IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 8, Issue 8, August 2020

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

QuizBot: A Dialogue-based Pedagogical Agent for Dynamic Generation of Questions Based on Student Performance

D M Vasuki[1], Dr. Krishnan R[2]

PG Student, Dept. of CSE., Dayananda Sagar College of Engineering, Bangalore, Karnataka, India[1]

Professor, Dept. of CSE., Dayananda Sagar College of Engineering, Bangalore, Karnataka, India[2]

ABSTRACT: Despite the impracticable proposals that emerging technologies bring to education, modern learning environments such as MOOCs or Webinars still suffer from satisfactory effective awareness and effective feedback mechanisms. This will often result in low engagement or desertion. Artificial Conversational Agents hold the grounds to ease the learner's isolation, due to the recent accomplishments of Machine Learning. Yet, a method that gives effective feedback remains undelivered. Our project moves towards this objective, implementing a smart conversational agent, QuizBot a dialogue-based pedagogical agent for dynamic generation of questions based on student performance which leverages the supervised Smooth Inverse Frequency (SIF) algorithm for finding similarities between correct answer and answer given by the user. And the DASH model for question sequencing. QuizBot is implemented for students to learn the programming language better.

KEYWORDS: Chatbot, Reinforcement learning, Smooth Inverse frequency, Sentence vectors, question sequencing.

I. INTRODUCTION

A chatbotis a software that carries out a dialog via aural or textual methods. Such software is usually planned to convincingly simulate how humans will behave as a conversational partner. The chatbots are made use of in dissertation structures for many applied motives composed of either client amenity or fact attainment. Chatbots provide an interactive learning environment to study, assess to their degree of knowledge, and help them improve in areas they are weak in. Intelligent tutoring systems help students learn in a regular chat conversation, but which provides useful knowledge.

Reinforcement learning is a branch of Machine Learning in which how software agents have to take actions in a surrounding to get the maximum idea of an increasing reward is concerned.A negotiator takes actions in an atmosphere, that is construed into a reward and depiction of the state, which is sent back to the negotiator.Word embeddings have been developed and extensive in Natural Language Processing.Text similarity is usually computed by first embedding the words in the two short texts and later computing the cosine similarity between those two short texts.The similarity methods used to compute textual similarities are baselines, Word Mover's Distance, and Smooth Inverse Frequency. In this paper, the reinforcement learning approach used to build the DASH environment for question sequencing and for finding the textual similarities to compare answers given by students and actual answers, a supervised smooth inversed frequency algorithm is used. These two concepts combined helps students learn better and gain maximum knowledge, in the python programming language.

II. RELATED WORK

The papers referred are categorized based on different applications of a chatbot, for example, they are used as intelligent tutoring systems, customer service, messaging apps, health care, etc., the platform used to implement the chatbot, response generation, and answer grading methods, text classification.

Chatbot developed as an intelligent tutoring system:
A programming course uses Canvas LMS, a massive open online course (MOOC) platform, to build the chatbot[1]. An intelligent pedagogical agent is built grounded on Reinforcement Learning collective with sentiment analysis[2]. The knowledge demonstrating stage determines how acquaintance is denoted and deposited in the acquaintance database. Both the dictionary used by the instructor and the directive in which views are offered in the conversational flow stage.

The platform used to implement the chatbot:
The chatbot is built with the help of IBM Watson API [6]. The paper outfits chatbot based on AIML and has binary core parts – an android application and a server platform. The proposed text-mining-based discourse system exploits the procedure of textual data analysis[7]. The main purpose of this paper is to provision researchers to discover various ways to create a user-friendly client for necessary fields. It uses the Python Framework to create the chatbot[9].

Methods for response generation and answer grading:
If there are obscurities or mislaid data in the request, the structure should enquire for explanation or the mislaid info. In such a discourse system, there can be a piece of precedent information about the domain[10]. This paper provides a method for response generation using a knowledge database. A dissolute, meek, and great-performance brief answer grading system is presented[11].

