

# Laugh Out Loud: Developing a Joke Android App using OpenAI GPT

Sushma Yadav Duvvi

Department of Computer Science and Information Systems  
Texas A&M University - Commerce  
Commerce, Texas, USA  
sduvvi@leomail.tamuc.edu

## ABSTRACT

Android devices have become ubiquitous today and have revolutionized the way we interact with technology. With a large and growing user base, the demand for Android applications has never been higher. As a result, there has been a significant increase in the number of developers creating apps for the Android platform. In this project, we are aiming to leverage the capabilities of the Android platform to develop a daily joke application using OpenAI GPT API. The app offers users the ability to request a new joke from one of five categories (at least): dogs, jobs, kids, cars, and others. The app will utilize the OpenAI GPT API to generate jokes for each category, which will be then displayed to the user within the Android Studio interface or a physical Android Mobile device. Further a group of individuals will be evaluating the jokes and rank them from 1 to 5, with 5 being a very good joke and 1 being not a good joke. The findings will show that OpenAI GPT is proficient at generating jokes in the certain category but has some room for improvement in the other categories. This project highlights the potential of combining Android Studio and OpenAI GPT API for developing humor-based applications.

## 1 INTRODUCTION

Humor is an important aspect of human communication that can build social bonds, reduce stress, and entertain. With the advancement of mobile applications and machine learning technologies such as natural language processing (NLP), it has become possible to develop more sophisticated humor-based apps that can generate creative and contextually appropriate content in response to user input. The OpenAI GPT API is one such model that has shown potential in generating high-quality and engaging responses. However, using GPT API for joke generation comes with challenges such as fine-tuning the model for the specific application, the need for large datasets, and potential biases in the generated content.

Despite the challenges, using machine learning-based approaches to generate humor content offers numerous benefits such as adaptability to user preferences, generating content on demand, and personalized user experiences. With the increasing demand for novel and engaging mobile apps, the development of a humor-based app that leverages machine learning technologies presents new opportunities for social interaction, entertainment, and creativity. Our project aims to create a joke Android app that

integrates machine learning with humor to deliver a unique and engaging user experience that enhances social interaction while providing humor-based entertainment. This project has the potential to contribute to the growing field of computational humor and provide a new form of mobile-based entertainment for users[1].

## 2 LITERATURE REVIEW

Humor has always been a vital aspect of human life, but its importance has grown even more pronounced in the aftermath of the COVID-19 pandemic. By providing a mental escape, fostering community and connection, and promoting physical and emotional health, humor has become an essential tool for navigating the challenges of the modern world. Humor has been found to have various benefits, such as improving mood, strengthening relationships, enhancing creativity, reducing stress, and promoting resilience. In the context of joke apps, humor is a key driver of user engagement and retention[4].

### 2.1 OVERVIEW OF HUMOR-BASED ANDROID APPS

Humor-based Android apps have become increasingly popular in recent years. These apps are designed to provide entertainment to users with jokes, funny images, and humorous videos. One of the main types of humor-based Android apps is the joke app, which provides users with a wide range of jokes and puns to share with their friends and family. Other humor-based apps include funny image and video apps, which provide users with a variety of humorous pictures and videos to enjoy and share. These apps have become very popular among users of all ages, as they provide a quick and easy way to add humor and laughter to their daily lives[1].

While humor-based Android apps have become very popular, developing such apps can be a challenge. One of the main challenges is developing content that is truly funny and engaging for users. This requires a deep understanding of the type of humor that will appeal to the app's target audience, as well as the ability to create content that is both creative and humorous. In addition, app developers must consider factors such as user engagement, user retention, and monetization strategies to ensure the long-term success of their apps. However, with the help of advanced AI models such as OpenAI's GPT, developers can now create humor-based Android apps that are more engaging and entertaining than ever before[3].

## 2.2 PREVIOUS JOKE GENERATION MODELS

In the early days of joke generation in mobile applications, developers primarily relied on rule-based systems and Markov chains to generate jokes. Rule-based systems are software programs that generate output based on a set of predefined rules, while Markov chains are mathematical models that use probability theory to generate output based on a given input. While these systems could generate jokes, the quality and novelty of the output were often limited, as the algorithms were not able to generate truly creative or unexpected jokes.

