A Smart Class Chatbot for Improving Student Learning and Engagement

2021-03-06

Eric Levas, Dionisio de Leon, Yanyan Li
California State University San Marcos

This document is made available through ScholarWorks, the shared institutional repository of the California State University System. Visit https://scholarworks.calstate.edu/ for more openly available scholarship from the CSU.

Repository Citation
A Smart Class Chatbot for Improving Student Learning and Engagement

Eric Levas, Dionisio de Leon, Yanyan Li

California State University San Marcos, San Marcos, CA

Abstract

Getting questions answered quickly is one of the most desired features during a student’s learning process. It may not, however, always be possible given that professors may not always be available. In the past, students would need to google the answer or sift through pages of information to find their answers. This is not cost effective and does not guarantee the answers they obtained are correct. To solve the problem, we developed a smart chatbot that is able to understand students’ questions and provide an instant answer based on the learning of an instructor-provided question and answer spreadsheet. The learning is done with the help of Google Dialogflow, a natural language understanding platform that can extract various keyword to determine the most appropriate response. The student interaction is achieved via the Slack app, through which our developed chatbot is available on all major platforms, including mobile, web and desktop.

1. Introduction

A professor has a busy schedule and teaches many students, so they might not always respond to a student instantly. Some questions that students may have are simple, which is why many professors provide additional resources for their class to communicate outside of the lecture and office hours, such as a Slack channel. A problem arises when other students might not be available or able to answer certain questions when required.

If an assignment or lab deadline is quickly approaching, or if it is late at night when no one is online to see a question posted in the Slack channel, it would be beneficial for students to have access to a resource that won’t require them to search through dozens of articles on Google.

Our plan was to create a chatbot for students that can answer questions about specific topics on homework assignments as well as syllabus-related requests. With a chatbot, students will be able to type questions that will lead them to an immediate, concrete response. Chatbots are useful for educational purposes because they enhance student engagement and create a more efficient learning environment [1]. Taking our own course material into consideration, we want the chatbot to operate in a different manner than a regular search engine; we want the chatbot to provide information that was outlined in the lectures and PowerPoint slides so that students can receive answers that better reflect what the professor considers to be an appropriate response.

The defining feature of our chatbot is that it can be easily manipulated by the professor, it targets a specific set of questions that students from a particular class might have, and it creates a versatile response system that allows students to correctly obtain answers to their questions, no matter how the question is worded.

With these aspirations, we chose Google’s natural language processing platform, Dialogflow, to complete the functionality of our chatbot. We considered building a chatbot with Python and machine learning from scratch, however, using Google Dialogflow would significantly speed up our development process and create a more full-featured result. Further, the chatbot will be more easily modifiable because it is not programmed from scratch. With a complete user interface, Dialogflow hides the complexity of machine learning within an easy-to-understand web application.

Typically, a chatbot’s information would be stored in a database. Our chatbot differs in relation to many other chatbots in that it uses an easily modifiable table created in Google Sheets to allow a professor to change the answers to the questions as the semester changes. Most chatbots using Dialogflow will need to be programmed by a computer literate individual, relegating their use to professors that have been trained in computer technologies. Our chatbot will be simpler to use for professors with little computer knowledge due to the integration of spreadsheets.

Further, our chatbot is designed to be user friendly for students, helping them better learn via correctly answering their questions. This is accomplished via targeting a specific set of topics determined by the professor. Students won’t need to search elsewhere for answers, possibly obtaining incorrect information. The machine learning capabilities of the chatbot will allow a student to misspell words, omit proper punctuation, or ask questions in a variety of ways.

The main contributions of this work are as follows:

1. We created a smart chatbot that will provide the correct answer to student questions asked in a variety of ways.
2. We integrated a spreadsheet-based modification system that allows a professor to easily make updates.

3. We focused on integration with Slack, a communication platform that students are familiar with.

The rest of the paper is organized as follows: Section 2 describes the related work. Methodology is presented in section 3. Section 4 goes over the case studies. Future work and conclusion are presented in section 5 and 6.

2. Background and Related Work

When considering the creation of a Slack chatbot, we considered many options, but we ended up creating the chatbot using Google Dialogflow due to its strong natural language processing capability that can greatly reduce our development time and increase our bot’s functionality.

