AYDIN ADNAN MENDERES UNIVERSITY
ENGINEERING FACULTY
COMPUTER ENGINEERING DEPARTMENT

PRACTICHAT: CREATING AN AI LANGUAGE TUTOR

Ayşegül ÇAĞLI
Vakkas KARAKURT

Supervisor:

Asst. Prof. Fatih SOYGAZİ

2024

AYDIN ADNAN MENDERES UNIVERSITY
ENGINEERING FACULTY
COMPUTER ENGINEERING DEPARTMENT

PRACTICHAT: CREATING AN AI LANGUAGE TUTOR

Ayşegül ÇAĞLI
Vakkas KARAKURT

Supervisor:

Asst. Prof. Fatih SOYGAZİ

2024

ABSTRACT

PRACTICHAT: CREATING AN AI LANGUAGE TUTOR

Ayşegül ÇAĞLI

Vakkas KARAKURT

B.Sc. Thesis, Computer Engineering Department

Supervisor: Asst. Prof. Fatih SOYGAZİ

2024, 23 pages

This thesis delves into the development and significance of chatbots, tracing their evolution from early models like ELIZA and PARRY to modern implementations such as Siri, Google Assistant, and ChatGPT. By exploring the methodologies behind chatbot creation, particularly fine-tuning pre-trained models, this work examines the technological advancements and challenges faced in making chatbots more human-like. In this study, we developed a chatbot by fine-tuning DialoGPT, which is trained with the GPT2 architecture, using a chat dataset. We shared the training data and a sample conversation of our chatbot, which we named PractiChat. PractiChat was quite successful in responding, asking questions, and remembering the topic. Finally, we calculated the difficulty levels of the sentences written by the user and PractiChat with an algorithm we developed and printed on the interface we designed for PractiChat. In future studies, we aim to have PractiChat generate responses according to the user's level, thereby enabling the user to practice English.

Keywords: *Chatbot, Fine-tuning, Natural Language Processing, Artificial Intelligence, Transformers.*

ÖZET

PRACTICHAT: YAPAY ZEKA İNGİLİZCE ÖĞRETME ROBOTU

Ayşegül ÇAĞLI

Vakkas KARAKURT

Lisans Bitirme Tezi, Bilgisayar Mühendisliği Bölümü

Tez Danışmanı: Asst. Prof. Fatih SOYGAZİ

2024, 23 sayfa

Bu tez, sohbet robotlarının gelişimi ve önemine odaklanarak, ELIZA ve PARRY gibi erken modellerden Siri, Google Assistant ve ChatGPT gibi modern uygulamalara kadar olan evrimlerini incelemektedir. Sohbet robotu oluşturmanın metodolojilerini keşfederek, özellikle önceden eğitilmiş modellerin ince ayarının yapılmasını ele alarak, bu çalışma sohbet robotlarının daha insan benzeri hale getirilmesinde karşılaşılan teknolojik ilerlemeleri ve zorlukları inceliyor. Bu çalışmada, GPT2 mimarisi ile eğitilmiş DialoGPT modelinin ince ayarını yaparak bir sohbet robotu geliştirdik. Bir sohbet veri seti kullanarak ince ayarını yaptığımız bu sohbet robotuna PractiChat adını verdik. PractiChat, kullanıcıya cevap verme, soru sorma ve konuyu hatırlama konularında oldukça başarılı oldu. Son olarak, kullanıcı tarafından yazılan cümlelerin zorluk seviyelerini ve PractiChat'in bu cümlelere verdiği cevapların zorluk seviyelerini bizim geliştirdiğimiz bir algoritma ile hesapladık ve PractiChat için tasarladığımız arayüzde bu bilgileri görüntüledik. Gelecekteki çalışmalarda, PractiChat'in kullanıcının seviyesine göre cevaplar üretmesini sağlayarak, kullanıcının İngilizce pratiği yapmasına imkân tanımak amacındayız.

Anahtar Kelimeler: *Sohbet robotu, İnce ayar, Doğal Dil İşleme, Yapay Zekâ, Transformers.*

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
NLP	Natural Language Processing
T5	Text-to-Text Transfer Transformer
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
Token ID	Token Identification
MIT	Massachusetts Institute of Technology
A.L.I.C.E.	Artificial Linguistic Internet Computer Entity
ARPANET	Advanced Research Projects Agency Network
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
GPU	Graphics Processing Unit

1.INTRODUCTION

Technology has always been a field that evolves to meet people's needs and develops in accordance with these needs. In this rapidly developing field in recent years, chatbots are almost at the forefront. As technology advances, the need for chatbots that can converse with humans like a human has also increased. Whether you visit a bank's website, an online shopping site, or purchase an airline ticket, it is possible to see chatbots developed to assist you faster, imitating humans and helping you solve your problem, in a corner of the screen.

