Al-Mustansiriyah Journal of Science
ISSN: 1814-635X (print), ISSN:2521-3520 (online) Volume 34, Issue 2, 2023 DOI: http://doi.org/10.23851/mjs.v34i2.1350

Research Article **Open Access**

Enhancement of ChatGPT using API Wrappers Techniques

Muhanad Tahrir Younis[1], Nadia Mahmood Hussien[1], Yasmin Makki Mohialden[1*], Komeil Raisian[2], Prabhishek Singh[3], Kapil Joshi[4]

[1]Department of Computer Science, Collage of Science, Mustansiriyah University, 10052 Baghdad, IRAQ.
[2]Faculty of Information Science and Technology, the National University of Malaysia, Selangor, MALAYSIA.
[3]School of Computer Science Engineering and Technology, Bennett University, Greater Noida, INDIA.
[4]Department of Computer Science and Engineering, Uttaranchal University Dehradun, INDIA.

[*]Correspondent contact: ymmiraq2009@uomustansiriyah.edu.iq

Article Info

Received
19/03/2023

Accepted
27/04/2023

Published
30/06/2023

ABSTRACT

This study looks at how API (Application Programming Interface) wrapper technology can make it easier to use complex functions by putting together a lot of API calls. These packages have non-real-time interfaces that are hard to use. ChatGPT is a chatbot-specific GPT-3 language paradigm. It lets developers create chatbots that respond intelligently to natural language user input, creating a more engaging user experience. This article shows that ChatGPT, Python, and API wrapper technology may be used to develop a smart chatbot. We show how to use the OpenAI API library to add ChatGPT to Python programs. This makes it easier for developers to make chatbots that sound and act more like real people when they talk. Our contribution to this field is showing that it is possible to make smart chatbots with ChatGPT and API wrapper technology. To reach this goal, we use a system that combines the OpenAI API with ChatGPT and Python. This gives us valuable information about how to make smart chatbots. The efficiency of the system has been tested many times while applying it to different environments, and the results are satisfactory.

KEYWORDS: API wrappers, ChatGPT, GPT-3, OpenAI, HTTP requests

الخلاصة

تهدف هذه الدراسة إلى التحقق من كيفية استخدام تقنية API wrapper لتسهيل تنفيذ الوظائف المعقدة عن طريق تجميع الكثير من استدعاءات API. توفر هذه الحزم واجهات قابله للاستخدام ليست في الوقت الحقيقي والتي قد تكون صعبة الاستخدام بشكل مستقل. ChatGPT هي النسخة الخاصة بروبوتات الدردشة من نموذج اللغة GPT-3. يمكن للمطورين من خلالها بناء روبوتات الدردشة التي يمكنها الاستجابة بذكاء لإدخالات المستخدم باللغة الطبيعية، مما يؤدي إلى تجربة مستخدم أكثر إثارة للاهتمام. يهدف هذا البحث إلى إظهار إمكانية بناء روبوت دردشة ذكي عن طريق دمج ChatGPT و Python مع تقنية API wrapper كذلك وضحنا كيفية استخدام مكتبة OpenAI API لإضافة ChatGPT إلى برامج Python، مما يجعل من الأسهل على المطورين جعل روبوتات الدردشة تبدو وتتصرف بشكل أكثر واقعية عندما يتحدثون. يتمثل مساهمتنا في هذا المجال في إظهار أنه من الممكن صنع روبوتات دردشة ذكية باستخدام ChatGPT وتقنية API wrapper. لتحقيق هذا الهدف، نستخدم نظامًا يجمع بين مكتبة OpenAI API و ChatGPT و Python، مما يمنحنا معلومات قيمة حول كيفية صنع روبوتات دردشة ذكية. تم اختبار كفاءة النظام عدة مرات أثناء تطبيقه على بيئات مختلفة ، وكانت النتائج مرضية.

INTRODUCTION

When there are multiple APIs in the same domain, new requirements may require API migration and re-engineering of the application to make it less dependent on a single API. Putting a wrapper around the original API makes it possible to use another API. Wrapper-based migration is useful because it lets the code of the application stay the same, and the wrappers can be used with more than one application. Even if the APIs that require wrapping are new, the wrapper design is still critical because the APIs in this scenario tend to be vastly different [1]. API libraries, which provide thousands of APIs, are essential for daily programming tasks.

APIs have long been studied by researchers. Formal specifications that define how APIs can be used to find legal uses have been studied [2].