Dialogflow, according to its official documentation, is “a natural language understanding platform that makes it easy to design and integrate a conversational user interface into your mobile app, web application, device, bot, interactive voice response system, and so on” [2]. Natural language processing allows for the creation of software that is capable of comprehending human language [3]. The platform allows us to create intents, entities, and actions for our chatbot. Intents represent the user’s intention. The intention can be a statement such as, “When is the final?” In this case, the user intends to obtain the date of the final for the particular course. The particular intent would be described with a name such as “schedule-final”. An entity will modify an intent. This permits the chatbot to extract keywords that allows it to more accurately choose the correct action for the chatbot. The action is the actual response from the chatbot. An action could elicit a response, change values in a database, or change settings in a program. In our case, the chatbot is designed for text inputs and outputs, however the design allows for future modifications and additions.

Using Dialogflow offered additional benefits that allowed us to further develop our chatbot. Dialogflow included natural integration with Slack that required setup via Slack’s API. In order to make our chatbot functional, we needed to create a Slack application that allowed for secure communication between Slack and Dialogflow. Our Slack application was named “CS436 Homework Helper”. The Slack application allows incoming webhooks that permit the posting of externally sourced messages to Slack. Webhooks allow the server-side application to notify the client-side application of an event occurrence [4]. One can also utilize event subscriptions that allow the chatbot to respond to messages in Slack. To allow this functionality, a request URL is required and verified, obtained from Dialogflow via the “integrations” menu.

Chatbots have many methods of implementation, each providing differing benefits. Ronan Collobert’s chatbot explores chatbot creation from scratch rather than utilizing tools such as Dialogflow [5]. A less resource-intensive chatbot could be developed when compared to a Dialogflow-based chatbot, however, developing a chatbot from scratch requires extensive computational knowledge, time, and effort.

Individuals such as Nudtaporn Rosruen have explored the use of Dialogflow for needs differing from that of education. Rosruen utilized Dialogflow to meet the needs of medical consultants, developing a chatbot based on medical symptoms and treatments [6]. Such needs differ greatly from that of an educational chatbot but utilize very similar principles.

Chatbot’s for educational purposes similar to ours have been developed using Dialogflow as well. Roberto Reyes’ chatbot, also developed for educational purposes, uses Dialogflow but does not include Slack or Knowledge Base integration [7].

An alternative to educational chatbots can be found in live tutoring options such as Wyzant. The cons of such tutoring services in relation to an instructional chatbot are many. A chatbot can be programmed to answer questions directly related to the course whereas a tutor will draw from a more generalized base of information. Also, the cost of running a chatbot is much less than using a live tutor, which typically costs $25 per hour or more [8].

Work by Jiaqi Yin provides evidence for the usefulness of a chatbot as a learning tool by conducting experimental campaigns comparing students using face-to-face delivery methods and those using only a chatbot. The results indicate that students perform just as well using a chatbot in place of face-to-face interaction [9]. Similarly, Suhni Abbasi compared the learning outcomes of students who used Google’s search engine and a chatbot, indicating that the chatbot increased memory retention [10]. Lastly, LuChaciana Benotti compared the use of a chatbot verses the educational programming language, Alice, providing evidence that student interest is enhanced via the use of a chatbot [11].

3. Methodology

The chatbot’s workflow is as follows, depicted in Figure 1. One can see that the student only interacts with the Slack application, asking the chatbot questions. The professor need not interact with the Slack application at all, but will create a spreadsheet and upload it into the Dialogflow Knowledge Base. When the student asks a question via Slack, the Slack API will send data allowed by the OAuth permissions to the Dialogflow Service. The data is retrieved from the Google Cloud and the response is sent back to the Slack API. The Slack API responds to the student via the Slack Application and the communication is completed.
Diving deeper into the methodology behind Slack and Dialogflow’s integration allows one to view the workflow in greater detail, depicted in Figure 2. Dialogflow and Slack integration utilizes OAuth 2.0, a protocol that authorizes an app request, allowing it to install on Slack. OAuth 2.0 is the standard protocol for authorization [12]. Using OAuth requires definition of permission scopes that determine Dialogflow’s capabilities within the Slack channel. An OAuth token, provided by the Slack application, is required by Dialogflow to ensure that Dialogflow is given the permissions determined by the Slack application administrator. A redirect URL is required from Dialogflow by the Slack application to complete the circle of communication between the two services.