Method for text classification:
This work provides a simple method to sentence embeddings, based on the discourse vectors obtained in the random walk model for producing text[12].

III. PROPOSED METHODOLOGY

The intelligent tutoring system proposed in this work helps students in learning the python programming language. It uses reinforcement learning to implement the DASH model[13] and the SIF algorithm to compute sentence similarities[12].

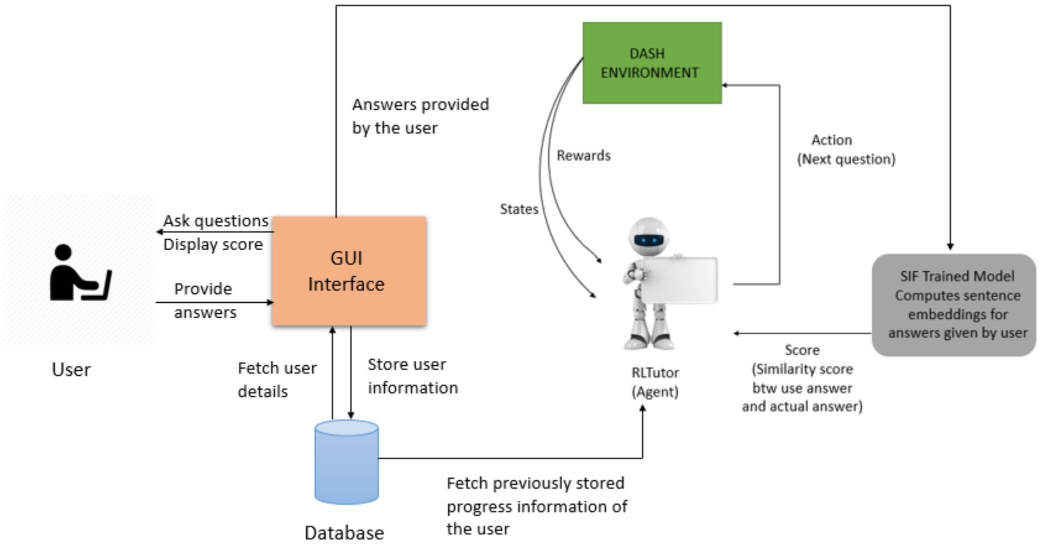

DASH ENVIRONMENT
Answers provided by the user
Rewards
Action (Next question)
States
Ask questions
Display score
GUI Interface
Provide answers
User
SIF Trained Model Computes sentence embeddings for answers given by user
Score (Similarity score btw use answer and actual answer)
RLTutor (Agent)
Fetch user details
Store user information
Database
Fetch previously stored progress information of the user

Figure 1: Software architecture of Quizbot

The software architecture of QuizBot depictsthe following:

1. Whenever the user answers a question the agent will take an action and there is a transition in the state, and a reward is given to the user as a point for each question answered correctly.
2. The state and the reward are fed back to the agent and the agent will take the next action by selecting an action from a pool of available actions.
3. In the case of Quizbot, the action is to select the next question to be answered by the user from a pool of available questions in data.
4. When this next question is asked to the user and the user provides an answer. There is a transition in state and then the reward is provided.
5. All these transitions in state and reward obtained are interpreted in the DASH environment.
6. The architecture also contains the SIF trained model, the role of this model is to compute the sentence embeddings of the answer provided by the user.
7. This model calculates the similarity score between the answer given by the user, which is fed into the model and the actual answer whose sentence vector is present in the model.

8. The RLTutor agent repeats the question for which the the answer was wrong after a few numbers of questions is been asked, based on the recall likelihood for that question.

Algorithms:

Algorithms used for the design of QuizBot are:

1. Supervised Smooth Inverse Frequency(SIF)
2. DASH(Difficulty, Ability, Study history)

A. Supervised Smooth Inverse Frequency(SIF)

The sentence embedding is defined as follows:

Given a sentence s, we do a MAP estimation of the discourse vectors that govern the sentence. The average of the embeddings of the words in the sentence multiplied by a scalar is the MAP estimate of the single discourse vector. The SIF model has two types of “smoothing term”, to account that some words occur out of context and some are frequent words that appear often.