Chatbots are artificial programs that can interact with humans in written or spoken form using natural language. "Chat" means to converse, while "bot" means robot. So, we can also call them conversation robots. Neff et al. (2016) defined a chatbot in their study as a program that interacts through conversation, generating responses either from preprogrammed schemas or with the help of adaptive machine learning algorithms. It is possible to customize a chatbot according to your purpose, the topics you want it to respond to, and the user age group. To achieve this, it is sufficient to fine-tune models that are already trained with large datasets using a dataset that suits your needs. Finetuning is feeding the top layer of a previously trained multi-layer model with the relationships and words you want the model to learn. This method, which allows us to use the information in the lower layers and capture the relationships between concepts, is less costly than training a model from scratch. Thus, customizable chatbots serve people in many areas such as education, health, entertainment, customer service, and tourism.

Let's take a look at the history of the chatbot concept and when it entered our lives. It all started in 1950 with the question posed by mathematician, cryptanalyst, and one of the pioneers of computer science, Alan Turing: Can machines think? Turing shared his thoughts in an article titled "Computing Machinery and Intelligence" published in the journal Mind (Turing, 1950). Additionally, there is a test named after him: the Turing Test. This test measures whether a chatbot with artificial intelligence can deceive a human into thinking it is a naturally intelligent human. In the test developed by Alan Turing, a human interacts with both another human and a bot through an interface in an experimental setting. During this communication process, it is tested whether the human can distinguish between the bot and the human. For an

artificial intelligence chatbot to pass the Turing Test, it must make the human believe that the robot it is conversing with is human. This is possible through the realistic imitation of human language and behavior by the robot.

The situation of making robots resemble humans has been a major focus in the field of psychology both then and now. Let's create a thought train. Is speaking like a human synonymous with making mistakes? Should a robot make spelling errors just like a human? Should it respond to questions based on cultural and ethical principles? Which person should we use as a basis for these principles? Who is the most moral person? Is an ideal human possible? Is every person we talk to moral? Is this what helps us distinguish that we are talking to a human? Should it possess the collective memory of society? Is it possible for a robot to form its own emotions and thoughts? On what basis do humans form these? Who should be held responsible for an accident caused by a robot, the person who created the robot or the robot itself that lacks the ability to think? These are some of the questions that have been asked since those years and remain unanswered.

John Searle (1980) proposed the Chinese Room metaphor as an alternative to the Turing Test. Let's briefly discuss the Chinese Room. Imagine a man in a room who doesn't know any Chinese, and we give him texts related to Chinese symbols and grammar rules. If we give the man a text in Chinese, he can use the guide to translate his response into Chinese symbols he doesn't understand and provide us with a correct answer. As a result, the man wouldn't have learned Chinese or given a correct answer because he knows Chinese. Despite not being proficient in Chinese, this man could pass the Turing Test for understanding Chinese based on his correct answer. From this, Searle (1999) argues that just as this man doesn't need to know Chinese to pass the Turing Test, any computer program or chatbot doesn't need to know to respond correctly. As Cole (2023) mentioned in his study, from Searle's theory, we can infer that syntax and semantics are different things, and the idea that semantics must exist whenever there is syntax is incorrect.

To develop a chatbot, it is possible to train a model from scratch with sufficient hardware resources and a large enough dataset, but as we mentioned, it is extremely costly. In this study, we will first discuss the history of chatbots. Then, we will examine the architectures used in chatbot development, such as neural networks and

transformers, how they work, and their distinguishing points. Following that, we will talk about pre-trained transformer models like BERT, T5, and GPT, and which natural language processing tasks they excel in. Next, we will discuss the concept of "finetuning," which is an alternative to training a model from scratch, meaning to retrain a pre-trained model for our specific purpose. We will go through the training of a chatbot step by step. Then, we will share our experiences with the chatbot we developed by finetuning. We will assign difficulty scores to the sentences entered by the chatbot and the user using an algorithm we developed. Finally, we will discuss our ideas and how this study can be further developed.