The OAuth privileges for our chatbot only allow the chatbot to listen and respond to messages that mention the chatbot directly. Other OAuth scopes allow our chatbot to read the chat in the channel and write text to the channel. It is important to create a narrow scope, only allowing Dialogflow to access necessary functions in Slack to avoid bugs and security issues while ensuring user confidence in the chatbot. Incorrect chatbot behavior could overload the conversation with unwanted responses. A chatbot should not gather data without the user’s consent, maintaining the privacy of the Slack channel and its users. The decision was made to restrict our chatbot to listening only when it was spoken to directly via the statement “@CS436 Homework Helper.” This restriction keeps the chat window clean and allows one to direct their questions at the chatbot only when desired. The professor may consider further restricting the chatbot’s conversation to direct messages or creating a channel specifically designated for questions that the chatbot can access by changing the OAuth privileges.

Another necessary setting within the Slack application is the bot events. Bot events allow the bot user to have access to various events. Such events include messages that mention the bot, messages that were posted to the channel, private messages, and direct messages. Bot events and OAuth scopes must be set similarly to ensure correct functionality of the chatbot. If set incorrectly, Dialogflow could receive the user’s input, process it, but be unable to send the response. Alternatively, Dialogflow may never receive the user’s input in the first place.

Dialogflow’s intents system allows one to write a number of possible responses that could be expected from the user called training phrases. The chatbot could also respond with a variety of answers. In our case, only one response is necessary. We determined that adding further functionality via spreadsheet integration would improve the usability of our chatbot significantly. Dialogflow’s Knowledge Base, a feature currently in beta, is a place to store documents that contain information that will aid Dialogflow in replying to a user’s response. These documents can come in a variety of forms such as a text file, a CSV file, or a Microsoft Excel file. When a user successfully uploads the document into the Knowledge Base, another component of Dialogflow called Knowledge Connectors parses the documents in order to structure the data in a format readable by Dialogflow as a list of intents.

Once these Knowledge Base documents have been parsed, Dialogflow will be able to respond appropriately to a user’s intent. Dialogflow is able to do this by receiving and examining the intent asked by the user. For example, a user may ask the question, “When is the midterm exam?”
and Dialogflow will analyze the components of the question, looking for potential keywords and any related words that could be found in its Knowledge Base. Dialogflow will then go through all of the questions found in its Knowledge Base and rank them based on how closely they match with the user’s original question. The rankings themselves are based on a match confidence scale that goes from 0 to 1.0, where 1.0 is an exact match. Once Dialogflow determines which question has the higher confidence level, Dialogflow will respond back to the user with the appropriate answer from its Knowledge Base. In the example, “When is the midterm exam?”, there is a question in our project’s Knowledge Base with that exact phrasing. Dialogflow will give that question a confidence level of 1.0 and reply back to the user with, “17-Oct-20”. In the situation where the user asks the chatbot a vague question like “exam?”, Dialogflow will still rank the questions and give a response to the user, but the answer may be ranked with a lower confidence level. If the user were to type a question that is not related to the class itself, Dialogflow will ask the user to further clarify the question.

4. Case Studies

There are two relevant use cases that exist when users interact with the chatbot. The first use case includes the upload of the spreadsheet via Dialogflow’s Knowledge Base by a professor. In this case, the professor would input questions in a spreadsheet. The spreadsheet requires two separate columns with the questions on the left and answers on the right. An example is provided in Figure 3. The professor then uploads a new file on their Dialogflow Knowledge Base with the option to delete the old document. Although manual effort is required to prepare the spreadsheet, we plan to automate the process as discussed in our future work.