Computing the average of the word embeddings in a sentence to give more weight to rather irrelevant words, semantic. Smooth Inverse Frequency resolves this problem in two means:

1. Weighting:
   SIF computes the weighted average of the word embeddings in the sentence. Each of the word embedding is weighted by the formula a/(a + p(w)), where a is a parameter, and p(w) is estimated using equation (1) as the frequency of occurrence of the word in a corpus.
2. Common component removal:
   Then SIF will compute the principal component of the resulting embeddings for the set of sentences given. It will then subtract from these sentence embeddings the projections of their sentence embeddings on their first principal component. This will remove the variation associated with frequency and syntax that is less applicable semantically.

Pr[w emitted in sentence s |cs] = αp(w) + (1 - α)$\frac{\exp(<\tilde{C}_s, Vw>)}{Z\tilde{C}s}$(1)

where$\tilde{C_s} = \beta c_0 + (1 - \beta)C_s,\ c_0 \perp\ c_s$, where α and β are scalar hyperparameters.

B. DASH(Difficulty, Ability, Study history)

The DASH model encodes student ability, item difficulty, number of attempts, and number of correct answers for n items, where items represent the questions.

Recall likelihood is computed using,

$Z \sim Bernoulli(m(1 + r\ \cdot D)(1 + r \cdot D)^{-f}$(2)

where D is the time elapsed since the item was last reviewed by the student, r is a constant that controls decay rate,

m in equation (2) is given as,

$m =\ \sigma(a - d + h_\theta(t_{1:k}, z_{1:k-1}))$(3)

In equation (3) above, σ is the logistic function, t1:k is the times at which reviews 1 through k occurred, z1:k-1 are the outcomes on reviews 1 through k – 1.

h_θ in equation (3) is given as,

$h_\theta(t_{1:k}, z_{1:k-1}) =\ \sum_{w=1}^{W} \theta_{2w-1} \log(1 +\ c_w) +\ \theta_{2w} \log\ (1 + n_w)$(4)

Study history on recall probability is summarized by h_θ.

IV. IMPLEMENTATION AND RESULTS

The proof of concept of the proposed solution is implemented in python. The solution is implemented into two components:

1. The SIF model and,
2. The RLTutor agent.

The SIF model is trained with the dataset and then used to compute sentence vectors for the answers provided by the user. The RLTutor agent is a reinforcement learning agent that takes the next action in the environment from a set of available actions. That is, it selects the next question when it accepts the current state from the environment. The transition in the state and reward is computed and sent back to the agent.

The next question is predicted based on the outcome of the previous answer given by the user. If the user gives the right answer, the next question will be selected from a harder set of questions. If the answer given by the user is incorrect an easier question will be selected.

Figure 2 is a snapshot that shows the account login window of the GUI interface of QuizBot. Figure 3 shows a snapshot of the registration window. The user should register with a unique username and password that is not taken by any other user or else an error message will be displayed.

