

Platform-Free Mobile Application: Chatbot That Uses ChatGPT

Marcos Pinto

Darling Cespedes

Kazi Tasin

Orin Gillis

NYC College of Technology, CUNY
Brooklyn, NY 11201, USA

Abstract

The mobile application enables any individual to communicate with a chatbot via a mobile device using natural language and receive replies on a wide range of topics, including business ideas, freelancing, blogging, email marketing, essay creation, coding, ebooks, and more. The choice of ChatGPT as the backend chatbot is justified by its growing popularity, as it is currently considered one of the most significant chatbots that leverages artificial intelligence to gather and curate data from various sources. It is a hands-on example that can be incorporated into college courses on Web services and/or mobile applications.

1. Introduction

Most people rely on technology every day and every hour of their lives. One of the most challenging issues we currently face is how to extract valid meaning from the immense amount of data we collect daily. Artificial intelligence (AI) has been introduced to, among other things, find meaning, trends, and interconnections between data. Data science has emerged as the field that combines principles and practices from mathematics, statistics, artificial intelligence, and computer engineering to analyze large amounts of data.

Our application is designed to address these fields of data science and AI. Essentially, it serves as a facilitator for a chatbot, which is a piece of software that mimics the interaction between a human and a machine, either vocally or through written text. The dialogue involves two parties: the user and the chatbot. Our chosen chatbot is ChatGPT, developed by OpenAI and launched in November 2022. ChatGPT is at its core a machine learning application that relies on a neural network and is trained on a vast dataset, spanning terabytes of data.

OpenAI, founded in 2015 in California, feeds its datasets with "different perspectives, voices, and experiences that form the full spectrum of humanity," according to its profile. In other words, ChatGPT enhances itself through users' interactions by training its algorithm with this data. Consequently, privacy concerns and issues related to sharing personal data arise. In simpler terms, we don't know how the algorithm is trained, what specific data is used, or where it comes from. Additionally, if there is bias in the incoming message, there may be bias in the bot's response.

The bottom line is that we should approach the bot's answers with a degree of suspicion because the response presented by the bot could originate from a questionable source. Furthermore, it lacks access to real-time information or knowledge beyond its training data. It's essential to remember that the bot operates within a defined scope, while humans can work within or outside of those boundaries.

However, on the positive side, due to its rapid and straightforward responses, the bot can serve as an aide to individuals, businesses, or educational institutions in certain situations, such as customer service operations, answering frequently asked questions, and troubleshooting technical problems within organizations.

Our proposed platform is versatile; it can run on both Android and iOS devices as a mobile application. Besides potentially offering reliable instructions and opinions, it can serve as auxiliary assistance in situations where resources are scarce or specialists are absent in certain fields. Therefore, gaining access to this type of mobile application may result in life-saving moments or improve the lives of underserved communities.

The range of situations where this application can provide assistance is quite extensive, considering that knowledge is an ever-changing subject. For example, in agriculture, ChatGPT can be trained on vast amounts of agricultural data, including information about soil conditions, weather patterns, and pest control. This enables it to provide farmers with personalized recommendations and advice on how to optimize their crops. Similarly, in the pharmaceutical industry, ChatGPT can analyze extensive data to identify patterns and trends, assisting researchers and scientists in making more informed decisions and expediting the drug discovery process. The same can be applied to the financial world, where the bot can analyze financial data, assist banks in identifying and managing potential risks, and flag suspicious transactions or potential fraud.

In this project, we use Flutter, a platform-independent mobile technology developed by Google. Flutter comprises an SDK (Software Development Kit) with an editor, compiler, and several libraries for application development. Flutter has its own programming language, Dart, which is an object-oriented programming language (OOP) somewhat similar to Java. It allows for the development of mobile, desktop, and web applications. Our application makes an HTTP call to ChatGPT using a provided API key to retrieve an answer from the bot, which is then displayed in the application view.

We provide technical details about the implementation of the proposed mobile application, outline the advantages and drawbacks of the technique, present examples from recent research on ChatGPT, and highlight current research trends to help readers delve into the matter.

2. Background

The natural language understanding problem is quite complex. Consider for example a communication either via text or as speech. The understanding of what is being conversed starts with the inference about the context of the messages from and to followed by an effective representation of knowledge and expectations of the domain and reason of the interaction. Adding to the difficulty of natural language systems to interpret messages are irony, sarcasm, ambiguity, colloquialism, slang, errors in text and speech, low-resources languages, etc.

Any natural language processing system is composed of the text itself, tokenization, sentence analysis, extraction, merging, and template generation. Tokenization establishes whether the word is an article, a preposition, a verb, an adverb, etc. Sentence analysis sets the word as an object or a subject. Extraction phase is the first entirely domain-specific stage of the process. During extraction, the system identifies specific relations among relevant components of the text. The merging phase must address issues such as synonym reference and anaphora resolution. Finally the template generation phase determines the number of distinct relationships in the text, map these extracted pieces of information onto each field of the template, and produces the final output template.

