International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 SJIF 2023: 8.176 ISSN: 2582-3930

TELEGRAM BOT

T Prasanth[1], DIYONA A B[2], SHALINI H M[3], SHWETHA S P[4], TANUJA N[5]

[1]- Associate Professor, Reva University, Bangalore - dr.tprasanth@gmail.com,
[2,3,4,5] - UG Scholar , Reva University, Bangalore

Abstract: Chatbots or virtual assistants are being used by industries all over the world, they can reduce human intervention and improve efficiency. These days smart-assistants such as Amazon Alexa and Google Assistant help users get quick access to most generic queries within seconds, but when it comes to students and their everyday queries, these assistants fall short in answering the queries they have related to their academic schedules i.e., timetable queries, online classes links, syllabus queries, test dates, etc. The motive behind building this chatbot is to help students get quick and accurate responses to their schedule and syllabus-related queries. Telegram was used to deploy the chatbot onto the real world to enable students to talk to this chatbot from the comfort of their smartphones.

Key words: *Chatbot, API, Telegram, Artificial Intelligence, Google Drive API, OAuth2, Natural Language Processing*

1 INTRODUCTION

Academic information is an important information for students in assisting academic activities, every effort has been developed to improve academic services. With the continuous development of various mobile devices or smartphones, the traditional way of Short Message Service (SMS) has been replaced by instant messaging applications that make the communication process more real-time.

So this research will try to use technology in instant message as a means of academic service information, it is expected study material can be delivered more quickly and up-to-date.

Telegram as one of the instant messaging applications that offers various advantages in its feature than other instant messaging application. The most popular feature and is being developed on telegram is the bot feature, where a third party or user can develop bot features according to user requirements [4].

The telegram can help overcome various problems such as study material access [5,6]. For that made the application of lecture information service using Telegram Bot. With the making of this application, facilitate communication and delivery of study materials to lecturers, students, and the academic community.

API stands for Application Programming Interface. An API is a set of protocols, routines, and tools for building software applications that specify how software components should interact with each other[7].

APIs provide a way for software components to communicate and exchange data with each other, allowing applications to access the functionality of other software components or services. For example, an API might allow a mobile app to access data from a server or allow a web application to access a third-party service like a payment gateway.

The Google Drive API, that can interact with Google Drive, a cloud-based storage service provided by Google. The API allows developers to access, manage, and manipulate files and folders stored on Google Drive, as well as to perform other tasks like sharing files and setting permissions.

The Telegram Bot API is provided by Telegram for building chatbots that can interact with users through the Telegram messaging app. Chatbots are software applications that can simulate human conversation, allowing users to interact with them through natural language.

The Telegram Bot API allows developers to build chatbots that can perform a wide range of tasks, from simple automated responses to complex interactions with external services and databases.

1.1 Overview or background and motivation

To create a Telegram bot using Node.js that can authenticate with the Google Drive API using OAuth2 credentials.

To allow the bot to download a document from Google Drive when provided with a Google Drive link on Windows OS.

To extract the text from the downloaded document using a third-party library and send it back to the user as a Telegram message.

To provide an easy access for student to download the required study materials.

These objectives are designed to ensure that the bot provides a reliable and user-friendly experience for users who want to read documents stored in Google Drive through Telegram.

1.2 Problem Statement

Students are facing difficulty in getting proper academic study material for their studies.

Time constraints: Students who are juggling multiple responsibilities such as work or family may struggle to find time to access and study academic materials, particularly if they are not available online or outside of regular class hours.

Limited access to technology: Some students may not have access to a computer or laptop. Considering the above problem statement, we need to create a Telegram bot that can read different types of documents present in Google Drive when provided with a Google Drive link on Windows OS.

The bot should be able to authenticate with the Google Drive API using OAuth2 credentials, download the document from Google Drive and send back to the user(students) as a Telegram message. The bot should support text documents and PDFs and handle unsupported file types gracefully.