ChatGPT is the most advanced chatbot in the world. It can write impressive prose in a matter of seconds, which distinguishes it from other chatbots. It has generated a lot of hype and doomsday predictions, particularly about how students are evaluated in college and other things. ChatGPT is a modern language model that is a variant of OpenAI's Generative Pretrained Modifier (GPT) language model. It is intended to generate fuzzy text from human-written text. It can simulate natural interactions with users [3]. This technology has an impact on both academic research and society at large, with researchers and others using ChatGPT and other large language models to write articles, discussions, summaries, draft and improve papers, identify study gaps, and write computer code with numerical analyses. Soon, this technology will be able to generate tests, write and complete manuscripts, conduct peer reviews, and make editorial decisions [4]. ChatGPT's authors are highly regarded by researchers [5]. In this paper, we propose using API integration methods to create an intelligent ChatGPT bot.

RELATED WORK

There are several works that can be referenced, including:

Brian Lucey and Michael Dowling (2023) [6] based on how finance journal reviewers rated the results, they demonstrated that the newly released AI chatbot ChatGPT can aid in the study of finance in a meaningful manner. In principle, these findings should apply to all areas of research. There are obvious incentives for coming up with ideas and gathering data. However, the technology isn't as effective at putting together a summary of the literature or creating good testing frameworks. They made it obvious that the value of the output is determined by how much private data and domain knowledge went into it. They arrived at this conclusion after considering what this new technology implies, particularly in terms of ethics. After a month, the authors of Taecharungroj, V, (2023) [7] are still tweeting about ChatGPT, a sophisticated AI chatbot. 233,914 English tweets were investigated to answer the question "What can ChatGPT do?" using latent Dirichlet allocation (LDA) topic modeling. There was new information, technology, and reactions. The author also cited creative writing, essay writing, prompt writing, code writing, and answering questions. The research discovered that ChatGPT could have an impact on both people and technology, both positively and negatively. The author concludes that this AI advancement presents four major issues: job evolution, a new technological landscape, the push for artificial general intelligence, and the progress-ethics conundrum.

Fan Huang, (2023) [8] investigates whether ChatGPT can be used to provide natural language explanations (NLEs) for implicit hate speech detection. They make prompts to get short NLEs from ChatGPT, and they do user studies to compare their strengths and weaknesses with the strengths and weaknesses of NLEs made by other people. In the context of implicit hate speech research, they talk about the pros and cons of ChatGPT. Kuzdeuov A, (2023) [9], talks about a speech-based system that lets people who are blind or have trouble seeing use ChatGPT. The system employs automatic speech recognition (ASR), text-to-speech (TTS), and a Telegram bot to provide people who are blind or have poor vision with a normal way to use ChatGPT. Their early tests showed that the suggested system could be used as an assistive technology. Hatim Abdelhak Dida, (2023) [10], they suggest using the advanced language model ChatGPT to improve TTS (text-to-speech) for big data. They compare the performance of the ChatGPT-based TTS conversion system to that of current TTS systems using a central real-world big data dataset. Their experimental findings show that ChatGPT can significantly improve the quality and effectiveness of TTS conversion for big data. Most of the above studies discuss the development of ChatGPT. In the proposed system, it suggested a new technology to make smart chatbots with ChatGPT and API wrapper technology.

MATERIALS AND METHODS

Proposed System

The current ChatGPT API uses the REST API technique to provide information to users. The API can be retrieved using the Python OpenAI module. Developers can use APIs to access OpenAI's services, but they must first register for an API key. The suggested next steps are as follows:
Stage 1: Create an API pass.

Stage 2: Import the OpenAI module.

Stage 3: Select the GPT-3 variant. ChatGPT is based on GPT-3, which has four primary models:

1. Da Vinci is the most capable GPT-3 model. It can do any job that the other models can do, often with better quality, longer output, and better following of instructions. It can also create word completions.
2. Curie: a highly capable model, but Da Vinci is faster and less expensive.
3. Babbage: capable of handling simple jobs, very fast performance, and lower expense. Ada: Ada can do simple tasks and is usually the fastest and least expensive model in the GPT-3 series.

Stage 4: Add a final touch. To begin a chat conversation with ChatGPT, we must first finish a prompt. The idea behind completion is that we provide some text as a prompt, and the model will try to complete our sentence. The create() method accepts the following arguments:

1. engine: the model to be used.
2. prompt: the statement to be sent to the chatbot
3. max tokens: the maximum number of tokens to generate in the completion. If you make it too low, the response may be incomplete.
4. temperature: a number between 0 and 2. A lower number makes the output more deterministic.

Figure 1 depicted the proposed system's general diagram. When an API call is generated, an API endpoint is undoubtedly specified. The API endpoint is the source of the API call or the resource that gets the API calls. API calls are typically accepted by servers or web applications. As a result, they are API endpoints, responsible for accepting API calls, processing them, capturing information, and delivering the response, answer, or output.

To guarantee that API calls reach the desired API endpoint. To accomplish this, each API request includes a URI. This is the business standard for determining the appropriate destination. URI can refer to a server, application, email contact, or website, depending on the circumstance. In the case of web APIs, URI becomes URL.

