

Developing an Educational Chatbot for Scientific Data Management Course Using DialogFlow

Manal M. A. Abdelmoiz^{1,*}, Mohamed M. M. Mostafa² and Taysir H. A. Soliman¹

¹Information System Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt

²Department of Foundations of Education, Faculty of Education, Assiut University, Assiut, Egypt

Received: 3 May 2023, Revised: 2 Feb. 2024, Accepted: 7 Feb. 2024

Published online: 1 May 2024

Abstract: The COVID-19 pandemic has radically altered the worldwide learning environments, setting the stage for Electronic Learning (E-Learning) advancement, where remote learning is facilitated through digital tools. The key stakeholders (professors, teaching assistants, and students) face bottlenecks as they shift to online education providing blended learning. New learning tools based on Natural Language Processing (NLP) are provided. Designing a chatbot is one of the solutions to deal with this issue. Chatbots are simple computer programs that attempt to simulate human conversation using Artificial Intelligence (AI) and NLP. They allow learners to have a standardized learning environment. In this paper, we have set up a chatbot application named Scientific Data Management BOT (SDMBOT) for handling E-Learning activities specifically for Scientific Data Management (SDM) courses based on AI and NLP techniques using Dialogflow Framework, a Google development platform for building NLP-based human-computer interface solutions. SDMBOT was trained on a dataset that was specifically created based on course content. The web or mobile app, via which our built chatbot is available, is used for student interaction. Students can ask the chatbot questions concerning SDM, and the chatbot will process the message and respond to the user by displaying the proper result. The accuracy of the SDM chatbot, which is calculated by using a confusion matrix indicated that our chatbot is 74 % accurate.

Keywords: Artificial Intelligence, COVID-19, Data Management, Educational Chatbot, E-Learning, Machine Learning, NLP.

1 Introduction

Technology's rapid growth has created many new solutions and approaches to meet all students' objectives based on their educational goals and demands. Today's students may take notes on a tablet, capture images or add screenshots of the lecturer's presentation to his notes, send the professor an email at any time, and verify any required tasks at a learning management platform. The ability to communicate with the instructor outside of the classroom becomes less effective as the number of students rises, even though these technologies benefit the students by enhancing their understanding of the lecturer's response to a particular question. Also, professors may not always reply to a student right away due to their busy schedules. A difficulty occurs when a lecturer might not be accessible or unable to respond to specific queries when necessary. Furthermore, the COVID-19 pandemic has radically altered the worldwide learning environments, providing the foundations for

developing E-Learning, where teaching on digital platforms is carried out remotely. One of the latest technologies that meet this need is chatbots. Software programs, known as chatbots or "conversational agents", imitate spoken or written human language to simulate a conversation or other contact with a real person. NLP is the innovation at the heart of the chatbot's rise. NLP can recognize text and spoken words just like humans, translate text from one language to another, answer voice commands, and quickly encapsulate huge amounts of text [1,2].

The use of chatbots in E-Learning can be viewed as a significant innovation. They are proving to be the most creative way to combine technology with education. Chatbots have the advantage of being available 24/7 to resolve user queries. They also can handle multiple requests without compromising the quality of interactions. The purpose of this study is to develop a chatbot application named SDMBOT using AI and NLP, which takes a text question from a student, extracts intent

* Corresponding author e-mail: manalabdelmoiz@aun.edu.eg

and entity from the query, and processes them using NLP to reply with accurate answers. Considering our course, the chatbot functions differently from a standard search engine, including details from the lectures so that students can get responses that more accurately reflect what the professor thinks is a suitable response [3,4].

In brief, our contributions are listed as follows:

1. We proposed a smart chatbot called SDMBOT using the Dialogflow framework (Google development platform for building NLP-based human-computer interface solutions), one of the most popular and widely used frameworks. SDMBOT is designed to answer student questions by displaying the proper result.
2. A simple English language dataset has been created for the SDMBOT. These datasets have been written in the form of a FAQ (Frequently Asked Questions) to integrate with the framework used. The proposed questions are about 500 questions in English, which concern the subjects introduced in the SDM Course.
3. A user interface was created using the Angular framework to make it easy for students to interact with SDMBOT. The interface can be accessed via web or mobile.

The organization of the paper as follows: Section 2 contains details on related work on chatbots. Section 3 contains the proposed work. Results and experiments are discussed in section 4. The conclusion and future work are in section 5.

2 Related Work

Recently, the development of conversational models (chatbots) has caught the attention of AI scientists for several years in different fields like education, medicine, and banking. In the medical sector, chatbots are beneficial for giving patients free consultations, recommending other drugs..., etc. even without traveling to the hospital or seeing a doctor. In [5] Podrazhansky, Ava, et al. described the layout of a chatbot mobile application for human mental health that, using machine learning algorithms, can identify whether a person has a mental disease and offer treatment options. Their app, which uses NLP and integrated continuous emotion conversation analysis and sentence/audio production, can communicate with users through texting, chatting, and videoing.

Comendador et al. [6] introduced another chatbot called Pharmabot, a pediatric generic medication consultant chatbot designed to prescribe and provide relevant information on generic drugs for children. The system's front end is in Visual C#, and the back end is in MS Access. It is not a web-based system and is designed to work on a standalone PC.

Doherty, Dana, et al. in [7] presented another chatbot for banking that can answer questions and help with

personal banking for consumers. Users of the app may view transaction history, account information, and balance. Moreover, it incorporates SMS, currency conversions, two-factor authentication, and the capability to use personal assistants like Google Assistant to communicate with the service. Their chatbot's benefit is that it enables consumers to communicate with their bank using NLP, making it easier and more effective for users.

As for education systems, many educational chatbot applications have been developed to contribute effectively to finding better solutions for E-Learning. One of these research works is Sarosa, M., et al [8] developed an application to be an English learning media within Facebook so that students can learn English more efficiently. This application is a form of Chatbot (response robot) that replaces teachers to help people who have difficulty learning English.

DOLY is a machine communication system that was developed by Kowsher, Md, et al. in [9] for the Bengali education system to provide human-like responses to user queries. It is an artificial intelligence-based human-computer interaction system that employs machine learning techniques and Bengali Natural Language Processing (BNLP). It uses the collection of knowledge to respond to a user's answer to a query and find the desired output. A train function adapter uses the DOLY's knowledge of responses to train it.