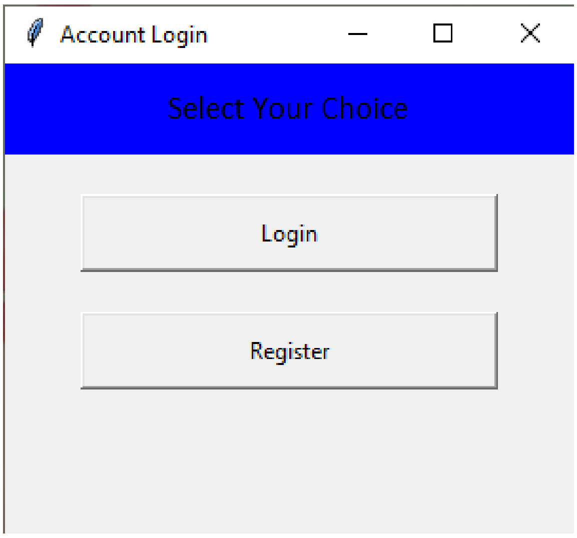


Figure 2: Account Login window

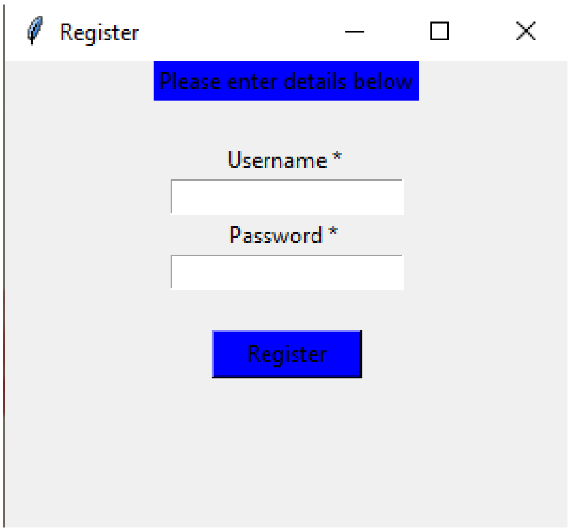
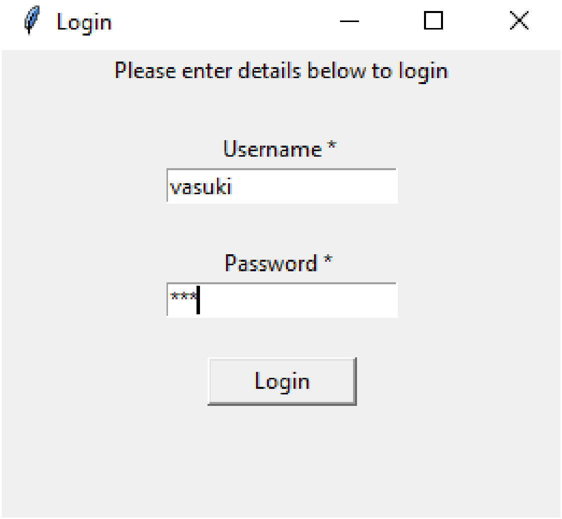


Figure 3: Registration window

Figures 4 and 5 show the snapshots of the login window and login successful message displayed respectively. The user will log in with the username and password selected by them. If a wrong username or password is provided an error message will be displayed.

