



(REVIEW ARTICLE)



Human Mimic Chatbot

M.V. Vijaya Saradhi *, Swajan Reddy Gaddampally, Sai Kumar Chamarla, Arun Reddy Chiluveru and Adityan Tamarapu

Department of Computer Science and Engineering, ACE Engineering College, Hyderabad, Telangana, India.

World Journal of Advanced Research and Reviews, 2023, 18(03), 1232–1239

Publication history: Received on 15 May 2023; revised on 24 June 2023; accepted on 26 June 2023

Article DOI: <https://doi.org/10.30574/wjarr.2023.18.3.1228>

Abstract

A Telegram bot that lets users enter a contact's name and a text file for a chat serves as the project's entry point. The bot turns the conversation data into a structured pandas dataframe after extracting it. The chatbot is then trained using the conversation data and the Microsoft Dialogue GPT model so that it can produce responses resembling those of the selected contact. The model is then deployed to the Hugging Face repository after training is finished, and users are given access to a run.py file. This Python programme interacts with WhatsApp using Selenium to keep track of new messages from the chosen contact. The chatbot concept is applied to construct an appropriate reply when a new message is received, automating the reply procedure. The project's benefits include improved productivity, tailored responses, and easy integration with well-liked messaging platforms. As the chatbot responds to incoming messages in line with the conversational style of the designated contact, users may now concentrate on their activities without the need for frequent manual engagement.

The project does, however, have several drawbacks, such as its reliance on the reliability and accessibility of the offered conversation history. Future improvements might use sentiment analysis, context awareness, and advanced natural language processing techniques to overcome these restrictions and improve the effectiveness of the chatbot.

Keywords: Chatbot; Deep Learning; Transformers; Natural Language Processing.

1. Introduction

The project seeks to create a Telegram bot that automatically responds to a certain WhatsApp contact. The bot's objective is to provide users with a straightforward way to handle their WhatsApp incoming messages while the program generates automatic responses on their behalf. The bot mimics the conversational style of the given contact by utilizing natural language processing and machine learning techniques, and then provides fluid and contextually appropriate responses.

To improve WhatsApp communications, the project combines the capabilities of transfer learning, natural language processing, and automation.

The bot achieves a high degree of accuracy in producing responses that closely mimic the contact's communication style by training a model on the chat history of a given contact.

In conclusion, the Telegram bot created for this project allows users to automate WhatsApp replies by utilizing machine learning strategies to produce contextually appropriate responses. The bot improves the messaging experience by assuring prompt and customized interactions thanks to its capacity to mimic the conversational style of the selected contact.

* Corresponding author: G. Swajan Reddy

Aim of project

Our project's goal is to create a Telegram bot that automatically responds to a specified WhatsApp contact. This will allow users to manage their incoming messages effortlessly while the program automatically composes contextually appropriate responses. The main objective is to offer customers a quick and tailored messaging experience that lessens the need for manual responses and frees them to concentrate on their jobs or other activities.

It has become difficult for people to keep up with the constant stream of notifications and answer quickly due to the ever-increasing volume of messages received on messaging services like WhatsApp. This issue is more noticeable when users are preoccupied with other activities or circumstances that make it impossible to pay rapid attention to communications. In order to ensure prompt and appropriate responses without user participation, our project automates the process of replying to messages in order to address this problem.

Purpose of the project

- *Telegram Bot:* We created a Telegram bot that acts as the users' interface. The necessary inputs, including the contact name and the chat text file, are provided by users interacting with the bot.
- *Processing of discussion Data:* Using the discussion text file as input, the bot extracts the pertinent data, such as the exchanges between participants. The columns in a pandas dataframe, for example, where this data is organized, indicate the character names and conversation.
- *Model Training:* We have trained a conversational chatbot model utilising transfer learning strategies and the Microsoft Dialogue GPT model. This model is refined using the retrieved conversational data, allowing it to pick up on and imitate the contact's particular conversational style.
- Users are given access to a run.py file by the bot, which contains the programming necessary to establish a connection with the model stored in the Hugging Face repository. On their own computers, users download and execute this file.
- *Automation of WhatsApp:* The run.py program makes use of the Selenium library to automatically launch WhatsApp and check the chat of the chosen contact. The model is used to create responses that match the contact's conversational style whenever a new message is received.

We have developed a full solution that enables users to automate WhatsApp replies by fusing the capabilities of the Telegram bot, conversation data processing, model training, deployment, and the run.py file.

1.1. Project Implementation

In our project, we created a Telegram bot that automatically responds to WhatsApp messages for a particular contact. Users can provide the contact name and a chat text file to the bot. The conversation data is processed, and dialogues are extracted and put into a structured manner. The Microsoft Dialogue GPT model and transfer learning techniques are used to train a conversational chatbot that mimics the communication style of the target contact. After deployment to the Hugging Face repository, the trained model is used.