For visually impaired people, Kumar, M. Naveen, et al [10] demonstrated an Android application that included an educational Chatbot. The chatbot can be used by asking questions in spoken natural language, which the app will convert to text and run against the Artificial Intelligence Markup Language (AIML) database, or if no pre-saved response is found, the Wikipedia API will be used. The response is then texted and voiced back to the chatbot.

Another study by Farhan M. et al [11] used a web bot in an E-Learning platform to handle the students' lack of real-time responses. If a student asks a question on an E-Learning platform, the teacher may respond later. This delay grows as the number of students and questions grows. A web bot is a chatbot that anticipates future occurrences using terms put on the Inbox. Pandora is used in this research as a bot that stores queries and replies to them in XML format AIML. This bot has been programmed with a series of questions and responses; when it is unable to respond to a query, a human user is tasked with doing so.

In [3] Zahour, Omar, et al. employed the DialogueFlow tool in their implementation. Dialogflow is a Google development platform for creating Natural Language Processing-based human-computer interaction solutions. Their chatbot proposes 301 questions in French regarding educational and professional orientation based on John Holland's concept and the RIASEC typology. These questions are divided into four categories: activities, skills, jobs, and interests. The character. The eventual purpose of their chatbot, according to the user's

responses, is to allow him to discover his dominant personality type to provide the proper employment to this user based on the Holland model.

Likewise, [12] Ranavare, Sushil S., and R. S. Kamath describe the design and implementation of an Artificial Intelligence (AI)-based Chatbot for managing professional college placement processes. To grasp the institute's service, they employed Dialogflow, an NLP module that translated students' requests during the discussion into structured data.

In [2] Bala Dhandayuthapani, V. created a chatbot to help staff members who are preoccupied with non-academic liabilities, such as supporting students. Their academic work suffers as a result of the time they spend on these tasks. This chatbot helps by providing information and training related to stress management, language development, taking medication for exams, and accessing information. The chat text can be in either English or their native or national language.

Shaha, Sarvesh, et al. [13] proposed a chatbot system that assists students in gaining access to college information and predicting admission. The technique benefits not just the student, but also the college, by reducing the need for extra staff. Students will have easier access to information due to the system. The system's input processing efficiently handles college-related keywords. The system is divided into two parts. The first is the chatbot user interface, which takes the form of an Android app and allows users to communicate with the system. The backend functionalities are the system's second half, consisting of a server with several modules, such as text recognition, context comprehension, and question classification.

And [14], Priadko, et al. outlined the process of creating a chatbot that used the Telegram mobile messenger to provide students with scheduling information. The following tasks have been implemented programmatically: to obtain data from multiple pages of the spreadsheet (college/institute, red/green week, group number, day of the week, issue number, subject name, teacher information); to fit messenger (The data is presented in the form of XML), a convenient data presentation mechanism is implemented in the messenger (chatbot). Using Python's Telegram API, the software was designed to increase the number of students to obtain schedule information. Get instant schedule information, minimize travel time, and optimize student activity planning and higher education functioning.

In [15] Patel, N. P., et al. created a chatbot that is being used to answer college or university-related questions at any time and from anywhere. The chatbot can be included on any college or university's website. They introduced the topic by describing an issue that students experience during the college admissions process when they ask a variety of questions about costs, schedules, and term starts, among other things. They further claimed that such a procedure is time-consuming and exhausting. To address these concerns, they developed a chatbot that can

be added to any university website and used at any time. HTML, CSS, and jQuery were used to create the model's front end. Ajax was used to make a request to a PHP file and receive a response, while jQuery was used to show the messages to the user. The answer to the query given by the user was stored in an SQL database by the authors.

Also, Ranoliya, B. R., et al. [16] discussed a chatbot design used to answer FAQs efficiently and accurately about university-related inquiries. The implemented chatbot is built on AIML, and the researchers hope to expand its capabilities by incorporating Latent Semantic Analysis in the future. The bot performed admirably in terms of providing students with assistance 24 hours a day, seven days a week.

Finally, Mondal, A., et al. [17] concentrated on developing a textual communication application, known as a chatbot, for the educational sector. They suggested chatbot supports users in addressing their questions. They used a random forest ensemble learning method in the presence of extracted features from their provided dataset to construct the system.

3 Proposed Work

SDMBOT was developed using the Dialogflow framework (Google development platform for building NLP-based human-computer interface solutions), one of the most popular and widely used frameworks because it has sophisticated NLP capabilities that make it possible for chatbots to understand and respond to natural language input, scalability, multiple integrations with a variety of third-party tools such as Facebook Messenger, Slack, and Google Assistant to make it easy to reach to our audience. A simple English language dataset, which concerns the subjects introduced in the SDM Course, has been created to train the SDMBOT. These datasets have been written in the form of a FAQ to integrate with the framework used. Also, a user interface was created using the Angular framework to make it easy for students to interact with SDMBOT. The interface can be accessed via web or mobile. This section outlines the four key phases and their subprocesses of creating SDMBOT, as seen in Fig. 1. The four phases of SDMBOT are as follows: Dataset preparation, GUI development, Agent development, and Updating dataset.

3.1 Dataset Preparation (Corpus) Phase

The most crucial component of any system is Dataset preparation. A simple English language dataset has been created for the SDMBOT. These datasets have been written in the form of a FAQ to fit the framework used. Initially, the proposed questions are about 500 questions in English, which concern the subjects introduced in the SDM Course. This course is concerned with managing

scientific data, including the scientific data management life cycle. This life cycle includes phases: data planning, collection, description, assurance, preservation, integration, and analysis. To ensure proper learning and application of this course, students are taught practical components which include: The OpenRefine tool, R language, and Ontology. This course is taught to senior-year Information Systems students.

These various question types are all used in English. The corpus containing these questions has been created using Python language. The output of Python code is a CSV file consisting of 2 columns (question, and answer).

These questions are distributed according to four modules namely: general questions related to the lectures, R language, Open Refine tool, and Ontology, as follows:

1. Lecture module

The lecture module is concerned with managing scientific data, including the scientific data management life cycle. This life cycle includes phases: data planning, collection, description, assurance, preservation, integration, and analysis. Also, it includes explanations of the previously mentioned practical part. Examples are illustrated in Table 1.