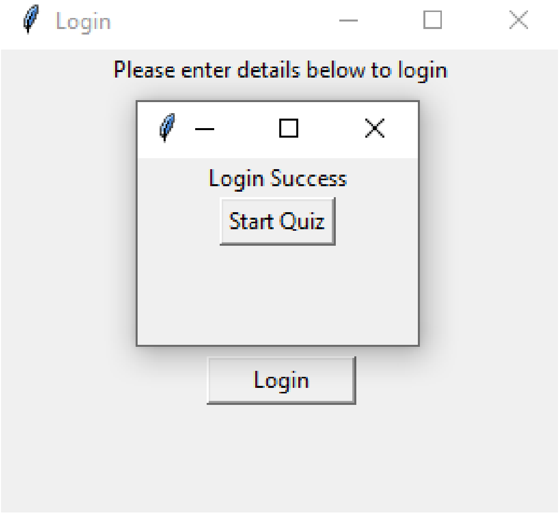


Please enter details below to login

Username * vasuki

Password * ***

Figure 4: Login window

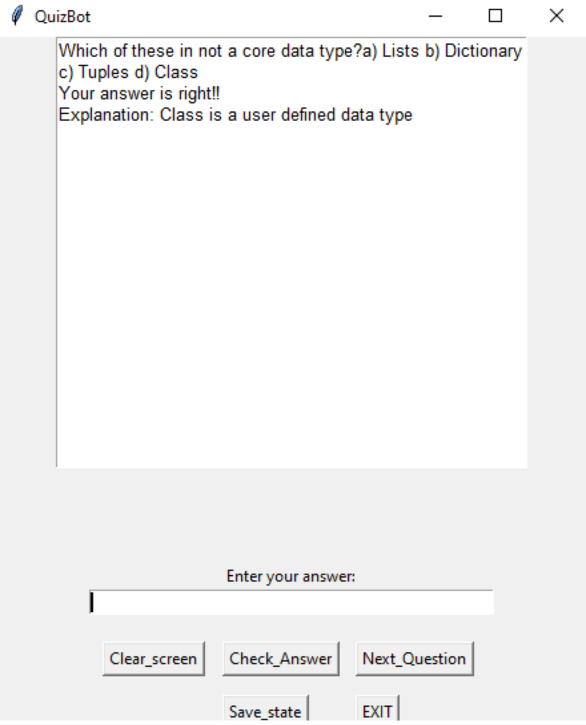


Please enter details below to login

Login Success

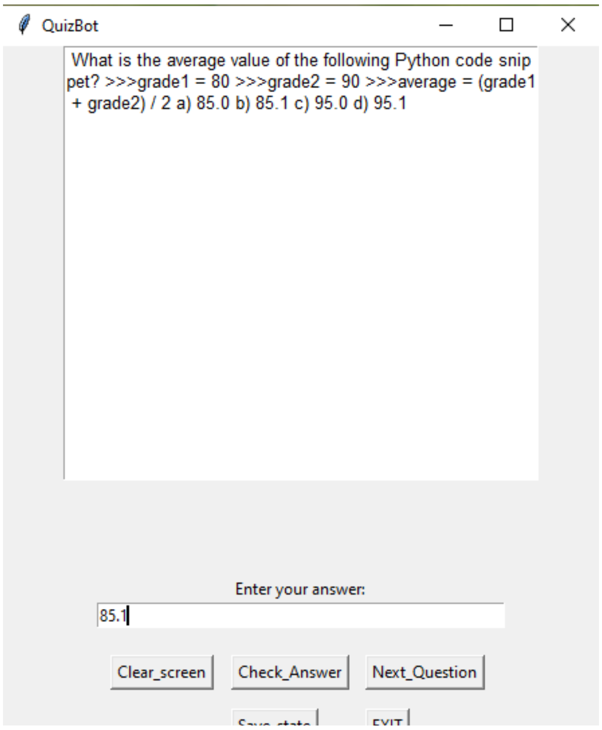
Figure 5: Login Successful

Figure 6 and 7 shows the quiz window, it contains Clear_Screen, Check_Answer, Next_Question, Save_state, and EXIT buttons. The text field can be cleared using the Clear_Screen button. After the user types in the answer, he should click on the Check_Answer button. Clicking on this button will fetch the answer, compute the sentence vector, and compare it with the actual answer to give the outcome of the answer given. Along with this, a brief explanation of the concept in the question is displayed for a better understanding of the user. In case the user gives a wrong answer, the explanation displayed will be useful to the user to learn the concept and recall later. When the Next_Question button is clicked, the next question is predicted for the user. The Save_state button will save the states of the environment to continue with the same states or progress when the user returns to take up the quiz. The EXIT button should be clicked after clicking the Save_state button. When the EXIT button is clicked user exits from the quiz window.



```
Which of these in not a core data type?a) Lists b) Dictionary
c) Tuples d) Class
Your answer is right!!
Explanation: Class is a user defined data type
```

