COMP5313WA 2024 Chatbot Contest

EXECUTIVE SUMMARY of Group No. 6

Students Names:
1226847 Sree Krishna, Chandana
1221643 Swain, Gurdeep
1219988 Tabrez, Shams
1210987 Tayo, Adeoye
1228363 Thakkar, Bhargav Hemantkumar
1094117 Vyas, Dron Sushrutkumar
1218119 Warnakulasuriya, Nimasha
1229740 Divyam, Patel

Overview - The Quick Pitch

We compare each model in a popular field known as the Internet of Things (IoT). The idea is to see how well, and gauge two popular large language models (LLM) can hold a meaningful conversation with a User and the ability to answer intelligent questions. IoT has been around for quite some time, and there are a lot of questions that users need answers to. We have a lot of resources on the internet to guide users on how to make the best of this emerging technology, but it gets overwhelming to read a lot of resources and productive time gets wasted. How about having a model that can help hit the nail on the head in seconds?

The Problem

Few online tools like this exist for new users who want complex information explained in layman's words. Beginners and seasoned engineers alike will utilize our services to find rapid answers to important queries. A beginner needs to learn about standard IoT components like Raspberry Pi and PIR motion sensors, but most resources available have a tad too advanced documentation. On the other hand, an experienced user requires more complex documentation about integrating Azure SignalR with Azure's IoT Hub services, which is information unique to Azure.

a. Absence of Human Interaction: A genuine teacher or tutor must provide human interaction different from ChatGPT and other generative models. This lack of human interaction can be detrimental for certain students, who could gain more from having a more intimate relationship with their teacher. According to a 2014 study by D'Mello and colleagues, students who interacted with a virtual tutor who imitated affective behaviour like a human had a better learning outcome than those who connected with a virtual tutor who did not.
b. Limited Understanding: The concepts that generative models assist students in learning still need to be fully understood, as their training is based on statistical patterns in the data.
c. Dependency on Data: Generative models are trained on a large amount of data, and the quality of the model is highly dependent on the quality and quantity of the data. If the data is insufficient or irrelevant, the model will not be able to perform as well. A study by Kocaguneli and colleagues (2019) showed that a generative model-based question-answering system performed poorly when the training data was irrelevant to the task.

Current IoT-Related Conversations in the Market

We currently have some applications that focus on IoT-related conversations, and we can review a few.

- **Adafruit IoT monthly**: The Adafruit IoT monthly is a service provided by Adafruit to educate new or experienced users about the Internet of Things. Adafruit is a popular company that is into IoT device

production. Adafruit IoT monthly has a chatbot interface that teaches its users about Adafruit and how to build IoT-related services using Adafruit products.

Advantages:
Support: Adafruit heavily supports its product and provides up-to-date solutions through conversational chatbots.

Disadvantages:
Cost: This service can be very costly for users seeking to gain knowledge about IoT or using Adafruit services.

Direct native projects: Using Adafruit IoT monthly services feels like gaining knowledge directly from a professor in a university. This helps users to build projects using Adafruit tools.

Continuity and reliability: Users who subscribe to this service can get long-term service support due to the use of their IoT components and high or premium-quality tools.

Method of payment: Adafruit IoT Monthly puts the 'monthly' in the name of practice as it requires a reoccurring fee from its users to people who might need a one-time question-answering session.

- **ThingSpeak IoT Talk**: This is a platform that ThingSpeak builds to promote the use of natural language commands to interact with IoT devices. We consider this conversational because the concept behind a conversation is an input and output between two parties to reach a goal. ThingSpeak is an IoT analytics platform that allows users to visualize, stream, collect, and analyze data in the cloud.

  Advantages:
  Integration: ThingSpeak allows its users to use the ThingSpeak IoT Talk platform by seamless integration using natural language commands. They are heavily backing it.

  Disadvantages:
  Heavy dependency: ThingSpeak IoT Talk heavily depends on the ThingSpeak platform, bringing many limitations.

  Ease of use: One sales point for ThingSpeak is using natural language commands to interact with the IoT devices.

  Requirements to learning: Using this platform requires the user to get used to some commands that will be needed to trigger some actions on the IoT devices.