Table 1: Examples of Question's Types for Lecture Module.

| Question Type | Example1 | Example2 | Example3 |
|-----------------------------|--|--|--|
| General or No/Yes Questions | A histogram is suitable for large datasets for the number of observations? | Is ontology a set of classes and relations only? | Is the relation in an ontology only have an "is-a" relation? |
| Special or WH-Questions | What Purposes Does Data Entry Serve? | Why do we use ontologies? | What are Data quality dimensions? |
| Essay Questions | List the four types of data structure growth. | Explain the publish-and-forget model | Mention what are the Kinds of ontologies |

2. R language module

R is a language and environment for visualization and statistical computation. It is a GNU project that John Chambers and colleagues created at Bell Laboratories (previously AT&T, now Lucent Technologies) [18]. For data analysis and creating statistical software, statisticians, bioinformaticians, and data miners all utilize R [19]. The questions created for this topic include the following topics: R Data types, Operators, Data Structures, Functions, Conditions, Loops, Reading and Writing files, Statistics, and Visualization. Examples are illustrated in Table 2.

Table 2: Examples of Question's Types for R Module.

| Question Type | Example1 | Example2 | Example3 |
|-----------------------------|---|---|---|
| General or No/Yes Questions | Do you reject the null hypothesis at the 0.05 significance level? | Does the R boxplot show the mean or median? | Can we update and delete any of the elements in a list? |
| Special or WH-Questions | Why would one use a boxplot? | Exactly how do I view charts in R? | How many groups can you compare with ANOVA? |
| Essay Questions | Explain the differences between lapply and sapply. | Describe the R process for making a list. | Explain How to use RStudio. |

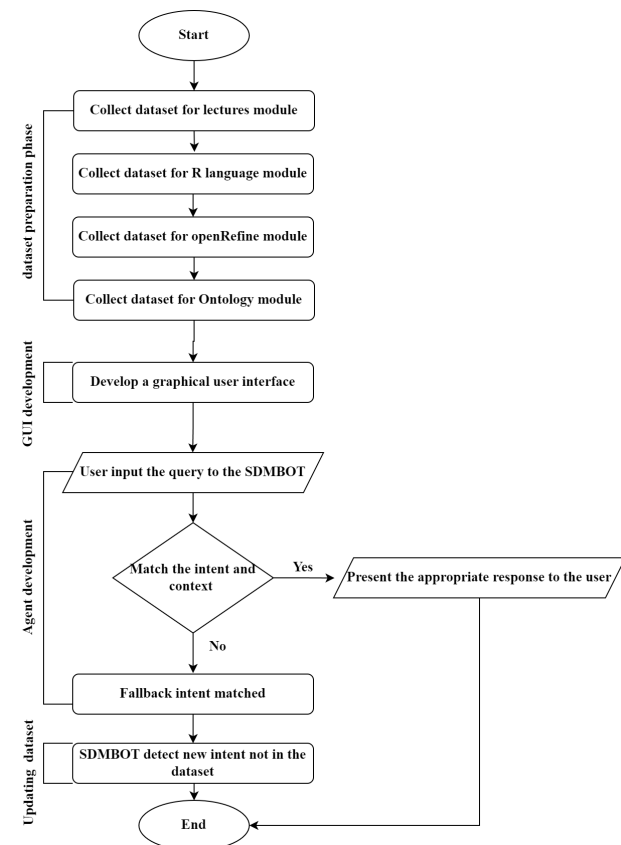


Fig. 1: Flowchart of the Main Phases and Sub-Processes of SDMBOT.

There are three types of questions:

1. General or Yes/No Questions

Common questions that can be answered with a simple "yes" or "no" are logically called yes/no".

2. Special or Wh-Questions

A special question asks a specific question by using a specific word at the beginning of the sentence. The words who, what, where, when, why, how, how many, and so on are used to begin the question.

3. Essay Questions

A question that requires an answer in the form of a sentence, paragraph, or short composition.

3. OpenRefine module

OpenRefine is a data manipulation tool that cleans, reshapes, and intelligently edits large amounts of dirty, unstructured data in batches. It is an open-source tool with code that may be reused in other projects. The questions created for this topic include the following topics: faceting, grouping, editing cells, reconciling, and extending web services. It aids in data analysis by filtering and faceting and turns the data into a more organized format [20]. Examples are illustrated in Table 3.

Table 3: Examples of Question’s Types for OpenRefine Module.

| Question Type | Example1 | Example2 | Example3 |
|-----------------------------|---|---|--|
| General or No/Yes Questions | Is it possible to split columns with open refine? | Is my data safe in OpenRefine? | Is OpenRefine only for the Windows Operating System? |
| Special or WH-Questions | What is the open column merging process? | How can I get started with OpenRefine? | In OpenRefine, how does clustering work? |
| Essay Questions | List types of Facets in open refine. | Describe the process of adding columns by fetching URLs | Mention the format in which you can export data from an OpenRefine project |

4. Ontology module

An ontology is a formal description of knowledge as a collection of concepts within a domain and their relationships. It promotes a shared understanding of information and makes domain assumptions clear, helping organizations to better comprehend their data [21]. The Protégé editor and RightField tool are taught to students to help them apply the notion of ontologies. Protege is a free, open-source ontology editor and intelligent system framework [22]. RightField is a free and open-source application for incorporating ontology term selection into Excel spreadsheets. A Template Creator uses RightField to construct semantically aware Excel spreadsheet templates [23]. The questions created for this topic include the following topics: Ontology basics, Protégé editor, SPARQL query, and the RightField tool. Examples are illustrated in Table 4.

Table 4: Examples of Question’s Types for Ontology Module.

| Question Type | Example1 | Example2 | Example3 |
|-----------------------------|--|-----------------------------------|---|
| General or No/Yes Questions | Is it possible for me to get RDF responses to my SPARQL queries? | Is SPARQL a programming language? | Can I filter my queries with SPARQL? |
| Special or WH-Questions | How do I get RightField? | How do you install a protege? | How do you visualize ontology in protege? |
| Essay Questions | list the building blocks of the semantic web | Describe the RDF type. | Describe OWL W3C. |

3.2 GUI Development Phase

GUI is an important component of any system. The Front-end is developed using the Angular framework. Angular is a platform and framework for building single-page client applications using HTML and TypeScript, as shown in Fig. 2.