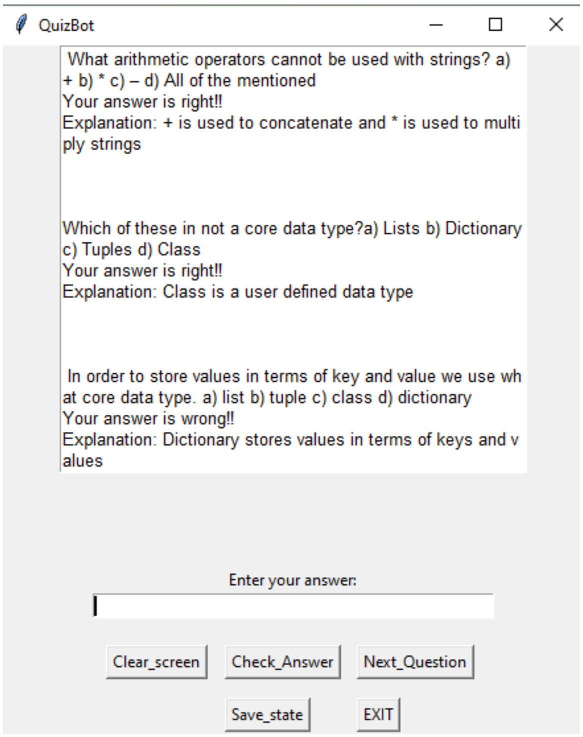
Figure 6: The answer given by the user is right.



```
 What is the average value of the following Python code snip
pet? >>>grade1 = 80 >>>grade2 = 90 >>>average = (grade1
 + grade2) / 2 a) 85.0 b) 85.1 c) 95.0 d) 95.1
```

Enter your answer: 85.1

Figure 7: The answer given by the user is wrong.

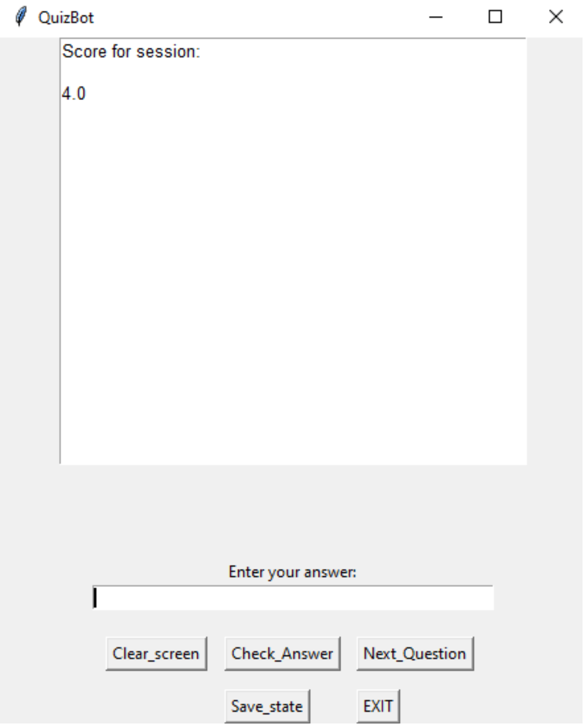


```
 What arithmetic operators cannot be used with strings? a)
+ b) * c) – d) All of the mentioned
Your answer is right!!
Explanation: + is used to concatenate and * is used to multi
ply strings


Which of these in not a core data type?a) Lists b) Dictionary
c) Tuples d) Class
Your answer is right!!
Explanation: Class is a user defined data type


 In order to store values in terms of key and value we use wh
at core data type. a) list b) tuple c) class d) dictionary
Your answer is wrong!!
Explanation: Dictionary stores values in terms of keys and v
alues
```

Figure 8: Sequence of questions



```
Score for session:

4.0
```

Figure 9: Score of the session being displayed

Figure 8 shows a sequence of questions asked to the user. Figure 9 shows the snapshot of the score being displayed at the end of the quiz session. The score is displayed after clicking on the Save_state button.

V. CONCLUSION AND FUTURE WORK

The paper focuses on implementing Quizbot with the use of reinforcement learning. The DASH environment is created using the gym environment, which is a toolkit used for developing and comparing reinforcement learning algorithms. The agent RLTutor achieves the goal of taking action in the DASH environment for predicting subsequent questions to be answered by the user. It uses a supervised smooth inverse frequency algorithm to compute the sentence vectors.