Another approach used in joke generation was the use of pre-written jokes or scripts, which were often manually curated and added to the app. While this approach allowed for higher quality and more engaging content, it was time-consuming and limited the app's ability to generate new and original content on a regular basis.

With the development of advanced machine learning algorithms such as OpenAI's GPT, developers now have access to more powerful tools for joke generation in mobile applications. These algorithms can generate more creative and engaging content that is tailored to the app's target audience, making them a valuable tool for developers looking to create high-quality humor-based Android apps.

## 2.3 OpenAI GPT API

The OpenAI GPT (Generative Pre-trained Transformer) API is a machine learning platform that has revolutionized the field of natural language processing (NLP) and artificial intelligence (AI). It is one of the most advanced language models available, capable of generating highly sophisticated and contextually relevant language. Developed by OpenAI, a research laboratory dedicated to advancing AI in a safe and beneficial manner, the GPT API is widely used by developers, researchers, and businesses to build innovative AI applications. In this paper, we focus on the use of the OpenAI GPT API for developing a humor-based Android application that generates jokes. We explore the advantages of using GPT for generating humorous content and discuss how it compares to older models used in joke generation, such as rule-based systems and Markov chains. We also discuss the challenges and limitations of using GPT in joke generation, and how these can be addressed to build a high-quality, engaging humor-based Android app[5].

## 2.4 USE OF OpenAI GPT API IN ANDROID APPS

The OpenAI GPT API has been used in a variety of AI-based Android apps, ranging from language translation to chatbots and text summarization. Its advanced natural language processing capabilities have made it an ideal tool for developing humor-based Android apps that generate jokes. Using the GPT API, developers can train their models on large datasets of text to generate highly contextual and relevant jokes that can keep users engaged and entertained. Additionally, the API's flexibility allows developers to fine-tune their models for specific use cases, ensuring that the generated jokes match the intended tone and style of the app. The

use of GPT in Android apps has shown great potential for improving the quality and interactivity of the user experience, and it is expected to become an increasingly popular tool in the development of AI-based Android apps in the coming years[2].

## 2.5 CHALLENGES OF GPT FOR JOKE GENERATION

While the OpenAI GPT API offers many advantages for joke generation in Android apps, there are also some challenges that developers need to be aware of. One of the primary challenges is the size of the model, which can be quite large and require significant computational resources to train and run. This can be a limiting factor for developers who have limited computing power or limited access to cloud-based resources. Additionally, the accuracy and quality of the generated jokes can vary based on the quality and quantity of the training data used, as well as the specific parameters and settings of the GPT model. Finally, as with any machine learning model, the GPT API may generate inappropriate or offensive content, which can be a concern for developers who want to ensure that their apps are appropriate for all users. Addressing these challenges requires careful attention to the training data, model settings, and content filtering mechanisms, as well as ongoing monitoring and refinement of the model over time.

## 3 METHOD

### 3.1 PROBLEM IDENTIFICATION

The COVID-19 pandemic has brought about unprecedented levels of stress and anxiety worldwide. With millions of people forced to stay indoors due to social distancing measures, there has been a surge in demand for entertainment and distraction. While there are many apps that provide entertainment, there is a lack of apps that specifically address the need for humor and levity during this difficult time.

Research shows that humor can have a positive impact on mental health, reducing stress and anxiety levels. However, many people are struggling to find ways to access humor and joy in their daily lives. There is a clear need for an app that provides a steady stream of humorous content, tailored to the interests and preferences of individual users[4].

This app aims to fill this gap by providing users with a platform to access a wide range of jokes in different categories. By leveraging the capabilities of the OpenAI GPT API, the app generates humorous content that is tailored to the user's interests and preferences. The app aims to help users alleviate stress and anxiety by providing a much-needed source of humor and levity during the ongoing pandemic.

### 3.2 COMPETITIVE ANALYSIS

To better understand the market and potential competitors, we conducted a thorough analysis of existing apps and solutions that aim to address the same problem. Our research revealed that there are several apps available that offer similar functionality to our joke app, one such example is Jokester.