The Solution

Our design is to have an application to leverage the model's API for use. These models are LLMs and are already trained on many resources. We can finetune the model to a specific domain and strictly train it in specific conversational contexts, but this will require a lot of datasets, GPU power, and time, to mention a few. Since these bots are pre-trained, we will have the basis of our comparison on the current knowledgebase of each model.

The center of our research is to see how well each model will hold meaningful conversations relating to the Internet of Things cutting across different domains in IoT and how well each model can help its users.

The learning requirements of IoT go beyond question-answering in a text-based approach, and we are going a step further to compare these bots based on their ability to read images within the context of the conversation.
This is important to users who are new to the world of IoT and would like to gain deeper knowledge about a device without knowing its name.

Our tests and focus are to compare the performance of these bots when it comes to dealing with images and staying within the context of the conversation with or without any intentional distraction from the user.
Looking at the larger scale, we can push this further to support voice translation, where the users can express themselves rather than plain text, which may be lengthy. In this sense, we can leverage on speech to text APIs and continue the conversation.

Highlights on the Chatbots Capabilities

1. Architecture:

ChatGPT is a model based on Transformers which is a model that is a solid backing when it comes to Natural Language Processing tasks. ChatGPT is a variant of the GPT family built on the GPT-2 model. ChatGPT was created with version 3, so we can call it ChatGPT-3. Our focus is on ChatGPT-4 which is a higher version than 3. We are going for version 4 because it works with images. Our research centers on how well ChatGPT can work with images within a conversation and keep to context, so we must use a model that supports it.