One application of natural language is a chatbot which is computer program that simulates a conversation oral or in text between a person and a digital equipment. Chatbots are relatively easy to implement, as they are often based on simple rule-based decision trees and they are typically pre-programmed with a limited set of responses. There is an increasing trend of 24% from 2023 to 2030 [2] on businesses using chatbots for customer support. Chatbots can help your brand scale 1:1 communication, but agents bring empathy to the table. Some inquiries require a human touch that only live agents can provide—meaning chatbots work best when they are used in tandem with humans.

In contrast, ChatGPT is more sophisticated than a common chatbot. It is an AI (Artificial Intelligence) bot capable of generating responses based on the context and tone of the conversation. ChatGPT contains 175 billion parameters, making it one of the largest and most powerful models for AI processing available today—hence its growing use in different occupations.

It is being integrated with other technologies, such as voice assistants and virtual and augmented reality platforms to be used as a personal assistant in varied fields such as education, health, agriculture, finance, transportation, etc.

ChatGPT makes use of user's feedback to train its language model to better align the bot's output with the user's intent. Reinforcement Learning from Human Feedback (RLHF) is described in depth in OpenAI's 2022 paper titled "Training language models to follow instructions with human feedback" [1]. This concept of user's feedback in a language model dates back to 2017 when Google introduced transformers, a tool for Large Language Models (LLMs), which is basically a neural network that learns context and assigns meaning to sequence of data such as words in a phrase. The model uses mathematical techniques that resembles the functioning of a parallel processing engine which means it runs fast. A year later, OpenAI introduced Generative Pre-Trained Transformer version 1 (GPT-1), the following year they upgraded it to GPT-2, then in 2020 GPT-3, and ChatGPT at the end of 2022. At present, 2023, GPT-4 is only accessible to those who have access to ChatGPT Plus, a premium service from OpenAI for which users have to pay \$20. Like all the good things in life, to access the impressive features of GPT-4, one needs to pay the price.

More and more AI chatbots [4] are being developed such as Google's Bard, Microsoft's Bing AI, Amazon's AI Model, Perplexity AI, etc. Most of these chatbots are built on a modern, research-based large language model (LLM), which is an in-depth knowledge of reliable information sources and user's feedback. However, some others AI chatbots launching are being held back over fears its AI tool would help spread false, biased or misleading information

ChatGPT made its debut in the scientific literature through published papers and preprints [4]. Although ChatGPT can undoubtedly benefit users of all backgrounds, it has sparked an upheaval in the scientific community and ignited debates around the ethics of using AI to write scientific publications that can influence the decisions of physicians, researchers, and policymakers.

As point out by OpenAI [5], the data which ChatGPT uses provide information from 2021 and earlier. The chatbot does not currently consider information reported in 2022 onward. 2 For a field that is driven by recent advances to boost knowledge, enhance interventions, and formulate evidence-based policies, this year-long (and growing) information gap is a stark hindrance. If scholars use ChatGPT to create content, attempting to publish papers that contain false or outdated information will tarnish authors' credibility among colleagues and peers.

For the time being, users might be able to use well-developed AI tools to increase work efficiency for tasks such as proofreading and manuscript checks [6].

3. Project Goals

ChatGPT was originally a web application, and only recently, May 2023, OpenAI launched the mobile version, which synchronizes user chat history with its web counterpart and features voice input supported by OpenAI's speech recognition model, Whisper.

In this paper, we have developed a platform-independent mobile application that facilitates communication between the user and ChatGPT. The essential point to note is that the bot retains user-entered queries, creating an exchange that resembles a normal conversation between two individuals. Consequently, our application must provide a scrolling window where the exchanges are displayed on the screen and can be scrolled back by the user to review previous interactions. Being platform-independent adds an advantage to our application, allowing it to run on any OS mobile device. As a point of self-praise, our application was proposed well before OpenAI's secret launch of its mobile version of ChatGPT.

At present, we cannot fully compare the performance of our AI chatbot application with ChatGPT since our application primarily serves as a shell for ChatGPT, and its performance depends on ChatGPT's performance. Some reviewers have noted that certain AI bots provide in-text links to sources throughout their answers, along with a source breakdown at the end of each response, which is not a reliable feature in ChatGPT [7].

The primary goal is to develop a mobile tool as an additional option that allows users to quickly access data-informed recommendations and instructions on various aspects of their lives. This basic intelligent mobile system can be included in an introductory course on the development of intelligent mobile applications.

4. Methodology

The platform-free mobile application is developed with Google's Flutter an open source SDK (Software Development Kit) with an editor, a compiler, code libraries, analytical tools, and documentation. The easiest way to use Flutter is to install it as a plugin (add-on) to an IDE (Integrated Development Environment) such as Google's Android Studio.