It is used to identify the designated internet destination. An ideal and feasible URL includes an application-layer protocol such as HTTP or HTTPS. HTTPS is more secure than HTTP and is mostly used by API security-conscious companies. HTTP-based API calls, on the other hand, are very prevalent because most web APIs are HTTP-based. HTTP API calls use normal HTTP verbs such as POST, PUT, and GET. API calls determine what kind of information or data that API location should provide based on the HTTP verb used. The GET request is used in the proposed system.

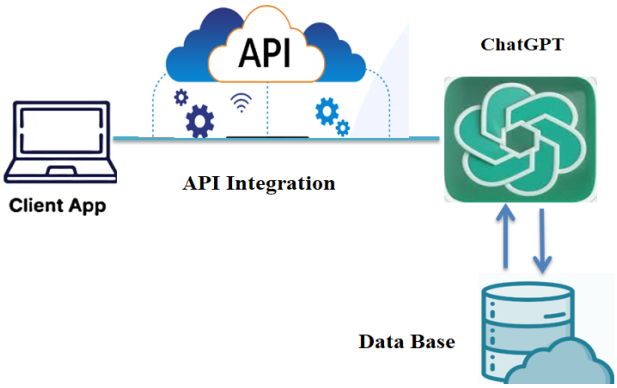

Figure 1. proposed system's general diagram.

Testing

The recommended way is to design and implement chatbots using API wrappers and an API integration system. The outcomes of ChatGPT, or any language model for that matter, can vary depending on the particular system of integration and usage.

The proposed system makes use of an API integration technique based on the OpenAI Python package, which offers powerful tools and resources for developers working on NLP and machine learning projects. Using pre-trained GPT models, fine-tuning tools, and other NLP tools, developers can make sophisticated language models and get useful information from text data. Figure 2 and Figure 3 depict system testing and demonstrate how the system works correctly: the user enters his query, and a smart Chatbot responds during the chat session. To compare the proposed method with other studies, Table 1 compares using ChatGPT via an HTTP API to other methods.

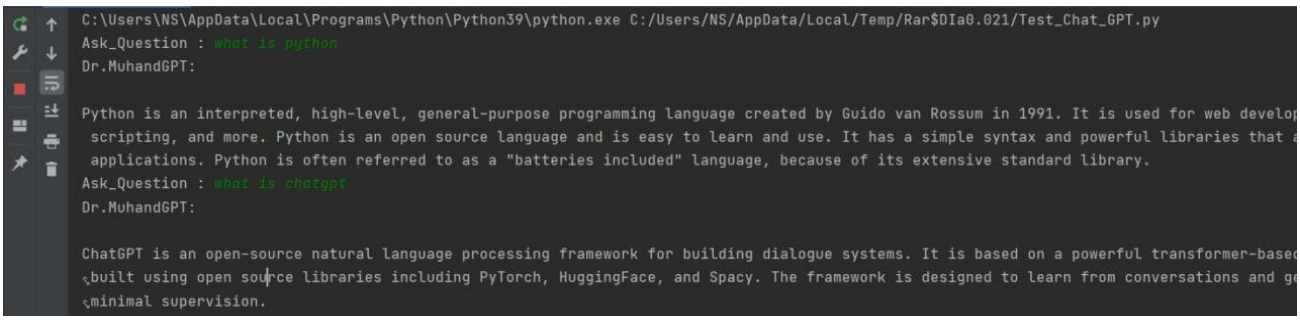

Figure 2. The first test result.

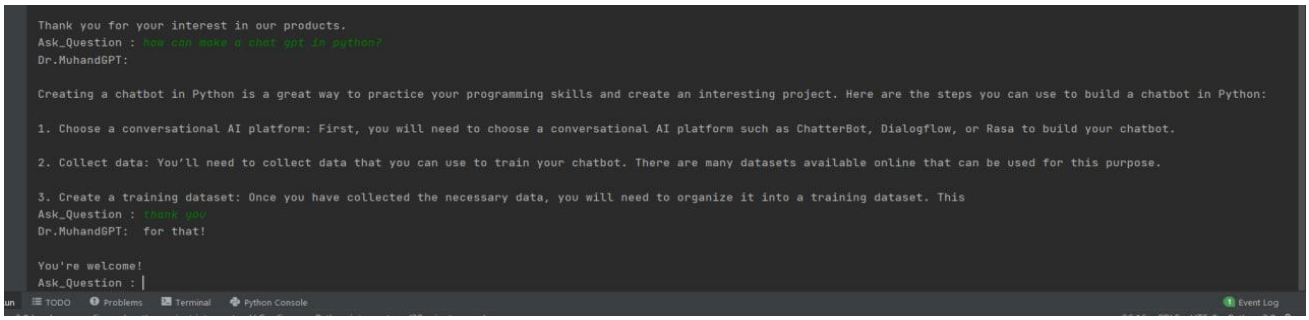

Figure 3. The second test result

Table 1. Compares using ChatGPT via an HTTP API to other methods.