2.LITERATURE REVIEW

If we return to the timeline, it is possible to say that the first chatbot was ELIZA, developed by Professor Joseph Weizenbaum at the MIT Artificial Intelligence Laboratory between 1964-1966 (Weizenbaum, 1966). ELIZA attempted to analyze input sentences based on keywords and generate responses accordingly. During those years, ELIZA's main concerns included recognizing keywords, capturing context, choosing appropriate transformations, generating responses when key words were absent, and providing an editing capability for ELIZA "scripts."

Another notable chatbot is PARRY, developed by American psychiatrist Kenneth Mark Colby, as introduced in the study by Colby et al. (1971). PARRY surpassed ELIZA in its capabilities by simulating responses similar to those of a patient with schizophrenia. To achieve this, adjustments were made to three emotions: fear, hostility, and mistrust, which influenced the responses generated as the conversation progressed. PARRY underwent a kind of Turing test where a group of psychiatrists engaged in conversations with both real patients diagnosed with schizophrenia and PARRY without knowing which was human and which was the chatbot. Another group of 33 psychiatrists then read transcripts of these interactions and were asked to determine which ones were human and which were chatbot. The psychiatrists correctly identified PARRY as the chatbot 48% of the time, which is close to the random guessing rate of 50% (yes or no). This illustrates the success of PARRY in simulating human-like responses. Furthermore, PARRY and ELIZA also conversed with each other over ARPANET.

Saygin et al. (2000) study is a comprehensive work that details the history and applications of chatbots with examples. For further elaborations and research, it should be visited.

In 1995, a chatbot named A.L.I.C.E. was developed by Richard Wallace. A.L.I.C.E. was built on AIML (Artificial Intelligence Markup Language), an XML-based language. AIML was developed between 1995 and 2000 by the software developer team of A.L.I.C.E. and Richard Wallace, as mentioned in Wallace's study (2003). AIML provides a method for defining conversational patterns and responses using tags to organize data and responses. A.L.I.C.E. won the Loebner Prize three times, as noted by Wallace (2003).

Another notable chatbot, Albert One, was developed by Robby Garner in 1997. Albert gained fame in those years for its highly human-like conversations and was listed in the 2001 Guinness Book of World Records as the most human computer program in the world. As mentioned in Deryugina's study (2010), Albert One was published online in 1995 to collect data on what people would say to a chatbot on the web, resulting in the collection of a large dataset.

Another prominent chatbot in the historical timeline is Jabberwacky, developed by Rollo Carpenter. Development began in 1988, and it went live on the web in 1997. What sets Jabberwacky apart from other chatbots is its ability to learn. It remembers everything you say, can generate contextually appropriate responses, and can learn new things from you. It was designed to make normal human conversation more interesting, exciting, and fun by being taught slang English, word games, jokes, and various language traits.

According to Fryer and Carpenter (2006), in their study on the Jabberwacky, its success is attributed to its continuous learning capability. They mention that Jabberwacky almost becomes human-like because it learns something new from every interaction with humans, constantly improving itself. Moreover, they note that Jabberwacky is enjoyable to interact with and can sustain longer conversations, which contributes to its addictive appeal among users. Jabberwacky won the Loebner Prize in 2005, underscoring its advancements in chatbot technology and its ability to engage users effectively.

Another chatbot developed by Rollo Carpenter is Cleverbot, which is considered the advanced version of Jabberwacky. While Cleverbot continues to operate actively today, Jabberwacky is no longer actively used.

If we briefly mention the Loebner Prize, it is a competition designed by Dr. Hugh Loebner in 1991 to be held annually, promising $100,000 to the creator of a chatbot that passes the Turing test. Some of the winning chatbots to date include Albert One (1998 and 1999), A.L.I.C.E. (2000, 2001, and 2004), and Jabberwacky (2005 and 2006). These chatbots were selected as winners because they met most of the competition criteria and performed better than other competing chatbots, despite not passing the Turing test. The promised prize amount has never been awarded. The organization of this competition has significantly contributed to the development of this field and motivated professionals working in this area.

Elbot, developed by Fred Roberts in 2008, nearly passed the Turing test by convincing 3 out of 12 human jury members. If it convinces one more jury member, this chatbot will pass the Turing test. As noted by Alazzam et al. (2023), it continued conversations by understanding synonyms, successfully capturing the flow between input and output, and thereby earning the 18th Loebner Prize.