The quizbot can be enhanced with the SIF model to accept answers and compute sentence embeddings for the answers by considering only the main intent or concept in the answer by omitting stop words in the sentences. The DASH algorithm can be further improved to provide the user with information on areas of improvement by depicting their level of knowledge of the user in a particular concept of the subject.

REFERENCES

[1] A Pilot Study Integrating an AI-driven Chatbot in an Introductory Programming Course, Matthew Verleger, and James Pembridge, 2018, 2018 IEEE Frontiers in Education Conference (FIE).
[2] Building Pedagogical Conversational Agents, *Affectively* Correct, Michalis Feidakis, Panagiotis Kasnesis, Eva Giatraki, Christos Giannousis, CharalamposPatrikakis, and Panagiotis Monachelis, April 2019, CSEDU 2019 - 11th International Conference on Computer Supported Education.
[3] A Knowledge-based Methodology for Building a Conversational Chatbot as an Intelligent Tutor, Xavier Sanchez-Diaz, Gilberto Ayala-Bastidas, Pedro Fonseca-Ortiz, and Leonardo Garrido, September 2018, Mexican International Conference on Artificial Intelligence.
[4] Detecting Phishing Attacks using Natural Language Processing and Machine Learning, Tianruipeng, Ian G. Harris, Yuki Sawa, 2018, 2018 IEEE 12th International Conference on Semantic Computing (ICSC).
[5] Automatic Extraction of Major Osteoporotic Fractures from Radiology Reports using Natural Language Processing, Yanshan wang, Saeed Mehrabi, Sunghwan Sohn, Elizabeth J. Atkinson, Shreyasee Amin, Hongfang Liu, 2018.
[6] Implementation of Chatbot for ITSM Application using IBM Watson, Neha Atul Godse, ShaunakDeodhar, Shubhangi Raut, PranjaliJagdale, 2018, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA).
[7] NLAST: A natural language assistant for students, Fernando A. Mikic Fonte, Martín Llamas Nistal, Juan C. Burguillo Rial, and Manuel Caeiro Rodríguez, April 2019.
[8] Conversational Chatbot System for Student Support in Administrative Exam Information, H. Abu Rasheed, J. Zenkert, C. Weber, M. Fathi, November 2019, 12th annual International Conference of Education, Research and Innovation (iCERi2019).
[9] A Platform for Human-Chatbot Interaction Using Python, Bhaumik Kohli, Tanupriya Choudhury, Shilpi Sharma, Praveen Kumar, 2018, 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT).
[10] An End-to-End Goal-Oriented Dialog System with a Generative Natural Language Response Generation, Stefan Constantin, Jan Niehues, and Alex Waibel, March 2018.
[11] Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 1070–1075.
[12] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. International Conference on Learning Representations (2017).
[13] Robert V Lindsey and Michael C Mozer. 2016. Predicting and improving memory retention: Psychological theory matters in the big data era. In Big Data in Cognitive Science. Psychology Press, 43–73.

BIOGRAPHY

D M Vasukicompleted her B.E in Computer Science and Engineering from Visvesvaraya Technological University, Karnataka. She is currently pursuing her MTech at Dayananda Sagar College of Engineering, Bengaluru, Karnataka in Computer Science and Engineering. Her areas of interest include Natural Language Processing, Web Application Development, Machine Learning, and Artificial Intelligence.

Dr. Krishnan is a Professor at Dayananda Sagar College of Engineering, Bangalore, Karnataka. Bearing an experience of 35 years. His areas of interest include Computer Vision, Software Testing, Artificial Intelligence,and Machine Learning. He is a senior member of IEEE and also holds membership in ACM.

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

📱 9940 572 462 ✆ 6381 907 438 ✉ ijircce@gmail.com

www.ijircce.com

Scan to save the contact details