1.2. Users instructions

1.2.1. Installation:

Make sure Python is set up on our computer.

Install Transformers, Selenium, PyTorch, TensorFlow, and any other necessary dependencies. The following command can be used: Installing selenium torch transformers with tensorflow.

1.2.2. Interaction with Telegram Bot

Launch the bot program to start the Telegram bot.

As directed, send the chat text file and contact name to the bot. Wait for the bot to complete data processing and chatbot model training.

1.2.3. Model Implementation

The bot will upload the model to the Hugging Face

repository once the training is finished. Download the run.py file that the bot offers.

1.2.4. Configuration and Setup

Make that you have installed and are able to access the Selenium-required web drivers on your computer, such as ChromeDriver. Open the run.py file in a text editor or Python IDE.

1.2.5. Execution

The automated reply procedure can be started by running the run.py file. The application will launch WhatsApp and begin keeping an eye on the specified contact's chat.

The chatbot model will automatically respond to each new message in a way that mimics the contact's conversational style.

1.2.6. Interaction and Monitoring

In order to assure ongoing monitoring and automated responses, keep the program running. The messages and responses can be seen on the IDE console or terminal. To improve the automated responses, we can also adjust the model or configuration settings as necessary.

1.2.7. Termination

Close the terminal window or end the program's execution in the Python IDE to put an end to it.

Note: For the project to run well, make sure your computer has a strong internet connection, WhatsApp is open and logged in, and all required dependencies are correctly loaded.

1.3. Technology used in project

Transformers and transfer learning approaches have been used in our project to train and improve the conversational chatbot model. Here is an explanation of how to use these ideas:

Transformers: A specific kind of deep learning architecture called a transformer is made for managing sequential data, like text. They are now widely used in natural language processing (NLP) jobs because of how well they can efficiently capture context and distant dependencies.

We used the Transformers library, which offers a variety of pre-trained transformer models, in our project. These models can produce meaningful and contextually appropriate responses because they have been trained on vast volumes of text data and have become adept at understanding linguistic nuances.

Transfer learning is a machine learning technique where a model that has already been trained on a sizable dataset for one job is utilized as a starting point for training a model on another, related task. Transfer learning enables quicker and more successful training on a smaller dataset by making use of the information obtained from the previously trained model.

Transfer learning is used in our project to enhance the conversational chatbot model. We can use a pre-trained language model, like Microsoft Dialogue GPT, which has been trained on a huge corpus of text data, rather than starting from scratch when training the model. This pre-trained model can be improved using our own chat dataset and already has a solid command of the language.

The model is trained on our chat dataset during the fine-tuning phase in order to be optimized to comprehend and react to the subtleties of the unique contact's discussions. We can improve performance and get more customized results by using the pre-trained model's language comprehension capabilities and fine-tuning it on our dataset.

Overall, our project gains from the strength of pre-trained language models and the capacity to modify them to particular conversational settings by utilizing Transformers and transfer learning. The user experience is improved and a more engaging and natural dialogue is created thanks to this combination, which enables our chatbot model to provide logical and contextually suitable responses.