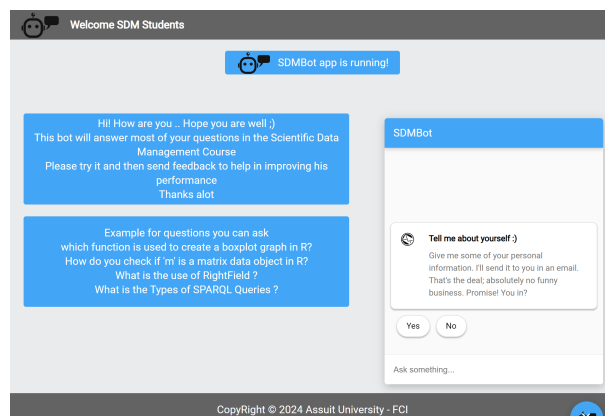


Fig. 2: Design of SDMBOT.

Angular (commonly referred to as "Angular 2+" or "Angular CLI") is a TypeScript-based free and open-source web application framework led by the Angular Team at Google and by a community of individuals and corporations. It was considered that the design of SDMBOT is suitable for laptops, tablets, or phones (responsive design), as shown in Fig. 3 and Fig. 4, respectively.

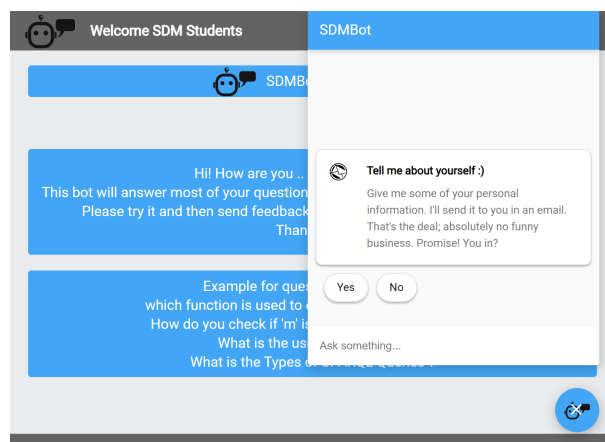


Fig. 3: Tablet View.

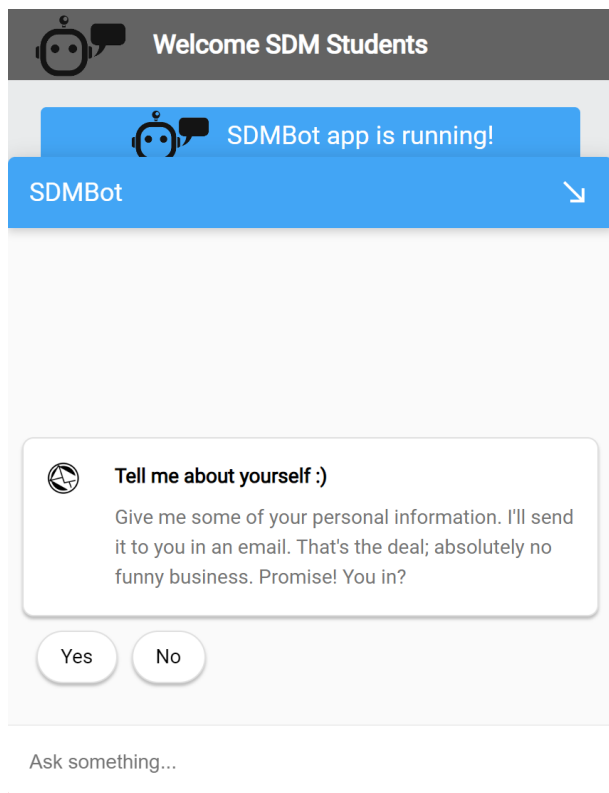


Fig. 4: Mobile View.

3.3 Agent Development Phase

To develop the SDMBOT, the Dialogflow framework has been used. Dialogflow (previously Api.ai, Speaktioit) is a Google development platform for building NLP-based human-computer interface solutions. Table 5 compares some common frameworks based on some features they provide to the users [24–28].

Table 5: Comparison of Common Chatbot Frameworks.

| Name | Online Integration | Languages Supported | Free/Paid | Company | Open-Source | Machine learning |
|------------|--------------------|---------------------|-----------|-----------|-------------|------------------|
| DialogFlow | Yes | Multilingual | Free | Google | No | Yes |
| IBM Watson | No | Multilingual | Free | IBM | No | Yes |
| RASA NLU | No | English, German | Free | RASA | Yes | No |
| Wit.ai | No | Multilingual | Free | Facebook | No | Yes |
| LUIS | No | Multilingual | Free | Microsoft | No | Yes |

Having compared various frameworks, assessed them based on specific feature criteria, and concluded that Google’s Dialogflow framework performs the best. The benefit of Dialogflow is that it can be used to create an agent, which can then be integrated with speech or text

across many services like Amazon Alexa, Google Assistant, Facebook Messenger, Twitter, etc.

Dialogflow is an NLU (Natural Language Understanding) engine that allows users to easily create intelligent chatbots, where NLU is a subfield of NLP. NLU comprises two fundamental concepts, which are the recognition of intent and entities. Generally, an algorithm that creates a model that understands meaning is classified as NLU rather than NLP.

To develop SDMBOT, Dialogflow creates agents. In the best sense, agents can be thought of as NLU modules. You can incorporate these into your software, item, or service. The agent manages the conversation flow and transforms what the user says into a specific purpose. Our system architecture is shown in Fig. 5.

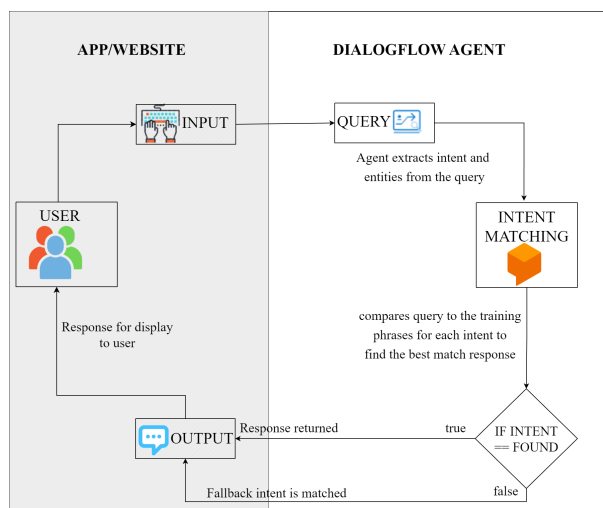


Fig. 5: SDMBOT Architecture.