However, upon closer examination, we found that these apps are either limited in their joke categories, have a smaller database of jokes, or lack the ability to generate new jokes on demand. Our app aims to stand out by offering a wider range of joke categories as well as the ability to generate new jokes using the OpenAI GPT API[5]. Additionally, our app has a user-friendly interface that allows users to easily navigate and request jokes.

We believe that our app offers a unique and valuable solution to the problem of providing users with fresh and funny jokes on demand and will stand out in a crowded market of joke apps.

### 3.3 TECHNOLOGY SELECTION

Our app leverages a variety of technologies to deliver a seamless and engaging user experience. The following are the key technologies we are using in the development of our app:

1. **Android Studio:** We are using Android Studio, the official Integrated Development Environment (IDE) for Android app development, to build our app.
2. **Kotlin:** We are using Kotlin, a modern programming language that is fully supported by Android Studio, to write the code for our app[6]. Additionally, in 2017, Google announced that Kotlin would be an official Android development language, which has contributed to its popularity and widespread adoption in the Android development community. As such, we are confident in Kotlin's stability, reliability, and continued support from Google, making it an excellent choice for our app development needs[7].
3. **OpenAI GPT API:** We are utilizing the OpenAI GPT (Generative Pre-trained Transformer) API to generate jokes in real-time for our users. This state-of-the-art API uses deep learning algorithms to generate natural language responses to user prompts[5].
4. **Retrofit:** We are using Retrofit, a type-safe HTTP client for Android and Java, to handle network requests and API calls in our app.
5. **Material Design:** We are following Material Design guidelines to create a beautiful and intuitive user interface for our app[8].

By leveraging these cutting-edge technologies, we can create an app that is fast, reliable, and easy to use. Our app leverages the power of deep learning algorithms to generate jokes in real-time, making it an innovative and engaging addition to the Android app market.

### 3.4 DEVELOPMENT PROCESS

The development process of our app involves several stages, starting with research and planning, followed by design, coding, and testing. We decided to use Kotlin as the primary programming language for our Android app and integrate the OpenAI GPT API to generate jokes[9].

During the research phase, we conducted a thorough analysis of existing humor apps and identified the need for an app that can generate jokes automatically. We also surveyed potential users to

better understand their preferences and expectations from such an app. After gathering requirements, we are designing the app's interface and functionality. We decided to keep the interface simple and user-friendly, with multiple buttons to generate jokes and the option to share them on social media platforms. Next, we began coding the app, utilizing Kotlin and further integrating the OpenAI GPT API.

**3.4.1 Integration of OpenAI GPT API:** We are integrating the OpenAI GPT API into our app to generate high-quality and relevant jokes. The API uses deep learning algorithms to generate natural language responses that are both grammatically correct and contextually relevant to user prompts. The integration is done using Python programming language and the requests library to send HTTP requests to the API endpoint. The API's response will be parsed using JSON to extract the generated joke. The use of OpenAI GPT API will allow us to provide a unique and engaging joke generation experience to our users[5].

**3.4.2 User Experience Design:** To design the user experience for our joke generator app, we started by creating a wireframe to map out the basic layout and functionality of the app. We want the app to be simple and intuitive to use, so we are focusing on creating a clean and minimalist design. We opted for a multiple-page design that features a prominent "Let's Laugh Out Loud" button on the first page. Upon clicking it, users will be directed to a page with multiple category buttons, such as "Dogs," "Kids," "Cars," "Jobs," "Food," "Work," etc. On clicking a specific button, a prompt is sent to GPT API in the backend and the generated joke will be displayed in a new page [8, 10, 11].

Additionally, we will include a "Share" button to allow users to share the joke on various social media platforms. A navigation bar at the top of the page will provide access to the "About" section, which will provide details about the app and the team behind it. To ensure a seamless user experience, we will conduct multiple rounds of user testing and incorporate feedback from testers to refine and enhance the design further. Our goal is to create an app that provides users with a fun and entertaining way to generate jokes on-the-go while being user-friendly and functional.

Once the coding is complete, we will test the app thoroughly to ensure that it is functioning correctly and that there are no bugs or errors. We will also seek feedback from users to further improve the app's functionality and user experience.