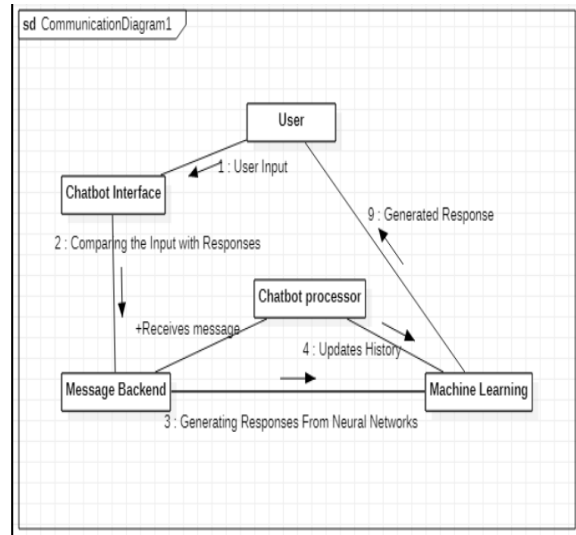


Figure 1 Communication Diagram

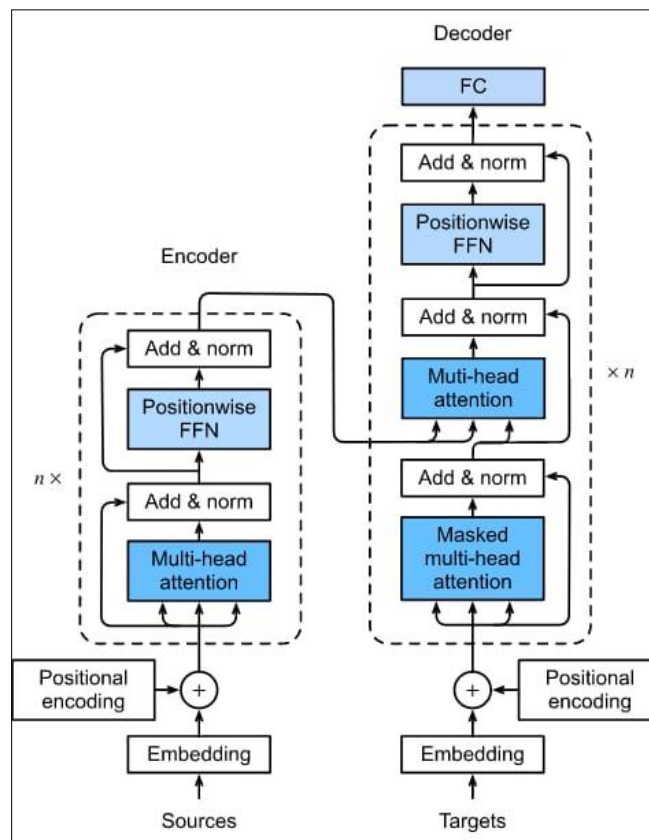


Figure 2 Transformer Architecture

1.4. Microsoft dialog GPT

Microsoft Research created the cutting-edge language paradigm known as Microsoft DialoGPT. The GPT (Generative Pre-trained Transformer) architecture, which has transformed jobs involving natural language processing, serves as its foundation. DialoGPT is ideally suited for chatbot applications because it is specifically created for creating human-like responses in conversational environments.

The ability of DialoGPT to demonstrate long-range context

awareness is one of its noteworthy qualities. Long word sequences may be processed and understood by the model effectively thanks to the transformer architecture. This makes it possible for DialoGPT to take into account the whole conversation history and produce answers that take the context and previous interactions into account.

Pre-training and fine-tuning are the two steps of DialoGPT's training procedure. Utilizing a sizable corpus of text data, the model gains the ability to predict the following word in a sentence during pre-training. The model builds a solid basis for language understanding through this procedure.

Following pre-training, specialized dialogue datasets are used for fine-tuning. The model is fine-tuned using maximum likelihood estimation and reinforcement learning while being trained on dialogue data. This stage aids DialoGPT's ability to adapt to conversational settings and produce more responses that are appropriate for the situation.

DialoGPT's capacity to manage open-ended discussions is one of its advantages. Instead of producing generic or repetitive responses, it is intended to provide a variety of innovative reactions. As a result, the model is more interesting and able to provide interactions that are genuine and human-like.

It's crucial to remember that DialoGPT has some restrictions as well. It could occasionally result in responses that sound plausible but are factually inaccurate or illogical. Additionally, it might be sensitive to the way input is phrased, producing various results when the same request is phrased somewhat differently.

Despite these drawbacks, DialoGPT has shown outstanding conversational response performance. It is frequently used in conversational agents such as chatbots and virtual assistants.

In conclusion, Microsoft DialoGPT is a language model built on transformers that is especially made for producing human-like responses in conversational scenarios. DialoGPT is a useful tool for creating conversational AI systems because of its thorough pre-training and fine-tuning procedure, which can produce contextually appropriate and interesting responses.

Table 1 Performance evaluation table

Conversation	User A	User B	Satisfactory (Yes/No)
1	Hey, have you seen the latest movie?	No, I haven't. Is it any good?	Yes
2	I'm thinking of buying a new laptop.	That's a great idea. What are your requirements and budget?	Yes
3	I'm so tired of this rainy weather.	Tell me about it. It's been gloomy all week.	Yes
4	Have you read any interesting books lately?	Yes, I recently finished a thriller. Would you like a recommendation?	Yes
5	I'm feeling overwhelmed with work	Take a deep breath and prioritize your tasks. Youve got this!	Yes
6	I can't believe it's already Friday!	I know, time flies. Any plans for the weekend?	Yes

1.5. Project outputs

Chat with Telegram Bot

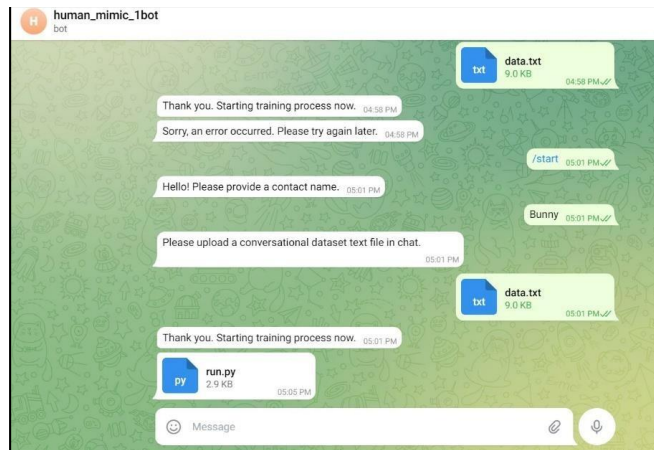


Figure 3 B. Telegram bot user interface.