Creating our SDMBOT involves 6 key steps: user input, dialog flow engine, intent matching, training phrases, responses, and website/ app integration. Here’s how it can happen in detail for the user query ”How do I get started with OpenRefine?”.

1. User Input

The user types a question or statement into the SDMBOT app Like “How do I get started with OpenRefine?”.

2. Dialog Flow Engine

The user query is sent to the dialogflow engine to process it and return the proper result.

3. Intent Matching

The chatbot analyzes the user input and identifies the intent behind it based on previously programmed intents. An intent is a relationship between what a user says and the action that our agent must perform. In most cases, a single agent will have multiple

intentions. Two algorithms are used to match intents: rule-based grammar matching and Machine Learning (ML) matching, simultaneously the two algorithms are attempted both and choose the best result [24]. Rule-based matching algorithms are useful for simple, predictable use cases, while ML algorithms are more appropriate for complex, variable use cases. Both algorithms can be used together to balance accuracy and efficiency, making it possible to handle a wide range of user inputs with varying degrees of complexity.

There is also the idea of default intents: welcome intent and fallback intent. When an end-user starts discussing with our agent, the default welcome intent is matched. Fallback intent, which is useful if the agent is confused. There are numerous static text replies that are preconfigured for this intent automatically, including "That eluded me. Could you repeat that?". Fallback intentions can be modified by altering the default text responses.

There are some steps involved in the intent-matching process including:

- (a)Preprocessing: The user input undergoes preprocessing to remove unwanted characters, stop words, stem, and lemmatize the text. So, "How do I get started with OpenRefine?" becomes "how do i get started with openrefine".
- (b)Tokenization: The input is broken down into individual tokens, which are individual words or parts of words. In this case, the tokens would be: "how", "do", "i", "get", "started", "with", "openrefine".
- (c)Spell checking: The query is checked for spelling errors to avoid misinterpretation of the query. For example, if the user writes the query "How do I get *stated* with OpenRefine?" with an error in the word "stated". Possible corrections for each word are generated based on word similarity metrics and dictionary lookups. In this case, possible corrections for "stated" may include "started", "stated" and "stated with". The best correction is selected based on the context and the likelihood of the correction being the intended word. The word "stated" would be corrected to "started". The resulting sentence becomes "how can i get started with openrefine".
- (d)Entity recognition: A particular content for intent is provided by an entity that reflects a word or objects relevant to intent. It must include a list of the potential values and synonyms that users might use to enter each entity. For dates, money, and many other things, Dialogflow offers built-in entities. New entities like Ontologies, OpenRefineImportingData, RDataTypes, ..., etc. have been defined in our agent.
- (e)Classification: The chatbot applies machine learning algorithms to classify the user's input into

one of the predefined intents. Intent classification using a rule-based matching algorithm involves defining a set of rules or patterns and then matching them against the user query. Rules are defined to capture different intents related to user queries. For the user query "how can I get started with OpenRefine" using a rule-based approach:

- i.Rule Definition: Rules are defined to capture different intents related to OpenRefine, such as "Get Started guide" or "Introduction to OpenRefine". - Example Rule 1: Pattern: "how do i get started with openrefine", Intent: "Get started guide."
 - Example Rule 2: Pattern: "What is openrefine", Intent: "Introduction to OpenRefine."
- ii.Matching: Each rule is applied to the preprocessed query to check if the pattern in the rule matches any part of the query.
 - Rule 1:
 - Pattern: "how do i get started with openrefine."
 - Matching: The rule matches the query as it contains the subsequence "how do i get started with openrefine".
 - Rule 2:
 - Pattern: "What is openrefine."
 - Matching: The rule does not match the query as it does not contain the exact subsequence "what is openrefine".
- iii.Intent Resolution: If a rule matches the query, the corresponding intent is assigned to the query. In this case, since Rule 1 matches the query, the intent would be "Get started guide".
- (f)Confidence score: The chatbot also calculates a confidence score for each intent, indicating the likelihood that the user's input matches that intent. evaluates probable matches using a confidence score for intent detection. These values range from 0.0 (uncertain) to 1.0. (Completely certain). Once intents have been scored, there are two possible outcomes:
 - i.The highest scoring intent is returned as a match if its confidence score is greater than or equal to the ML Classification Threshold setting [17].
 - ii.A fallback intent is matched if none of the other intents meet the Threshold.
- (g)Response selection: Based on the predicted intent and confidence score, the chatbot selects a response that is appropriate for the user's query and the context of the conversation.
- (h)Contextual analysis: The agent also considers the context of the conversation, such as previous user inputs and the current conversation, to further refine intent matching.

4.Training Phrases

These are examples of end-user expressions, which are words or phrases that users could write or say. If these phrases are described in various ways, then

Chatbot can respond effectively [5]. The agent matches the intent when an end-user expression resembles one of these phrases. For each intent, many training phrases have been created. Additionally, paraphrasing has been done using Python code to make each training word more recognizable. Each intent had at least 3 training phrases. For example

- (a)What are the main components of an Ontology?
- (b)Show me the main components of an Ontology.
- (c)The main component of an Ontology.

The more training phrases an agent has for each intent, the more accurately it can recognize and interpret user input.

5. Responses

The chatbot delivers the response to the app. For each intent, response data was provided when developing our agent. There are two common response types: text and image responses.

6. Website/App Integration

The best thing about Dialogflow is that you can integrate it into the Assistant app as well as integrate more than twenty platforms, including Twitter, Facebook, Slack, Viber, Kik, Twitter, and Angular. To add a web demo of SDMBOT to our website, integration features introduced by Dialogflow have been used by clicking the Integrations option in the left-hand panel and then clicking the Dialogflow Messenger tile button. Our app receives a response from the previous step and views it for the user.

3.4 Updating Dataset Phase

The updating dataset phase is an essential step in SDMBOT development. It involves reviewing and refining the chatbot's intent, entities, and response messages based on students' queries to improve its conversational flow and accuracy. In essence, this means determining which students' queries are not defined in our knowledge base and adding them to it. Also, it involves maintaining and updating the SDMBOT dataset continuously to ensure the chatbot remains efficient, effective, and relevant to user needs.