Factors	API-based integration	Direct integration	Quality of training data	Customization
Response time	Slower than direct integration	faster response times.	slow in response	The response time is determined by the use case chosen.
Quality	Very good	Very good	Poor	Good
Implementing	Easy in implementation, it involves making HTTP requests and returning the response in JSON format.	Requires more technical expertise to implement because there is no overhead from making HTTP requests.	It is used to train ChatGPT and can affect the quality of the generated responses.	It used to customize and train ChatGPT based on specific use cases that needed

CONCLUSIONS

The proposed ChatGPT API system is a simple way for developers to use the OpenAI GPT-3 models' features when making chatbots. The general steps in design such systems include: making an API key, importing the OpenAI library, choosing a GPT-3 model, and giving a prompt to make a final change before starting a chat with the ChatGPT API. The API gives developers four GPT-3 models with different levels of capability, speed, and cost. This lets them choose the model that works best for their application. The technique can be an efficient and effective way to create a chatbot that can perform a variety of tasks, making it a valuable tool for developers looking to improve user experiences in a variety of industries. The system can be enhanced to include more facilities which can make enhanced from answer

ACKNOWLEDGMENT

The Authors would like to thank Mustansiriyah University(https://uomustansiriyah.edu.iq/) Baghdad –Iraq for its support in the present.

Disclosure and Conflict of Interest: The authors declare that they have no conflicts of interest.

REFERENCES

[1] T. T. Bartolomei, K. Czarnecki, and R. Lämmel, "Swing to SWT and back: Patterns for API migration by wrapping," in 2010 IEEE International Conference on Software Maintenance, 2010, pp. 1-10: IEEE. https://doi.org/10.1109/ICSM.2010.5610429

[2] H. Zhong and H. Mei, "An empirical study on API usages," IEEE Transactions on Software Engineering, vol. 45, no. 4, pp. 319-334, 2017. https://doi.org/10.1109/TSE.2017.2782280

[3] J. Rudolph, S. Tan, and S. Tan, "ChatGPT: Bullshit spewer or the end of traditional assessments in higher

education?," Journal of Applied Learning Teaching, vol. 6, no. 1, 2023. https://doi.org/10.37074/jalt.2023.6.1.9

[4] E. A. van Dis, J. Bollen, W. Zuidema, R. van Rooij, and C. L. Bockting, "ChatGPT: five priorities for research," Nature, vol. 614, no. 7947, pp. 224-226, 2023. https://doi.org/10.1038/d41586-023-00288-7

[5] A. H. Ali, M. G. Yaseen, M. Aljanabi, and S. A. Abed, "Transfer Learning: A New Promising Techniques," Mesopotamian Journal of Big Data, vol. 2023, pp. 31-32, 2023. https://doi.org/10.58496/MJBD/2023/004

[6] M. Dowling and B. Lucey, "ChatGPT for (finance) research: The Bananarama conjecture," Finance Research Letters, p. 103662, 2023. https://doi.org/10.1016/j.frl.2023.103662

[7] V. Taecharungroj, ""What Can ChatGPT Do?" Analyzing Early Reactions to the Innovative AI Chatbot on Twitter," Big Data Cognitive Computing, vol. 7, no. 1, p. 35, 2023. https://doi.org/10.3390/bdcc7010035

[8] F. Huang, H. Kwak, and J. An, "Is ChatGPT better than Human Annotators? Potential and Limitations of ChatGPT in Explaining Implicit Hate Speech," arXiv preprint arXiv:.07736, 2023. https://doi.org/10.1145/3543873.3587368

[9] A. Kuzdeuov, S. Nurgaliyev, and H. A. Varol, "ChatGPT for Visually Impaired and Blind," TechRxiv, 2023. https://doi.org/10.36227/techrxiv.22047080.v1

[10] H. A. Dida, D. Chakravarthy, and F. Rabbi, "ChatGPT and Big Data: Enhancing Text-to-Speech Conversion," Mesopotamian Journal of Big Data, vol. 2023, pp. 33-37, 2023. https://doi.org/10.58496/MJBD/2023/005

How to Cite

M. T. Younis, N. M. . Hussien, Y. M. . Mohialden, K. . Raisian, P. . Singh, and K. . Joshi, “Enhancement of ChatGPT using API Wrappers Techniques”, Al-Mustansiriyah Journal of Science, vol. 34, no. 2, pp. 82–86, Jun. 2023.