LITERATURE REVIEW

2.1 The aim is to create a chatbot in this study is deve2 loped using Telegram API and webhook method. Other than that, Telegram Bot is used to exchange messages with the users (in this case, the students), higher education institution administrator, and the chatbot. Simple testing shows that all chatbot's functions run well[1].

2.2 The motive behind building this chatbot is to help students get quick and accurate responses to their schedule and syllabus-related queries. This chatbot was developed with the Rasa, it is a framework for developing contextual AI assistants and chatbots. Rasa enables the use of components in the NLU pipeline to customize the intent classification, entity extraction, and response selection. This paper goes through the pipeline customizations that were necessary to process the schedule specific queries from students. Telegram was used to deploy the chatbot onto the real world to enable students to talk to this chatbot from the comfort of their smartphones[2].

2.3 Automatic document categorization divides and organizes text based on a set of specified categories, allowing for quick and easy retrieval of data during the search phase. Text classification, which aims to assign predetermined categories to a given text sequence, has long been a classic task and a popular research topic in the field of natural language processing (NLP). BERT is used in this project, fine-tuned, and applied to documents for automatic document classification. Then, this BERT model is compared to another model called XLNet to see if BERT outperforms XLNet. The accuracy of the BERT and XLNet models are 96.8% and 97.30%, respectively, when tested with test data, indicating that the XLNet model slightly beats the BERT model[3].

3. TOOLS AND METHODOLOGY

3.1 Methodology

Here is a possible methodology for building a Telegram bot using Node.js that can read different types of documents present in Google Drive: Define the scope and requirements of the Telegram bot, including the types of documents to be supported, the expected user interactions, and the desired performance metrics.

Research and select appropriate Node.js libraries and APIs for working with Telegram, Google Drive, and document extraction. Create a development environment for the Node.js application and set up the necessary dependencies.

Define the bot's architecture and design the various components, such as message processing, document analysis, and user interaction.

Implement the bot's functionality using Node.js code, following best practices for code organization, documentation, and testing.

Test the bot's functionality and performance using sample documents and user scenarios .Deploy the bot to a cloud or server environment for production use. Monitor and optimize the bot's performance, using metrics such as response time, error rates, and user feedback.

3.2 Domain Area

3.2.1 Technology and Domain

OAuth2 -a popular authentication framework used by many web applications and APIs, including the Google Drive API, to authorize third-party applications to access user data without sharing login credentials.

NLP-using NLP techniques to improve the bot's ability to understand user queries and provide informative feedback

Artificial Intelligence-Artificial intelligence is the simulation of human intelligence processes by machines, especially computer systems.

3.2.2 Database and Tools

Node.js-an open-source, cross-platform JavaScript runtime environment that allows developers to build server-side applications using JavaScript.

Google drive API- an API provided by Google that allows developers to access and manipulate files stored on Google Drive.

Telegram-Telegram is a cloud-based instant messaging app

3.3 Module Explanation

Telegraf : Telegraf is a modern, powerful and easy-to-use Telegram bot framework for Node.js. It allows developers to easily create and deploy bots on the Telegram platform. In this code, Telegraf is used to handle incoming messages from the user and reply to them based on the contents of the message.

googleapis : The googleapis module is a Node.js client library for using Google APIs. It provides easy-to-use methods for interacting with various Google services, such as Google Drive, Gmail, Google Calendar, and more. In this code, googleapis is used to connect to the Google Drive API and fetch information about the files in a specific folder.

google-auth-library : The google-auth-library module is a Google authentication library for Node.js. It provides authentication credentials for accessing various Google services, such as Google Drive, Gmail, and Google Calendar. In this code, google-auth-library is used to authenticate the user's access to Google Drive, using the API credentials stored in a file.

fs : The fs (File System) module is a built-in module in Node.js that provides methods for working with the file system. In this code, the fs module is used to read the Google API credentials from a file and pass them to the google-auth-library module for authentication.