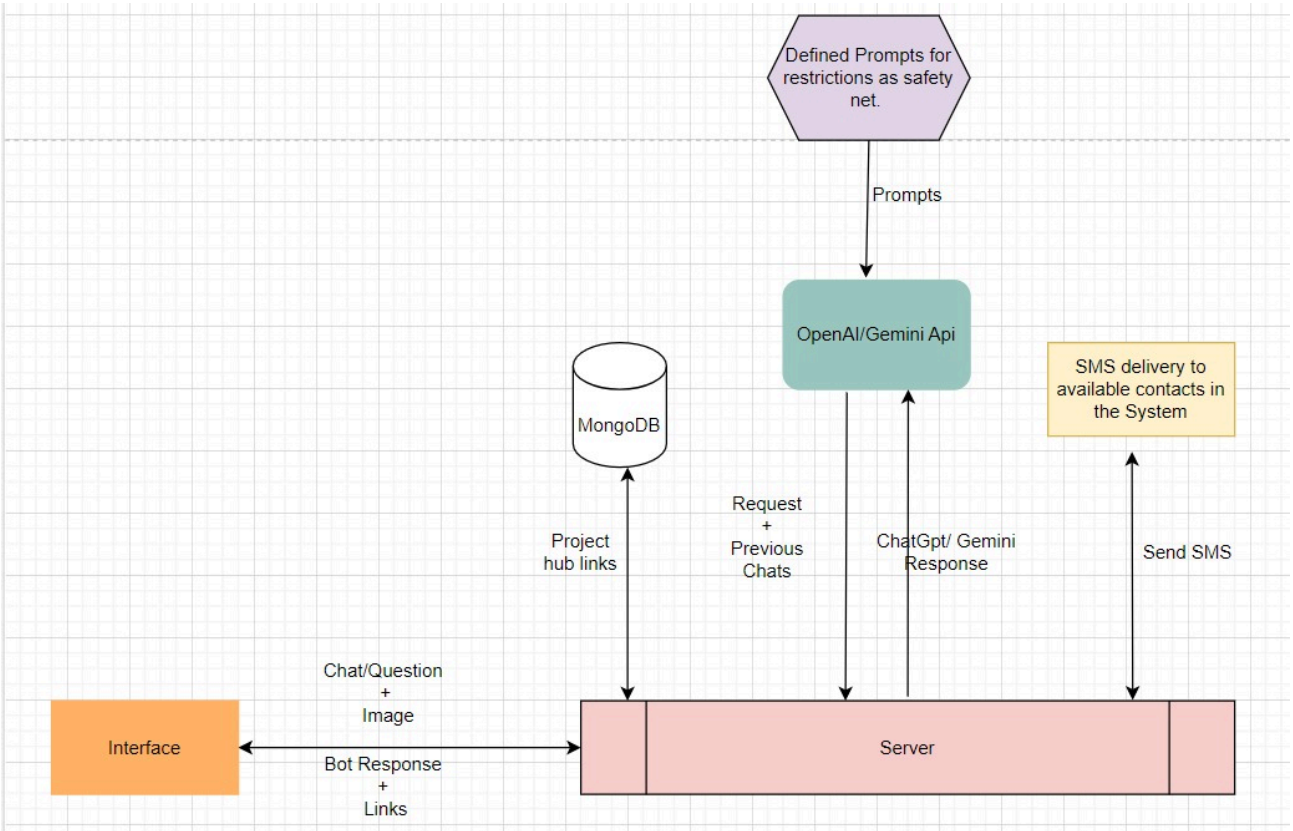

Fig1.1 Data Flow

Gemini, on the other hand, is based on a Mixture of Experts architecture. This multimodal model was built for supporting tasks that cut across various domains, hence, the name multimodal.

2. Features:

a. ChatGPT:
   i. It is heavily backed by the Transformers library in Python.
   ii. It can chat in natural language, mimicking humans.
   iii. It can generate output in different text-based formats ranging from chats, auto-completion, and code generation to original ideas from multiple inputs.
   iv. It can learn a pattern within a conversation and stick to it within that conversation.
   v. It provides an API for integration with other applications so developers can integrate with it and build applications on them.

b. Gemini:
   i. It is based on various models and can be used beyond conversations.
   ii. It can generate output in different text-based formats ranging from chats, auto-completion, and code generation to original ideas from multiple inputs.
   iii. It provides an API for integration with other applications so developers can integrate with it and build applications on it.

3. Model:

a. ChatGPT: The model for this comparison is the ChatGPT-4
b. Gemini: The model for this comparison is the Gemini Pro

Walkthrough the Gemini Chatbot Settings an Example of Use

Our variant for the Gemini chatbot is the Gemini Pro model. We are incorporating some mechanisms, like a safety net, to guide the chatbot in the background. This safety net is essential to keep the chatbot in check and to stay within the domain it is built for. A major feature in our application is to interact with Gemini using images, and we will use the in-built based64 library in Python. Our scenario is to build a functional chatbot that can guide various ranges of mindsets on resources and provide guidance from start to finish on concerns revolving around the Internet of Things. Our prototype is simple, and a high-level overview is an application that will run on the web.

Walkthrough the ChatGPT Chatbot Settings an Example of Use

Our variant for the ChatGPT chatbot is the ChatGPT-4 model. Just like the Gemini model, we are also adding some mechanisms to form a safety net always to keep the chatbot in check and sync all the time. The sample application is the same for the Gemini model on producing a bot that can answer intelligent questions, also with images, on concerns surrounding the Internet of things for a large range of users, from newbies to professionals. Our development and testing approach remains the same as the Gemini Pro, which runs on the web.

Code Explanation:

a. ChatGPT

- Library imports & KEYS:

  gradio: Gradio is a Python library that creates customizable UI components for machine learning models and other tasks.

  OS, time, base64, Vobject, requests: Standard Python libraries for operating system functionalities, time-related operations, base64 encoding, handling vCard files, and making HTTP requests, respectively.

  OpenAI: The Python package for accessing OpenAI's APIs, used here specifically for interacting with the GPT-4 model.

  Twilio: The Python package for interacting with Twilio's APIs is used here to send SMS messages.

```
import gradio as gr
import os, time, base64, vobject, requests
from openai import OpenAI
from twilio.rest import Client

client = OpenAI(api_key=os.environ.get('OPENAI_KEY'));
account_sid = os.environ.get('TWILIO_ACCOUNT_SID');
auth_token = os.environ.get('TWILIO_AUTH_TOKEN');
twilio_number = os.environ.get('TWILIO_NUMBER');
```

Fig1.2 Library Imports for GPT

```
gradio==4.22.0
openai==1.2.4
vobject==0.9.7
twilio==9.0.3
```

Fig1.3 Requirements for Chatgpt bot

- Bot Class: The Bot class encapsulates various methods and functionalities related to the chatbot:
  Downloading Files: Download files from a given URL.
  Reading VCF Files: Reads vCard files (commonly used for storing contact information) and extracts contact details.
  Sending SMS: Utilizes Twilio's API to send SMS messages.
  Building Conversation: Constructs conversation data in a structured format.
  Image Conversion: Converts image files to base64 format.
  User Interaction Handling: Methods for processing user messages and generating responses.
  GPT-4 Integration: Utilizes OpenAI's GPT-4 model for generating responses to user inputs.