Overall, the development process involves careful planning, thorough research, and effective use of technologies such as Kotlin and the OpenAI GPT API.

### 3.5 TESTING AND VALIDATION:

To ensure the quality and reliability of our joke generator app, we have planned a comprehensive testing process. Our testing plan includes both automated and manual testing methods. We will use JUnit and Espresso testing frameworks to test the app's functionality and identify any issues or bugs in the code. Additionally, we will utilize Firebase Test Lab to test the app on various devices and ensure that it works properly across different screen sizes, operating systems, and hardware configurations. In

addition to automated testing, we will also conduct manual testing to evaluate the user experience and usability of the app. We will request a group of users to provide feedback on the overall experience and to evaluate different design and functionality options. We are committed to fixing any issues or bugs identified during the testing process to ensure that our app provides a seamless and enjoyable user experience [9, 11].

Overall, our testing and validation process will help us to ensure that our app is functioning correctly and meeting the needs of our users. We will continue to monitor the app and make improvements as necessary to ensure that it remains reliable and effective.

### 3.6 MAINTENANCE AND UPDATES

We are developing a comprehensive plan to maintain and update our app, which includes regular monitoring, bug fixing, updating content, incorporating user feedback, and planning for future enhancements. We will closely monitor the app's performance and user feedback to quickly identify and address any issues or bugs. We will continuously update the app's content to keep it fresh and engaging, and actively seek out and incorporate user feedback to ensure that we are meeting their needs. Additionally, we will stay ahead of the curve by planning for future enhancements based on industry trends and user needs. By following this plan, we are committed to providing our users with the best possible experience.

### 3.7 FUTURE DEVELOPMENT

In the future, we are planning to further improve the app's functionality by incorporating more advanced machine learning algorithms and natural language processing techniques. This will allow the app to generate even more sophisticated and nuanced jokes, and potentially even adapt to individual users' preferences over time. We also plan to expand the app's user base through targeted marketing and outreach, and to incorporate user feedback and suggestions into future updates. Overall, our goal is to continue enhancing the user experience and providing a fun and engaging platform for joke lovers everywhere.

## REFERENCES

- [1] Md Romael Haque and Sabirat Rubya. 2022. "For an App Supposed to Make Its Users Feel Better, It Sure is a Joke" - An Analysis of User Reviews of Mobile Mental Health Applications. *Proc. ACM Hum.-Comput. Interact.* 6, CSCW2 (November 2022), 421:1-421:29. DOI:<https://doi.org/10.1145/3555146>
- [2] Oğuzhan Katar, Dilek Ozkan, GPT, Özal Yildirim, and U Rajendra Acharya. 2022. *Evaluation of GPT-3 AI language model in research paper writing*. DOI:<https://doi.org/10.13140/RG.2.2.11949.15844>
- [3] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners.
- [4] humor\_in\_medicine.pdf. Retrieved February 17, 2023 from [https://www.utmb.edu/gem/pdfs/humor\\_in\\_medicine.pdf](https://www.utmb.edu/gem/pdfs/humor_in_medicine.pdf)
- [5] OpenAI API. Retrieved February 17, 2023 from <https://platform.openai.com>
- [6] Kotlin for Android | Kotlin. *Kotlin Help*. Retrieved March 10, 2023 from <https://kotlinlang.org/docs/android-overview.html>
- [7] Android Announces Support for Kotlin. *Android Developers Blog*. Retrieved March 10, 2023 from <https://android-developers.googleblog.com/2017/05/android-announces-support-for-kotlin.html>
- [8] Material Design for Android. *Android Developers*. Retrieved March 10, 2023 from <https://developer.android.com/develop/ui/views/theming/look-and-feel>
- [9] The activity lifecycle | Android Developers. Retrieved March 10, 2023 from <https://developer.android.com/guide/components/activities/activity-lifecycle>
- [10] Add buttons to your app | Android Developers. Retrieved March 10, 2023 from <https://developer.android.com/develop/ui/views/components/button>
- [11] Design for Android. *Android Developers*. Retrieved March 10, 2023 from <https://developer.android.com/design>