4 Experiments and Results

SDM Chatbot is implemented to meet the student's academic needs in the SDM course. To develop this chatbot, the Dialogflow framework has been used for building NLP-based human-computer interface solutions, a Python programming language is used to develop our dataset, and for the GUI, Angular framework has been used. This will help the student to fetch information for any topic they ask like Lecture topics, R language, Open Refine tool, and ontologies. SDMBOT has been integrated into a website page which is

<http://195.246.49.66:8080/>. A snapshot of the proposed chatbot is shown in Fig. 6,7,8,9.

Based on the interaction of students with the chatbot in the four modules related to the SDM course, the results were counted as shown in Fig. 10,11,12,13.

As shown in Fig. 14, 440 conversations from students have been collected, with 3777 questions.

A confusion matrix is a tool used to measure the accuracy of a chatbot. It is based on a prediction model used to evaluate a classifier's performance [29]. The confusion matrix consists of four elements: true positives, false positives, true negatives, and false negatives. True positives are those that the chatbot successfully identified as the correct answer. False positives are those incorrectly identified by the chatbot as the correct answer. True negatives are those that the chatbot accurately identified as incorrect, while false negatives are those incorrectly marked as incorrect, results shown in Table 6.

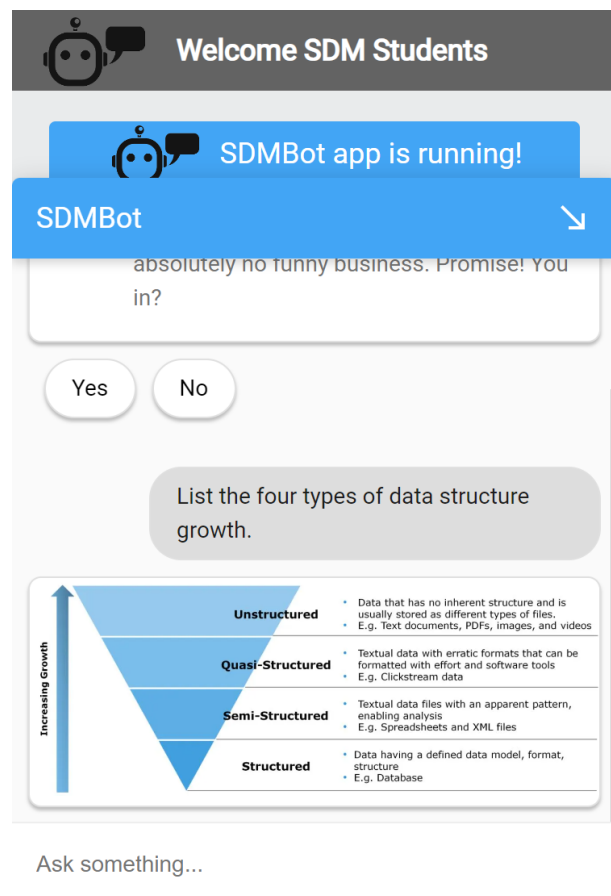


Fig. 6: Chatbot Response Snapshot for Questions about the Lecture module.

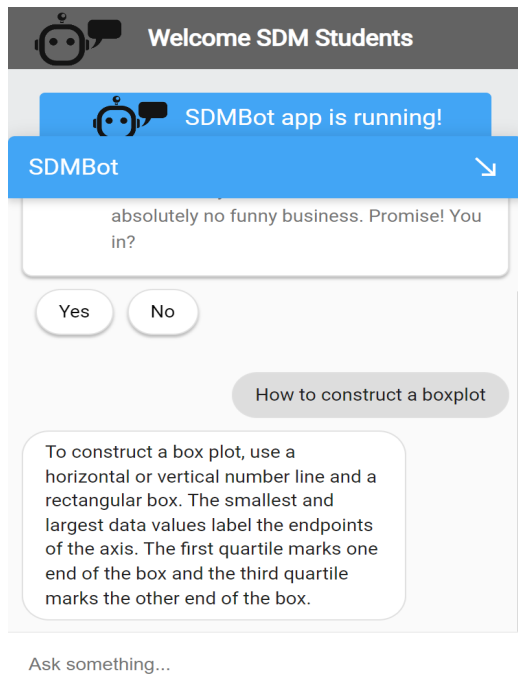


Fig. 7: Chatbot Response Snapshot for Questions about the R module.

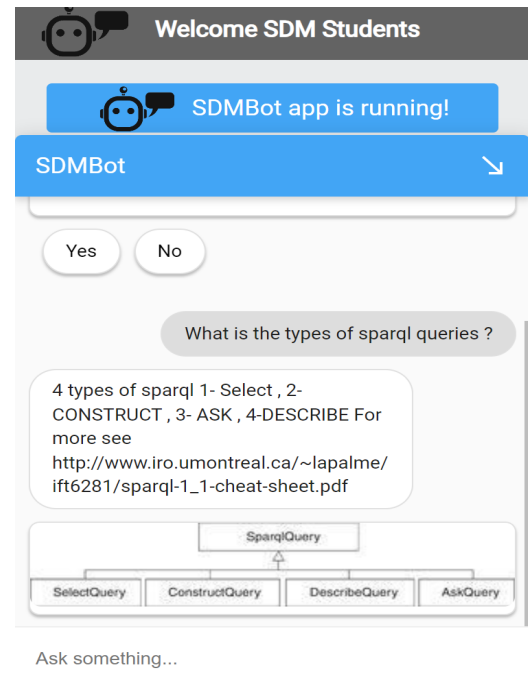


Fig. 9: Chatbot Response Snapshot for Questions about the Ontology module.

