Malhotra and Ashish Anand. Teaching a university-wide programming laboratory: Managing a c programming laboratory for a large class with diverse interests. In *Proceedings of the Twenty-First Australasian Computing Education Conference*, pages 1–10, 2019.

[18] C Jesennia, Amilkar Puris, David Benavides, et al. Recommender systems and scratch: An integrated approach for enhancing computer programming learning. *IEEE Transactions on Learning Technologies*, 2019.

- [19] TV Kondratyuk. Fourth industrial revolution: What competences are necessary for employees? *Strategic decisions and risk management*, (3):66–79, 2018.
- [20] NV Komarova, AA Zamkovoi, and SV Novikov. The fourth industrial revolution and staff development strategy in manufacturing. *Russian Engineering Research*, 39(4):330–333, 2019.
- [21] Yizhou Qian and James Lehman. Students' misconceptions and other difficulties in introductory programming: A literature review. *ACM Transactions on Computing Education (TOCE)*, 18(1):1–24, 2017.
- [22] Roger O Smith, Marcia J Scherer, Rory Cooper, Diane Bell, David A Hobbs, Cecilia Pettersson, Nicky Seymour, Johan Borg, Michelle J Johnson, Joseph P Lane, et al. Assistive technology products: a position paper from the first global research, innovation, and education on assistive technology (great) summit. *Disability and Rehabilitation: Assistive Technology*, 13(5):473–485, 2018.
- [23] Michael J Lee and Bruce Ferwerda. Personalizing online educational tools. In *Proceedings of the 2017 ACM Workshop on Theory-Informed User Modeling for Tailoring and Personalizing Interfaces*, pages 27–30. ACM, 2017.
- [24] Sharob Sinha, Shyanka Basak, Yajushi Dey, and Anupam Mondal. An educational chatbot for answering queries. In *Emerging Technology in Modelling and Graphics*, pages 55–60. Springer, 2020.
- [25] Xuan Lam Pham, Thao Pham, Quynh Mai Nguyen, Thanh Huong Nguyen, and Thi Thu Huong Cao. Chatbot as an intelligent personal assistant for mobile language learning. In *Proceedings of the 2018 2nd International Conference on Education and E-Learning*, pages 16–21. ACM, 2018.
- [26] Bhavika R Ranoliya, Nidhi Raghuvanshi, and Sanjay Singh. Chatbot for university related faqs. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 1525–1530. IEEE, 2017.
- [27] Katlego Mabunda and Abejide Ade-Ibijola. Pathbot: An intelligent chatbot for guiding visitors and locating venues. In *2019 6th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 160–168. IEEE, 2019.
- [28] Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Hsin-Min Wang. A chatbot using lstm-based multi-layer embedding for elderly care. In *2017 International Conference on Orange Technologies (ICOT)*, pages 70–74. IEEE, 2017.
- [29] Sherry Ruan, Liwei Jiang, Justin Xu, Bryce Joe-Kun Tham, Zhengneng Qiu, Yeshuang Zhu, Elizabeth L Murnane, Emma Brunskill, and James A Landay. Quizbot: A dialogue-based adaptive learning system for factual knowledge. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2019.
- [30] Kyo-Joong Oh, Dongkun Lee, Byungsoo Ko, and Ho-Jin Choi. A chatbot for psychiatric counseling in mental healthcare service based on emotional dialogue analysis and sentence generation. In *2017 18th IEEE International Conference on Mobile Data Management (MDM)*, pages 371–375. IEEE, 2017.
- [31] Jon TS Quah and YW Chua. Chatbot assisted marketing in financial service industry. In *International Conference on Services Computing*, pages 107–114. Springer, 2019.
- [32] Agnese Augello, Manuel Gentile, Lucas Weideveld, and Frank Dignum. A model of a social chatbot. In *Intelligent Interactive Multimedia Systems and Services 2016*, pages 637–647. Springer, 2016.
- [33] Petre Angheliescu and Stefan Vladimir Nicolaescu. Chatbot application using search engines and teaching methods. In *2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–6. IEEE, 2018.