- Gradio Interface: The script sets up a Gradio interface for the chatbot.
  Chatbot UI: creates a chatbot interface using Gradio's Chatbot component.
  Text Input Box: provides a text input box to type messages.
  File Upload: allows users to upload files (including images).
  Send SMS Button: enables users to send the chatbot's responses as SMS messages.

- User Interaction:
  Handlers are defined for processing user inputs and generating responses:
  add_message Method: Processes user messages and generates responses using the GPT-4 model. Also handles sending SMS messages if specified.
  Bot Method: Generates responses to user messages using the GPT-4 model.
- Execution: The script runs the Gradio interface, allowing users to interact with the chatbot in real time through a web browser.

b. Gemini

- Library imports and KEYS:
  gradio is imported as gr, a library for creating web-based user interfaces for machine learning models.

Other imports include modules for handling authentication (google.oauth2.service_account), sending HTTP requests (requests), interacting with the Vertex AI generative models (vertex), processing vCard files (object), and sending SMS messages (Twilio).

```
google-cloud-aiplatform==1.45.0
google-auth==2.27.0
gradio==4.22.0
requests==2.31.0
vertexai==1.43.0
vobject==0.9.7
twilio==9.0.3
```

Fig1.4 Requirements for Gemini Bot

- Bot Class:
  The Bot class contains various methods for the chatbot functionality. Here are some key methods:
  download: Downloads a file from a URL.

  read_vcf: read contacts from a vCard file.
  send_sms: sends an SMS message using Twilio.
  send_sms_to_contacts: sends an SMS message to multiple contacts.
  image_to_base64: converts an image file to a base64 encoded string.
  seed: Initializes the conversation seed with predefined messages.
  recognize: utilizes a generative model to recognize objects in an image.
  print_like_dislike: prints the index, value, and whether it's liked.
  add_message: adds a message to the conversation history.

- Gradio Interface: gradio interface where users can interact with the chatbot. It includes components like Chatbot, MultimodalTextbox, and Button. The submit and click events trigger the methods defined in the Bot class for adding messages and sending SMS.

Execution: Like the Chatgpt bot, the Gemini bot script runs the Gradio interface, allowing users to interact with the chatbot in real-time through a web browser.

Comparison and Analytics

Our ground is between Gemini Pro and ChatGPT-4; Gemini is a better model than ChatGPT per our chosen context. We push that the Gemini Pro model is much more reliable for our use case. We do not think ChatGPT comes close to Gemini in being the best bot that can be used to act as a human. One of our evaluation metrics is to see how well the responses can act human-like, and Gemini takes the lead. We have taken some manual approaches to evaluate these bots. We have divided them into categories as so:

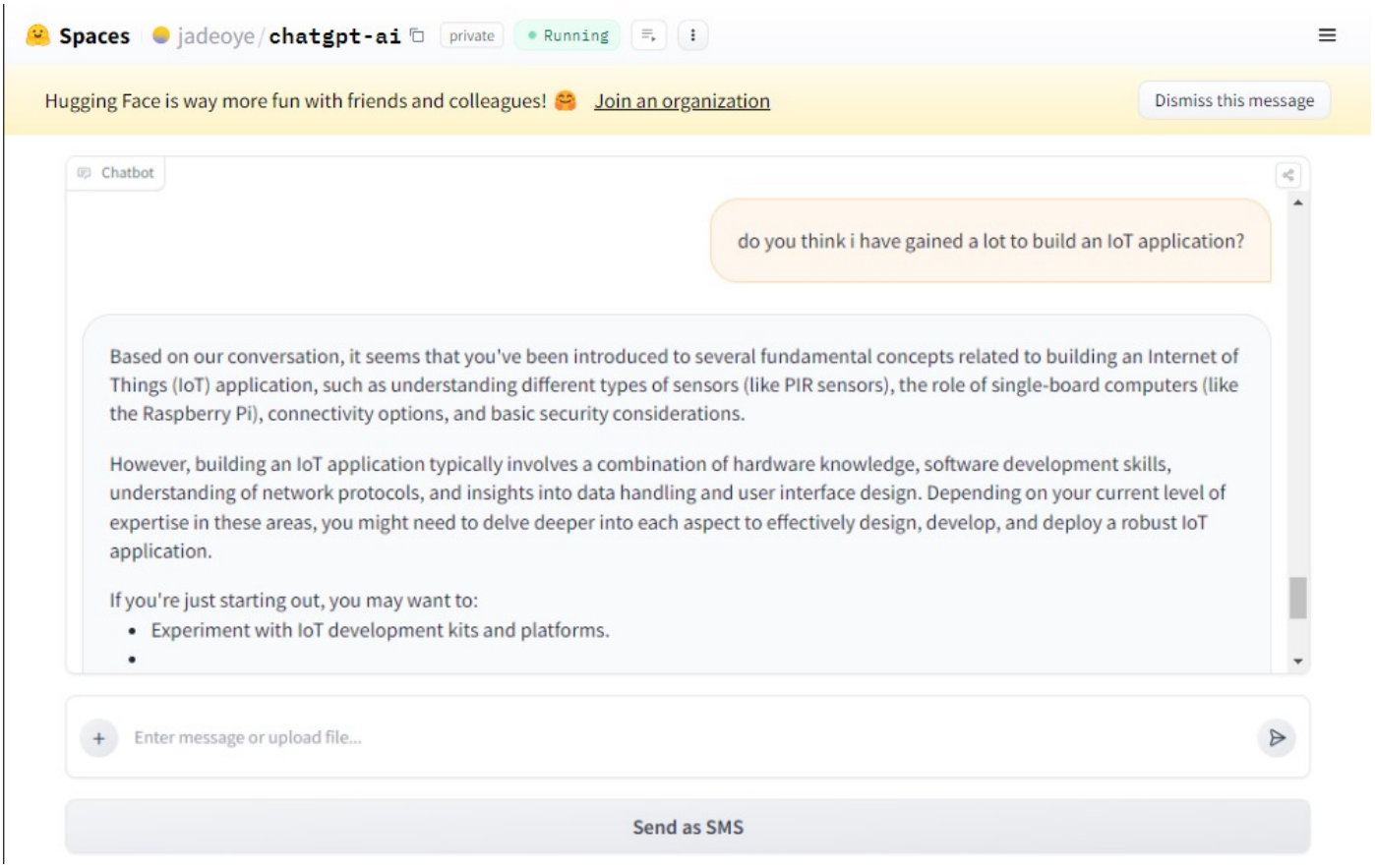

Fig1.5 conversation with ChatGPT-4

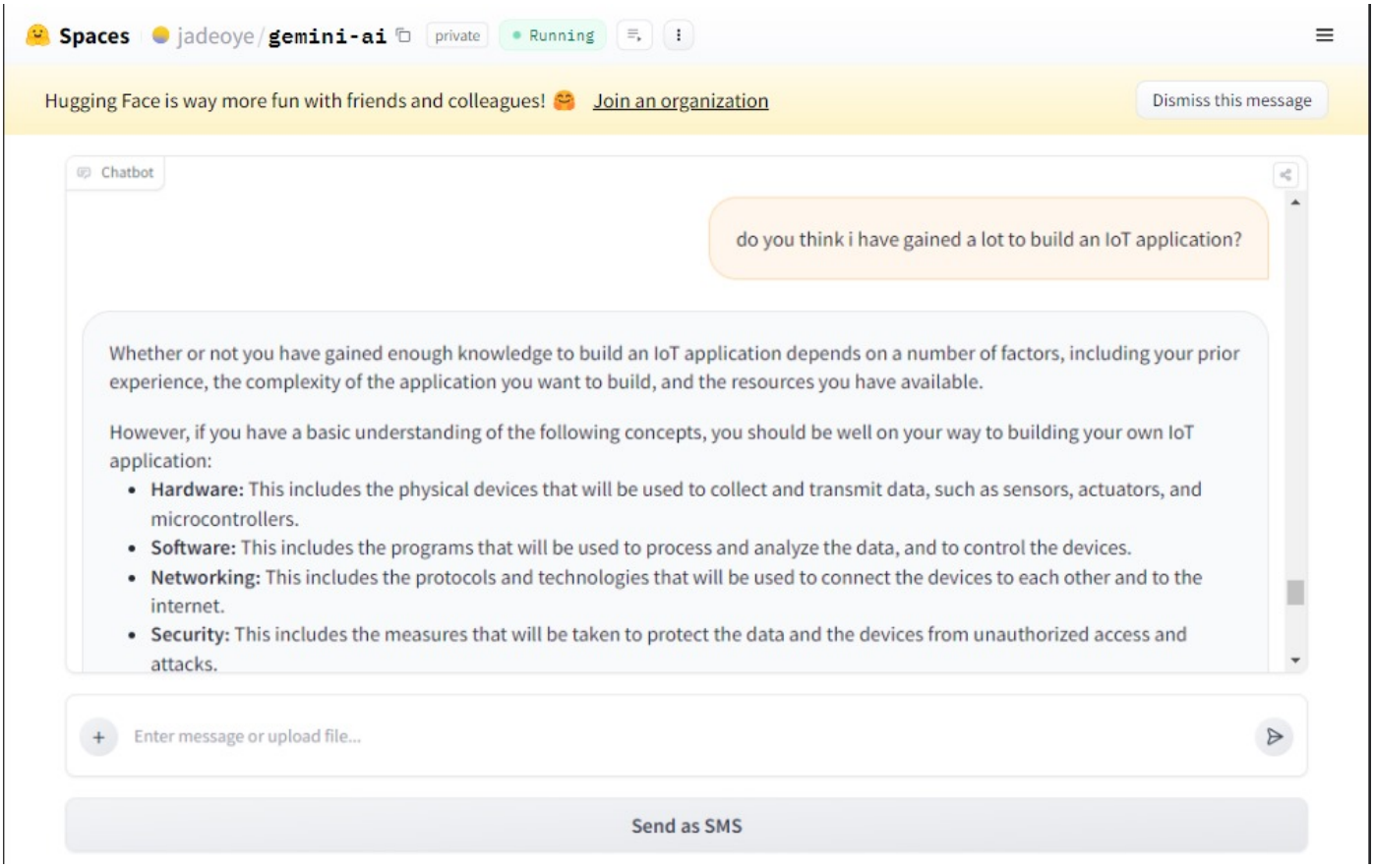

Fig1.6: Conversation with Gemini Pro

Category	Gemini Pro	ChatGPT-4	Conclusion
Image Recognition	It was able to identity the same image repeatedly with confidence	It was able to recognize the same image and if given the same image again, it says it is not sure	Gemini performs much more in recognizing images accurately with confidence
Context retention	It tried as much to stay within context of the conversation even when we distract it intentionally, but it fails sometimes.	It also fails to maintain context sometimes and it diverts a few times.	Gemini sticks to the conversation context during the chat while ChatGPT can break within the chat.
Provision of related materials for additional knowledge	It was able to provide related links where necessary during the conversation to give tips to the User	It did not provide any external resource such as links to the User	Gemini gives related links and helpful tips.
Response Length	Its responses are generally lengthy but, well detailed and explanatory	Its responses are usually short and concise but can need more for in-depth conversations	Gemini gives a more useful answer to the end user.
Time taken to get User's satisfaction	Gemini Pro took more time than usual to get to the end of the conversation due to the lengthy responses.	ChatGPT took a much shorter time to get to the end of the conversation	ChatGPT gets to the end of the conversation much faster
Relevance of response	Gemini's responses are more relevant and practically thinkable to a newbie	ChatGPT gives a more advanced response that may require an IoT-experienced user to read and understand	Gemini gives more useful response to the context of the conversation
Confidence in response	Gemini apologies even if the user falsely tells it that it has a wrong answer	ChatGPT apologies even if the user falsely tells it that it has a wrong answer	Both bots fail to stand their ground on their responses
Making inferences based on current conversation	Gemini gave an inference based on the general overview of the context of the chat, without considering the user's responses	ChatGPT made an inference based on the conversation and a summary of the chat	ChatGPT was better at concluding the conversation to give a better recommendation.
Human-like responses	Gemini gives a more human friendly response	ChatGPT responses are direct and more robot-like	Gemini makes a better human mimic.