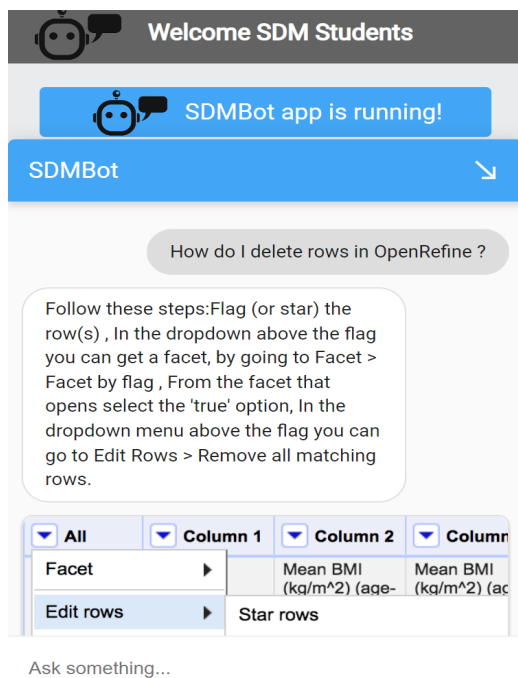


Fig. 8: Chatbot Response Snapshot for Questions about the OpenRefine module.

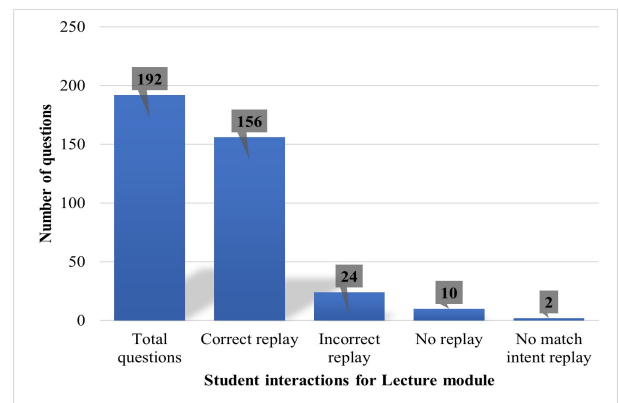


Fig. 10: The Number of Questions According to Student Interactions for the Lecture module.

The accuracy of the SDM chatbot is calculated by taking the sum of true positives and true negatives and dividing it by the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} = \frac{2622+174}{2622+174+917+64} = 0.74027$$

the result of the above equation indicated that our chatbot is 74 % accurate.

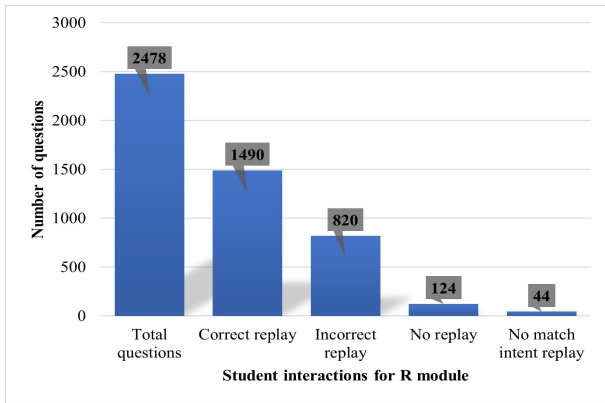


Fig. 11: The Number of Questions According to Student Interactions for R module.

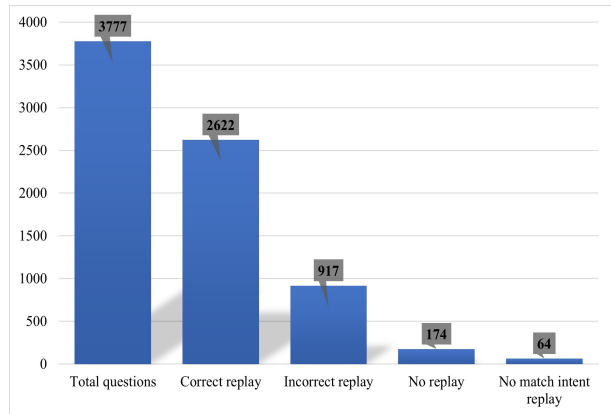


Fig. 14: Total Number of Questions According to Student Interactions for All Modules.

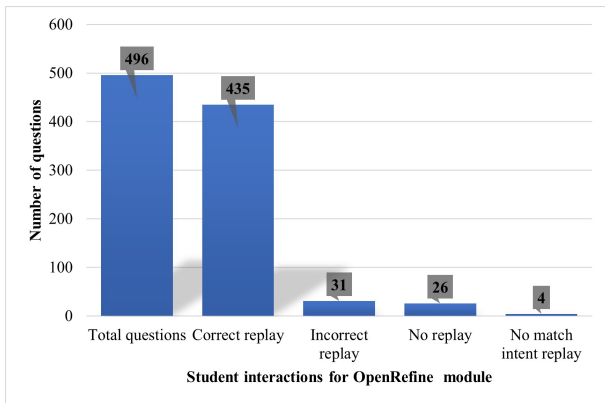


Fig. 12: The Number of Questions According to Student Interactions for the OpenRefine module.

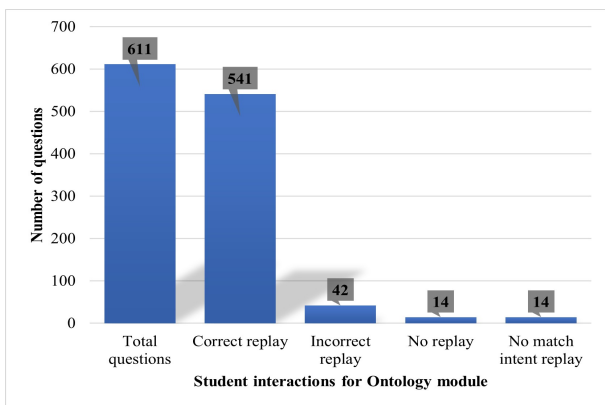


Fig. 13: The Number of Questions According to Student Interactions for the Ontology module.

Table 6: Confusion Matrix Result.

| Actual Predicted | Positive | Negative |
|------------------|---------------------|--------------------|
| True | TP = 2622 questions | TN = 174 questions |
| False | FP = 917 questions | FN = 64 questions |

A confusion matrix is an effective tool for measuring the accuracy of a chatbot. It is based on a prediction model and consists of four elements, which can be used to measure the accuracy of the chatbot. The accuracy can be further improved by identifying the most common errors made by the chatbot and making adjustments accordingly.