- [34] Ying Chen, JD Elenee Argentinis, and Griff Weber. IBM Watson: how cognitive computing can be applied to big data challenges in life sciences research. *Clinical therapeutics*, 38(4):688–701, 2016.
- [35] Veton Kepuska and Gamal Bohouta. Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 99–103. IEEE, 2018.
- [36] Jyri Lehvä, Niko Mäkitalo, and Tommi Mikkonen. Case study: building a serverless messenger chatbot. In *International Conference on Web Engineering*, pages 75–86. Springer, 2017.
- [37] Sherry Ruan, Angelica Willis, Qianyao Xu, Glenn M Davis, Liwei Jiang, Emma Brunskill, and James A Landay. Bookbuddy: Turning digital materials into interactive foreign language lessons through a voice chatbot. In *Proceedings of the Sixth (2019) ACM Conference on Learning@ Scale*, pages 1–4, 2019.
- [38] Jason D Williams. An empirical evaluation of a statistical dialog system in public use. In *Proceedings of the SIGDIAL 2011 Conference*, pages 130–141. Association for Computational Linguistics, 2011.
- [39] David Suendermann-Oeft, Vikram Ramanarayanan, Moritz Teckenbrock, Felix Neutatz, and Dennis Schmidt. Halef: An open-source standard-compliant telephony-based modular spoken dialog system: A review and an outlook. In *Natural language dialog systems and intelligent assistants*, pages 53–61. Springer, 2015.
- [40] Filip Jurčiček, Ondřej Dušek, Ondřej Plátek, and Lukáš Žilka. Alex: A statistical dialogue systems framework. In *International Conference on Text, Speech, and Dialogue*, pages 587–594. Springer, 2014.
- [41] Sam Newman. *Building microservices: designing fine-grained systems*. "O'Reilly Media, Inc.", 2015.
- [42] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. Microservices: yesterday, today, and tomorrow. In *Present and ulterior software engineering*, pages 195–216. Springer, 2017.
- [43] Carlos M Aderaldo, Nabor C Mendonça, Claus Pahl, and Pooyan Jamshidi. Benchmark requirements for microservices architecture research. In *2017 IEEE/ACM 1st International Workshop on Establishing the Community-Wide Infrastructure for Architecture-Based Software Engineering (ECASE)*, pages 8–13.

- IEEE, 2017.
- [44] Raymond ST Lee. Natural language processing. In *Artificial Intelligence in Daily Life*, pages 157–192. Springer, 2020.
- [45] Oliver Knill, Johnny Carlsson, Andrew Chi, and Mark Lezama. An artificial intelligence experiment in college math education. *Preprint available at <http://www.math.harvard.edu/knill/preprints/sofia.pdf>*, 2004.
- [46] Anke Berns, José Miguel Mota, Ivan Ruiz-Rube, and Juan Manuel Dodero. Exploring the potential of a 360 video application for foreign language learning. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 776–780. ACM, 2018.
- [47] György Molnár and Zoltán Szüts. The role of chatbots in formal education. In *2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 197–202. IEEE, 2018.
- [48] Aliane Loureiro Krassmann, João Marcos Flach, Anita Raquel Cestari da Silva Grando, Liane Margarida Rockenbach Tarouco, and Magda Bercht. A process for extracting knowledge base for chatbots from text corpora. In *2019 IEEE Global Engineering Education Conference (EDUCON)*, pages 322–329. IEEE, 2019.
- [49] Damla Ezgi Akcora, Andrea Belli, Marina Berardi, Stella Casola, Nicoletta Di Blas, Stefano Falletta, Alessandro Faraotti, Luca Lodi, Daniela Nossa Diaz, Paolo Paolini, et al. Conversational support for education. In *International Conference on Artificial Intelligence in Education*, pages 14–19. Springer, 2018.
- [50] Florian Daniel, Maristella Matera, Vittorio Zaccaria, and Alessandro Dell’Orto. Toward truly personal chatbots: on the development of custom conversational assistants. In *Proceedings of the 1st International Workshop on Software Engineering for Cognitive Services*, pages 31–36. ACM, 2018.