Flutter uses Google's Dart programming language. Dart is similar to Java and Javascript. So you build an application that runs pretty much on any mobile device plus the web and you use only one code. One codebase for various platforms and devices means more development efficiency, easier maintenance, and smoother onboarding of new developers.

The core of any flutter application are widgets which are not simple UI (User Interface) elements like buttons and text boxes. They include complex widgets like scrolling lists, navigations, sliders, and many others. These widgets help to build simple to complex UIs, save you time and let you focus on the business logic of your application. Flutter offers a widget catalog of 14 elements: Accessibility, animation & Motion, assets & images & icons, async, basics, cupertino, input, interaction models, layout, material components, painting & effects, scrolling, styling, and text. Flutter widgets are designed so developers can easily customize them. Flutter achieves this through a composition approach, sort of parent-child properties' inheritance. This means most widgets are made up of smaller widgets, and the most basic widgets have specific purposes.

Before the coding begins we need to do two things: first, add the http dependency to the configuration file "pubspec.yaml":

```
name: chatgpt
description: A new Flutter project.
publish_to: 'none'
version: 1.0.0+1
environment:
  sdk: '>=3.1.0-163.0.dev <4.0.0'
```

Dependencies specify other packages that your package needs in order to work.
dependencies:

```
http: ^0.13.5
flutter:
  sdk: flutter
cupertino_icons: ^1.0.2
dev_dependencies:
  flutter_test:
    sdk: flutter .....
```

and then add Internet permission to "AndroidManifest.xml" located at projectname/android/app/src/main:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <uses-permission android:name="android.permission.INTERNET" />
  <application ..... >..... </application></manifest>
```

Now we can start the coding by importing the necessary libraries: material.dart (considered as one of the top widgets on the widget tree), and http.dart (allowing to make a connection to a website's API (Application Programming Interface)). Next we have the main() - any good Java-like language requires a main class. We now have to decide whether our application is just for browsing and displaying content (stateless) or for user's interaction (stateful).

Our application, stateful, provides a Textfield widget where the user enters a text (message) and after either pressing the carriage return or hitting an arrow-button, the message is passed to ChatGPT through the use of the http protocol which allows communication between a client (the user) and a server (ChatGPT's server: api.openai.com). The response we get from ChatGPT is then displayed on the device's screen inside of a ListView widget.

The code contains the build() method that represents the screen layout (schematic) shown below in Figure 1, and the screen of an iOS emulator is shown in figure 2.

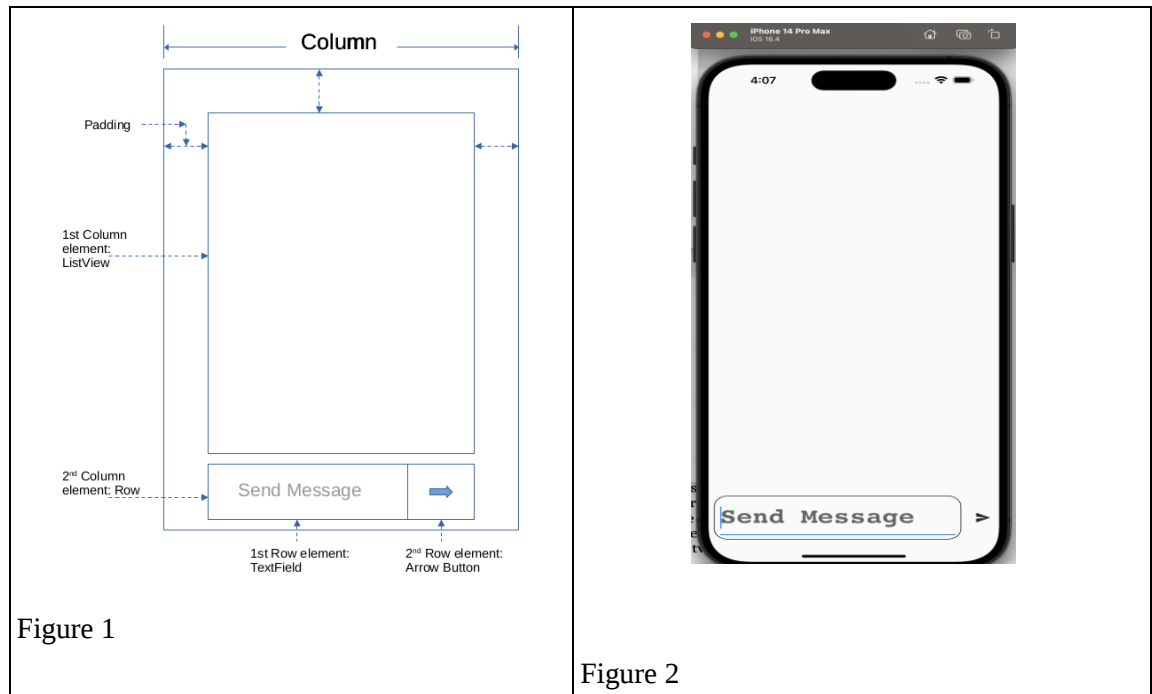


Figure 1

Figure 2

The crux of the project: the function that receives the user's message and pass it to ChatGPT using HTTP and ChatGPT's API. The user must apply to ChatGPT to get a personal key code. Here is the function we called chatGPTResponse:

```
Future<String> chatGPTResponse(String message) async {
  final apiKey = '.....'; // Replace the dots with your OpenAI API key
  final url = Uri.https('api.openai.com', '/v1/completions');
  final response = await http.post( url,
    headers: {'Content-Type': 'application/json', 'Authorization': 'Bearer $apiKey', },
    body: json.encode({
      'model': 'text-davinci-003',
      'prompt': message,
      'max_tokens': 2000,
      'top_p': 1,
      'frequency_penalty': 0.0,
      'presence_penalty': 0.0}), );
  if (response.statusCode == 200) {
    Map<String, dynamic> jsonResponse = jsonDecode(response.body);
    final chatGPTResponse = jsonResponse['choices'][0]['text'];
    return chatGPTResponse; } else {throw Exception('No ChatGPTResponse.')} } }
```

5. Results

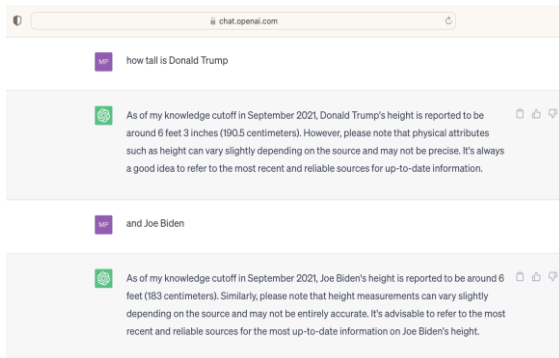
The responses from ChatGPT are shown in the scrollable ListView. The table below shows the initial screen for an Android device and an iOS device proving that the code is good for both mobile platforms.



The next table shows message exchange in both devices (Android and iOS)



There are two points of discord between the mobile application and the web application: one - context is not being included in the mobile chatbot response while the context is recognized in the web application as shown on the table below; and two – the responses are different, the one in the mobile is different from the one in the web application. The table below shows the responses: mobile and web.



6. Conclusion

The project involves porting a popular web AI-based chatbot application to a mobile application that can run on any existing platform. After obtaining an API code from OpenAI, we embarked on building the mobile chatbot application, which initially had no counterparts in the market.

We chose Google's Android Studio as our Integrated Development Environment (IDE) to aid in the development of the mobile application. Additionally, we integrated Flutter as a plugin into Android Studio. Flutter is an open-source User Interface (UI) toolkit developed by Google for creating natively compiled applications for mobile, web, and desktop platforms from a single codebase. It empowers developers to build high-performance, visually appealing, and cross-platform applications using a single programming language, Dart. We incorporated dependencies and libraries, and subsequently, we constructed our UI. HTTP is the communication protocol employed to access ChatGPT responses to users' messages.

Overall, the application functions as expected for a typical chatbot, providing correct answers for both mobile and web applications. However, there is a difference in the response wording between the mobile and web applications. Additionally, our mobile application does not exhibit any signs of context processing, whereas the ChatGPT iOS mobile application, launched at the end of May 2023, demonstrates context inclusion in the interaction between the user and the chatbot. The only plausible explanation is that mobile applications, apart from the one from OpenAI, do not incorporate natural language context processing.

References

- [1] <https://arxiv.org/pdf/2203.02155.pdf>
- [2] <https://www.grandviewresearch.com/industry-analysis/chatbot-market>
- [3] <https://www.techopedia.com/who-are-the-competitors-of-chatgpt>
- [4] Stokel Walker C., "ChatGPT listed as author on research papers: many scientists disapprove", Nature. 2023;613(7945):620-621. 10.1038/d41586-023-00107-z
- [5] OpenAI. ChatGPT general FAQ. 2023. Accessed February 10, 2023. <https://help.openai.com/en/articles/6783457-chatgpt-general-faq>
- [6] Thorp H. H., "ChatGPT is fun, but not an author", Science. 2023;379(6630):313. 10.1126/science.adg7879
- [7] The AI writing on the wall. Nat Mach. 2023;5(1):1. 10.1038/s42256-023-00613-9; <https://teche.mq.edu.au/2023/02/why-does-chatgpt-generate-fake-references/>