![Figure 3. An example spreadsheet used as the Knowledge Base of Dialogflow.](image)

From the students’ perspective, the chatbot needs to adequately and accurately answer the questions relevant to the course. When a student asks a question preceded by the chatbot’s name, they will receive a response. The student can ask a question in a multitude of ways and still obtain a relevant response as long as the Knowledge Base contains a matching question associated with an answer. Figure 4 shows a conversation with the chatbot depicting the smart functionality that allows the chatbot to respond correctly despite different questions being asked. According to the related work discussed in section 2, chatbots have proven to be effective tools for students. We also plan to conduct a user study to better understand the effectiveness of a chatbot in helping students learning and engagement.

![Figure 4. A conversation with the chatbot.](image)

5. Discussion and Future Work

Considering the cost, Dialogflow is not entirely free. We are using the standard edition of Dialogflow that provides basic capabilities for chatbot creation. The cost will be 0.2 cents per text request (equivalent to $1 for 500 questions) and 0.65 cents per 15 seconds of audio for users. Overall, the cost would be minimal in the long run.

While our chatbot currently works as intended, there is room for improvement and added functionality. Planned new features include the following: 1) web scraping, 2) automatic spreadsheet updates, 3) notifications, and 4) voice functionality. By adding the above features, the chatbot further adds to its usefulness as a convenient learning tool for both professor and student.

Web scraping would greatly improve the functionality of the chatbot. We plan to auto scrape documents such as the syllabus and course schedule as well as automatically collect the questions and answers from the Slack channel. Such functionality would also allow the reporting of an answer’s source (e.g., syllabus) to students.

Automatic updating of the spreadsheet is another feature we plan to implement as it would streamline the question updating process. Integration between Google Drive and Dialogflow allows the spreadsheet to be easily updated. This functionality would further simplify the implementation of a chatbot for an instructor.

A notification function could be implemented that notifies students about course changes via the chatbot, including the alteration of a due date or the posting of new assignments. This function could also notify the professor specifically if the bot fails to answer the students’ question. The professor would be able to follow up with the student personally and further examine the chatbot response.

Lastly, voice functionality could be implemented to accommodate students who need hands-free access to the chatbot due to a disability or the need to multitask.

6. Conclusion
Modern applications have lowered the barrier to entry for chatbot development. Still, many technologies are required to create a functional chatbot. Important technologies involved are as follows: Machine learning allows the chatbot to accurately respond to a question despite the question’s format. OAuth’s token system allows third-party programs to communicate with each other securely and with necessary restrictions. Dialogflow acts as a natural language understanding platform that utilizes machine learning to simulate the chatbot functionality. Slack’s API allows for the chatbot app creation that is designed to integrate with other third-party applications such as Dialogflow. Google’s cloud storage allows the chatbot’s data to be stored remotely. Lastly, Dialogflow’s Knowledge Base provides a simple way to update and organize the chatbot’s bank of intents. All of these technologies work together to emulate the simple conversational capabilities of a chatbot.

Throughout this project, we have been able to use these tools in order to create a useful and manageable academic resource for both professors and students. As we developed our chatbot and discussed what tasks it should carry out, we considered what we would personally find useful from such a chatbot in our courses. We considered the real questions asked by the students during lectures and in the course’s Slack channel to further determine the requirements for an educational chatbot. Thus, the initial problem involving the difficulties experienced by students with questions is remedied by the chatbot’s convenience and adaptability. The student will spend less time perusing internet sites for unverified answers, avoid distractions commonly present via internet sites, and spend more time engaging in the course material [13]. We believe that by taking advantage of existing technology, students will be encouraged in the learning process. Students need exert little effort to find answers to their questions. Further, our chatbot is extremely simple to setup and modify, so professors can customize the chatbot to fit the specific needs of the class. In turn, the instructor’s valuable time is spent on more pressing matters. We acknowledge that there are many features that would further enhance our chatbot, and we believe that with the right adjustments and updates, our chatbot, along with others, can be a crucial resource for both students and professors.

Acknowledgments

Authors wish to acknowledge the contribution of Taylor Fernandez and Ronalyn Castilla for providing questions and answers for the Knowledge Base spreadsheet and contributing to the creation of the project’s scope.

References


Address for correspondence:
Eric Levas
333 S Twin Oaks Valley Rd, San Marcos, CA, 92024.
levas003@cougars.csusm.edu