Public Github Link: https://github.com/chandana-sree-krishna/COMP5313_CAI_Contest

References

1. G“Gemini Versus ChatGPT: Applications, Performance, Architecture, Capabilities, and Implementation.” Nitin Rane, Saurabh Choudhary & Jayesh Rane. (2023) Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4723687

2. “A Brief Analysis of “ChatGPT” – A Revolutionary Tool Designed by OpenAI Authors.” Md. Asraful Haque. (2023) Available online: https://publications.eai.eu/index.php/airo/article/view/2983

3. “Gemini technical report” https://storage.googleapis.com/deepmind media/gemini/gemini_1_report.pdf

4. “Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning”. David Baidoo-anu & Leticia Owusu Ansah. https://dergipark.org.tr/en/pub/jai/issue/77844/1337500

5. Abukmeil, M., Ferrari, S., Genovese, A., Piuri, V., & Scotti, F. (2021). A survey of unsupervised generative models for exploratory data analysis and representation learning. Acm computing surveys (csur), 54(5), 1-40. https://doi.org/10.1145/3450963.

6. Alshater, M. (2022). Exploring the role of artificial intelligence in enhancing academic performance: A case study of ChatGPT (December 26, 2022). Available at SSRN: https://ssrn.com/abstract=4312358 or http://dx.doi.org/10.2139/ssrn.4312358

7. Aydın, Ö., & Karaarslan, E. (2023). Is ChatGPT leading generative AI? What is beyond expectations? Academic Platform Journal of Engineering and Smart Systems.

8. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and individual differences, 103, 102274.

9. Pearl Mike, “The ChatGPT chatbot from OpenAI is amazing, creative, and totally wrong". Mashable, December 3, 2022. Available online: https://mashable.com/article/chatgpt-amazing-wrong.

10. “How Chat GPT utilizes the advancements in Artificial Intelligence to create a revolutionary language model”, Pegasus One, December 17, 2022. Available online: https://www.pegasusone.com/how-chat-gpt-utilizes-the-
11. advancements-in-artificial-intelligence-to-create-a-revolutionary-language-model.

12. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2023). A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology.

13. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Fedus, W. (2022). Emergent abilities of large language models. arXiv preprint arXiv:2206.07682.

14. Perera, P., & Lankathilake, M. (2023). Preparing to revolutionize education with the multi-model GenAI tool Google Gemini? A journey towards effective policy making. J. Adv. Educ. Philos, 7, 246-253.

15. Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q. L., & Tang, Y. (2023). A brief overview of ChatGPT: The history, status quo and potential future development. IEEE/CAA Journal of Automatica Sinica, 10(5), 1122-1136.

16. Team, G., Anil, R., Borgeaud, S., Wu, Y., Alayrac, J. B., Yu, J., ... & Ahn, J. (2023). Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805.

17. "Designing IoT Introductory Course for Undergraduate Students Using ChatGPT". Abdallah Al-Zoubi (2024) https://link.springer.com/chapter/10.1007/978-3-031-51979-6_

18. A comprehensive survey of the internet of things (IOT) and ai-based Smart Healthcare | IEEE Journals & Magazine | IEEE Xplore. (n.d.). https://ieeexplore.ieee.org/document/9311140/

19. team, Aic. (2024, April 2). *Chatgpt in IOT: Connecting Devices and enhancing communication*. https://aicontentfy.com/en/blog/chatgpt-in-iot-connecting-devices-and-enhancing-communication

20. Yu, S. (n.d.). *Natural interactions in IOT: Combining Mqtt and chatgpt*. www.emqx.com. https://www.emqx.com/en/blog/natural-interactions-in-iot-combining-mqtt-and-chatgpt

21. Edditter.io. (2023, December 10). *Gemini unveiled, AI in Healthcare & Iot Tracking Innovations*. LinkedIn. https://www.linkedin.com/pulse/gemini-unveiled-ai-healthcare-iot-tracking-innovations-edditter-io-04sef/?trk=organization_guest_main-feed-card_feed-article-content