- [51] Bertrand David, René Chalon, Bingxue Zhang, and Chuantao Yin. Design of a collaborative learning environment integrating emotions and virtual assistants (chatbots). In *2019 IEEE 23rd International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 51–56. IEEE, 2019.
- [52] Martha C Pennington and Pamela Rogerson-Revell. Using technology for pronunciation teaching, learning, and assessment. In *English Pronunciation Teaching and Research*, pages 235–286. Springer, 2019.
- [53] George Suciu, Adrian Pasat, Teodora Uşurelu, and Eduard-Cristian Popovici. Social media cloud contact center using chatbots. In *International Conference on Future Access Enablers of Ubiquitous and Intelligent Infrastructures*, pages 437–442. Springer, 2019.
- [54] Ismail Mohammed and Abejide Ade-Ibijola. Lecturer’s apprentice: A chatbot for assisting novice programmers. In *IEEE International Multidisciplinary Information Technology And Engineering Conference (IMITEC)*, pages 74–81. IEEE, 2019.
- [55] Luciana Benotti, Mara Cecilia Martnez, and Fernando Schapachnik. A tool for introducing computer science with automatic formative assessment. *IEEE Transactions on Learning Technologies*, 11(2):179–192, 2017.
- [56] Jingjing Zhao, Tao Song, and Yuxia Sun. Apihelper: Helping junior android programmers learn api usage. *IAENG International Journal of Computer Science*, 47(1):pp 92–97, 2020.
- [57] Khin Khin Zaw, Win Zaw, Nobuo Funabiki, and Wen-Chung Kao. An informative test code approach in code writing problem for three object-oriented programming concepts in java programming learning assistant system. *IAENG International Journal of Computer Science*, 46(3):pp 445–453, 2019.
- [58] Ho Thao Hien, Pham-Nguyen Cuong, Le Nguyen Hoai Nam, Ho Le Thi Kim Nhung, and Le Dinh Thang. Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support. In *Proceedings of the Ninth International Symposium on Information and Communication Technology*, pages 69–76. ACM, 2018.
- [59] Christof Röhrig and Daniel Heß. Omniman: A mobile assistive robot for intralogistics applications. *Engineering Letters*, 27(4):pp 893–900, 2019.
- [60] Rainer Bischoff, Ulrich Huggenberger, and Erwin Prassler. Kuka youbot-a mobile manipulator for research and education. In *2011 IEEE International Conference on Robotics and Automation*, pages 1–4. IEEE, 2011.
- [61] Mario Anzures-García, Luz A Sánchez-Gálvez, Miguel J Hornos, and Patricia Paderewski-Rodríguez. A knowledge base for the development of collaborative applications. *Engineering Letters*, 23(2):pp 65–71, 2015.
- [62] Enric Mor, Francesc Santanach, Susanna Tesconi, and Carlos Casado. Codelab: Designing a conversation-based educational tool for learning to code. In *International Conference on Human-Computer Interaction*, pages 94–101. Springer, 2018.
- [63] World Economic Forum. Shaping the future of technology governance: Artificial intelligence and machine learning, 2020. <https://www.weforum.org/platforms>. Accessed on 14/06/2020.
- [64] Bryan Edward Penprase. The fourth industrial revolution and higher education. In *Higher education in the era of the fourth industrial revolution*, pages 207–229. Springer, 2018.
- [65] Don Tapscott. *The digital economy: Promise and peril in the age of networked intelligence*, volume 1. McGraw-Hill New York, 1996.
- [66] Haluk Demirkan, James C Spohrer, and Jeffrey J Welser. Digital innovation and strategic transformation. *IT Professional*, 18(6):14–18, 2016.
- [67] Wayne Holmes, Maya Bialik, and Charles Fadel. *Artificial Intelligence In Education: Promises and Implications for Teaching and Learning*. Center for Curriculum Redesign, Boston, MA, USA, 2019.