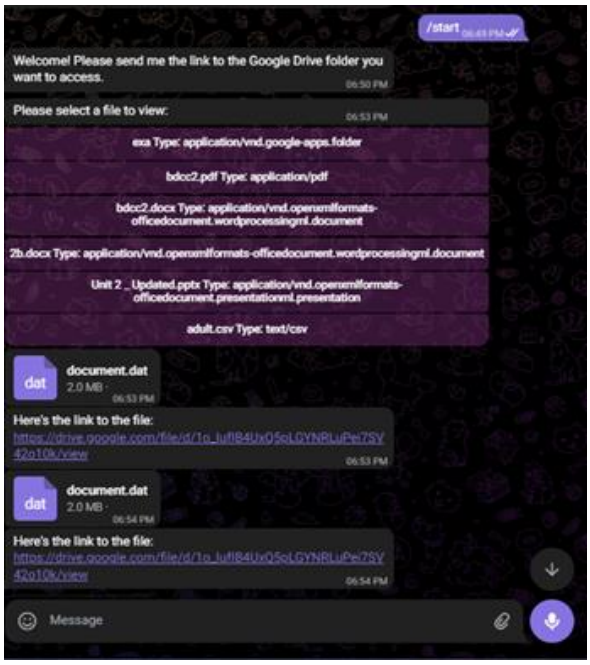

Figure1 - Output Screen

4 CONCLUSION .

Based on the problem statement and proposed methodology, it is feasible to develop a Telegram bot using Node.js that can read different types of documents present in Google Drive when provided with a Google Drive link on Windows OS.

By using a modular architecture, the model can also allow different document extraction libraries to be used depending on the type of document being analysed.

However, it is important to note that building a robust and efficient Telegram bot can require significant time and resources, including expertise in Node.js, Telegram API, Google Drive API, and document analysis.

Additionally, there may be challenges in handling different types of documents with varying formats and structures, as well as ensuring data security and privacy .

Overall, with proper planning, design, and implementation, a Telegram bot for document analysis in Google Drive can provide a valuable tool for students, enabling them to extract insights and required materials from collections of documents.

4.1 Future scope

Here are some potential areas for future work on the Telegram bot for document analysis in Google Drive.

Integration with other messaging platforms: While the current model is built specifically for Telegram, there is potential to extend the model to other messaging platforms such as WhatsApp, or Facebook Messenger.

Expansion of document types: The current model focuses on text-based documents such as PDFs and Microsoft Office files. However, there is potential to expand the model to handle other types of documents such as images, videos, and audio files.

Natural Language Processing (NLP) capabilities: The current model can extract text from documents, but it does not yet have the ability to analyse the content using NLP techniques such as sentiment analysis, topic modelling, or entity recognition

Cloud-based processing: While the current model is designed to run on a local Windows OS .Migrate it to cloud , This could provide scalability, reliability, and cost savings.

User interface and user experience (UI/UX) improvements Future work could focus on designing more intuitive and user-friendly interfaces, voice-based interface.

REFERENCES

[1] Danny Sebastian Teknik Informatika Universitas Kristen Duta Wacana Yogyakarta, “Academic Customer Service Chatbot Development using TelegramBot API”, 2022.
[2] Tanya Dinesh, Anala M R, T. Terry Newton and Smitha G R - “AI Bot for Academic Schedules using Rasa”, 2021.
[3] Parsa Sai Tejaswi, Saranam Venkata Amruth, Prakya Tummala, and M. Suneetha - “Automatic Documents Categorization Using NLP”, 2022.
[4] Dr Geetha M, Akshaya K, Bhuvaneshwari G and Harini H, "Chatbot for Educational Institution," Journal of Xi'an University of Architecture & Technology, 2020, doi: 10.37896/JXAT12.05/1705 (IEEE)
[5] https://auth0.com/intro-to-iam/what-is-oaut
[6] https://youtu.be/6nrP_5ORXEw
[7] https://pipedream.com/apps/google-drive/integrations/telegram-bot-api