5 Conclusion and Future Work

Chatbots have become a vital asset for many businesses, especially in the educational sector. They have helped reduce the time and effort for teachers and students to access various resources and information quickly and accurately. By providing a personalized experience to users, chatbots are revolutionizing the educational industry and providing an increased level of convenience to all involved. They are also helping to create a more efficient learning environment by providing real-time feedback and assistance. Chatbots powered by Dialogflow have proven effective and efficient tools for businesses and organizations of various sizes and industries. Dialogflow’s NLP abilities allow chatbots to understand and interpret users’ requests accurately, providing a personalized and user-friendly experience. Integrations with other platforms and tools enable chatbots to become

more versatile and perform complex tasks. Overall, chatbots using Dialogflow offer numerous benefits and opportunities for businesses to improve their customer service, increase productivity, and reduce costs. In this paper, details about the design, framework, dataset used, and implementation of the SDMBOT is presented. The user will type his or her queries and then the system will extract the proper keyword from the given query and will produce the response.

In the future scope, we will work to improve the accuracy by increasing the size of the dataset, and will also try to add additional features to our chatbot, which will help students in different courses not only SDM courses.

Conflict of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

References

- [1] R. Reyes, D. Garza, L. Garrido, V. De la Cueva and J. Ramirez, Methodology for the implementation of virtual assistants for education using google dialogflow, in *Advances in Soft Computing: 18th Mexican International Conference on Artificial Intelligence, MICAI 2019, Xalapa, Mexico, October 27–November 2, 2019, Proceedings 18*, Springer2019, pp. 440–451.
- [2] V. B. Dhandayuthapani, A proposed cognitive framework model for a student support chatbot in a higher education institution, *International Journal of Advanced Networking and Applications* 14(2) (2022) 5390–5395.
- [3] O. Zahour, A. Eddaoui, H. Ouchra, O. Hourrane *et al.*, A system for educational and vocational guidance in morocco: Chatbot e-orientation, *Procedia Computer Science* 175 (2020) 554–559.
- [4] E. Levas, D. de Leon and Y. Li, A smart class chatbot for improving student learning and engagement, in *1st Computer Science Conference for CSU Undergraduates*, 2021.
- [5] A. Podrazhansky, H. Zhang, M. Han and S. He, A chatbot-based mobile application to predict and early-prevent human mental illness, in *Proceedings of the 2020 ACM Southeast Conference*, 2020, pp. 311–312.
- [6] B. E. V. Comendador, B. M. B. Francisco, J. S. Medenilla and S. Mae, Pharmabot: a pediatric generic medicine consultant chatbot, *Journal of Automation and Control Engineering* 3(2) (2015).
- [7] D. Doherty and K. Curran, Chatbots for online banking services, in *Web Intelligence*, 17(4), IOS Press2019, pp. 327–342.
- [8] M. Sarosa, M. Kusumawardani, A. Suyono and M. Wijaya, Developing a social media-based chatbot for english learning, in *IOP Conference Series: Materials Science and Engineering*, 732(1), IOP Publishing2020, p. 012074.
- [9] M. Kowsher, F. S. Tithi, M. A. Alam, M. N. Huda, M. M. Moheuddin and M. G. Rosul, Doly: Bengali chatbot for bengali education, in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, IEEE2019, pp. 1–6.
- [10] M. N. Kumar, P. L. Chandar, A. V. Prasad and K. Sumangali, Android based educational chatbot for visually impaired people, in *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*, IEEE2016, pp. 1–4.
- [11] M. Farhan, I. Munwar, M. Aslam, A. M. Enriquez, A. Farooq, S. Tanveer and A. P. Mejia, Automated reply to students' queries in e-learning environment using web-bot, in *2012 11th Mexican International Conference on Artificial Intelligence*, IEEE2012, pp. 63–65.
- [12] S. S. Ranavare and R. Kamath, Artificial intelligence based chatbot for placement activity at college using dialogflow, *Our Heritage* 68(30) (2020) 4806–4814.
- [13] S. Shaha, R. Pokalwar, S. Agrawal, S. Udupikar and B. Dhurape, Information chatbot for an educational institute, *VIT College* 6(11) (2020) 2395–0072.
- [14] A. O. Priadko, K. P. Osadcha, V. S. Kruhlyk and V. A. Rakovych, Development of a chatbot for informing students of the schedule (2020).
- [15] N. P. Patel, D. R. Parikh, D. A. Patel and R. R. Patel, Ai and web-based human-like interactive university chatbot (unibot), in *2019 3rd international conference on electronics, communication and aerospace technology (ICECA)*, IEEE2019, pp. 148–150.
- [16] B. R. Ranoliya, N. Raghuvanshi and S. Singh, Chatbot for university related faqs, in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE2017, pp. 1525–1530.
- [17] A. Mondal, M. Dey, D. Das, S. Nagpal and K. Garda, Chatbot: An automated conversation system for the educational domain, in *2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (ISAInLP)*, IEEE2018, pp. 1–5.
- [18] R: What is r?" r: What is r? www.r-project.org/about.html, (2023), Last accessed 24 Apr. 2023.
- [19] F. M. Giorgi, C. Ceraolo and D. Mercatelli, The r language: an engine for bioinformatics and data science, *Life* 12(5) (2022) p. 648.
- [20] Openrefine. www.openrefine.org, (2023), Last accessed 24 Apr. 2023.
- [21] What-are-ontologies www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies, (2023), Last accessed 24 Apr. 2023.
- [22] Prot eacute,g eacute www.protege.stanford.edu, (2023), Last accessed 24 Apr. 2023.
- [23] Rightfield www.rightfield.org.uk/about.html, (2023), Last accessed 24 Apr. 2023.
- [24] Dialogflow documentation www.cloud.google.com/dialogflow/docs, (2023), Last accessed 24 Apr. 2023.
- [25] Ibm watson assistant - virtual agent. www.ibm.com/products/watson-assistant, (2023), Last accessed 24 Apr. 2023.
- [26] Introduction to rasa open source and rasa pro. www.rasa.com/docs/rasa, (2023), Last accessed 24 Apr. 2023.
- [27] Wit.ai. www.wit.ai/docs, (2023), Last accessed 24 Apr. 2023.
- [28] Luis (language understanding) - cognitive services - microsoft. www.luis.ai, (2023), Last accessed 24 Apr. 2023.

- [29] S. Das and E. Kumar, Determining accuracy of chatbot by applying algorithm design and defined process, in *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, IEEE2018, pp. 1–6.
-