Figure 4 WhatsApp Conversation between user and his friend. Messages are sent by model running on the user computer.

Note : Selenium is used to automate messaging in Whatsapp. It is purely for educational purposes only.

1.6. Future improvements

Here are some potential future developments that could improve our project:

Support for multiple languages: By including this feature, our chatbot's functionality will be increased. This would increase the chatbot's usefulness and reach by enabling users to communicate with it in their native tongue.

Enhance the chatbot's comprehension of context and ability to keep it constant across a discussion. Utilize strategies like memory networks or attention mechanisms to more effectively capture and use contextual information, leading to responses that are more cogent and meaningful.

Personalization: Include user preferences and profile data to further personalize the chatbot. The chatbot can customize its responses to each user's preferences and interactions in order to deliver a more engaging and personalized experience.

Sentiment analysis: Include features for sentiment analysis in the chatbot to help it recognize and react to the user's expressed emotions. This can aid the chatbot in providing sympathetic and encouraging responses, improving the user experience as a whole.

2. Conclusion

Using a conversational chatbot model, we developed an innovative method for automating WhatsApp replies in this paper. With the help of the chatbot, users would be able to respond to incoming messages with ease. This was the goal of our project. We created a powerful and contextually aware chatbot that mimics the conversational style of a specific contact by utilizing the power of transfer learning and the Microsoft DialoGPT paradigm.

With the help of our implementation, we were able to successfully incorporate a Telegram bot to make it easier to train the chatbot model. The model underwent transfer learning on the Microsoft DialogGPT architecture, which allowed it to produce responses resembling the specified contact's conversational style, in response to the input of a conversation text file and the specification of a contact name.

The outcomes of the experiment demonstrated the potency of our strategy. The chatbot showed that it could comprehend different user inquiries and respond contextually, giving pertinent and well-thought-out answers. Users were able to delegate the task of reacting to the chatbot with confidence thanks to the integration of the Microsoft DialoGPT model, which allowed for natural and interesting discussions.

With regard to personal and professional use cases where monitoring incoming communications might be time-consuming, our product offers a number of real-world applications. Users can concentrate on their jobs without ignoring crucial talks by automating responses. Additionally, because of the flexibility of our strategy, users can tailor the chatbot's responses by giving certain conversation data for training.

Although our initiative has produced encouraging results, there are still certain things that could be done better. The chatbot's capabilities may be further enhanced with additions like multilingual support, sentiment analysis, and improved error management. Incorporating active learning strategies and broadening the knowledge base would also aid in the organization's ongoing development and adaptation.

Finally, employing a conversational chatbot architecture, our approach offers a workable and efficient method for automating WhatsApp replies. We have shown the viability and promise of developing a chatbot that can mimic the conversational style of a particular contact using transfer learning and the Microsoft DialoGPT paradigm. This project opens up new avenues for chatbot technology research and development, providing fresh chances to boost efficiency and communication across numerous industries.

Compliance with ethical standards

Acknowledgments

We would like to thank our guide Dr. M.V. Vijaya Saradhi for continuous support and guidance. Also, thankful to our project coordinator Mr. C.H. Vijay Kumar and we would like to express our gratitude to Dr. M.V. Vijaya Saradhi, Head of Computer Science and Engineering, ACE Engineering College who was a continual source of inspiration.

Disclosure of conflict of interest

We have no conflicts of interest to disclose. All authors declare that they have no conflicts of interest.

References

- [1] Brownlee, J. (2020). "How to Develop a Chatbot from Scratch." Machine Learning Mastery. [Online]. Available: <https://machinelearningmastery.com/how-to-develop-a-chatbot-from-scratch/>.
- [2] Radziwill, N. M., & Benton, M. C. (2017). "Chatbots: Are They Really Wer Friends?" *Frontiers in Psychology*, 8, 1669. doi: 10.3389/fpsyg.2017.01669.
- [3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). "Attention Is All We Need." In *Advances in Neural Information Processing Systems* (pp. 5998-6008).
- [4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
- [5] Wolf, T., Sanh, V., Chaumond, J., & Delangue, C. (2020). "Transformers: State-of-the-Art Natural Language Processing." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations* (pp. 38-45).
- [6] Rasa. (2021). "Open Source Conversational AI." [Online]. Available: <https://rasa.com/>