In addition to the widely discussed concept of "chatbot," another term used is "chatterbot," introduced by Mauldin (1994). This term has become more prevalent in research and refers to programs capable of mimicking human relationships and potentially passing the Turing test.

3. INSIGHTS INTO CHATBOTS

In recent years, there has been a significant revolution in the field of chatbots. The introduction of Siri by Apple in 2010 as a digital assistant marked the beginning of a new and advanced competition. There was a strong focus on personalizing chatbots. In 2012, Google introduced Google Now, later rebranded as Google Assistant, which could answer questions, perform web searches, and provide recommendations.

Microsoft developed Cortana in 2014, which can send emails, answer questions, find locations, and engage in conversations. Amazon entered the scene in 2014 with Alexa, an intelligent home device that responds to voice commands. Alexa can play

music, perform internet searches, and engage in conversations based on voice commands.

ChatGPT was developed by OpenAI in 2021 and has arguably become one of the most well-known chatbots. Due to its requirement for vast amounts of data, significant computational power, and high hardware capabilities, it has piqued curiosity and been widely used by many.

As mentioned earlier, chatbots can be utilized in various fields such as customer service, translation programs, and personal digital assistants. One common feature among them is their ability to use natural language to communicate with humans.

Architectures used in natural language processing (NLP) programs such as text generation, text summarization, or translation include convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and transformers. Neural networks have been highly successful in detecting objects in images. However, they have not been as successful in tasks like text summarization, text generation, or translation. On the other hand, transformers excel in text processing tasks.

In the RNN architecture, there are input and output layers, with hidden layers that establish connections between them. Inputs are sequentially processed, which means RNNs suffer from the issue of forgetting earlier inputs when processing long sequences, leading to context loss. This limitation becomes problematic, especially in translation tasks where sentence structures and grammatical rules can differ between source and target languages.

Moreover, because RNNs process inputs sequentially, training a large model with big data is not feasible. They cannot exploit parallelism effectively, so adding more Graphics Processing Units (GPUs) does not significantly speed up training or testing times.

4. TRANSFORMERS

In 2017, Vaswani et al. (2017) introduced a new method that would surpass CNNs, RNNs, and LSTMs: Transformers, in their paper "Attention is All You Need." The Transformers architecture includes an encoder, decoder, softmax activation function,

and self-attention mechanism. These features enable the Transformer architecture, as discussed in Dai et al. (2019), to process long sentences without losing context.

Vaswani et al. (2017) describes how tokenization works in the Transformer architecture. First, incoming input is processed by a tokenizer, which converts words into corresponding numerical tokens from a token ID library. Since computers do not understand the meaning of words and processing letters, including special characters, can be complex, assigning numbers to words and processing them in this way simplifies the task.

Next, the embedding stage follows, where these tokens are transformed into vectors in a multidimensional space. Just like in the tokenizer dictionary, the positions of these vectors in space are predefined. Words that are semantically similar to each other are positioned closer together in this space, reflecting their semantic proximity. Then, positional embedding is calculated. This specifies the position of each word in the sentence, capturing context. This is helpful for understanding long texts and determining word order when translating between languages. Next, token embeddings and positional embeddings are summed up, combining the spatial positions of words and their positions relative to other words in the sentence to create the final input embedding. Subsequently, these final input embedding vectors are passed to the self-attention mechanisms inside the encoder and decoder. Here, attention weights (parameters) are assigned to each word based on its semantic distances to other words in the sentence. This process repeats multiple times, hence it's called the multi-headed self-attention mechanism. The goal here is to capture different aspects of the language with each iteration. The next stage is the feed-forward network. Here, a score is assigned to the upcoming new word based on the attention weights obtained from previous stages, and the model attempts to predict accordingly. Afterwards, the softmax activation function is applied. In this stage, probability values are assigned to candidate words for the upcoming new word, ensuring their sum equals 1, and the word with the highest probability is selected. Finally, the selected word is converted back to token IDs or words through de-tokenization, completing the tokenization process.

This mechanism enables translations like transforming "The cat is..." into "The cat is apple." rather than "The cat is sleeping," achieving a potentially correct result. In

another example, if a sentence contains two names and later refers back to the first name as "it," the self-attention mechanism can capture this context effectively.

Transformer architecture's another facilitating feature is its ability to process sequences in parallel rather than sequentially as in RNNs. This parallelization greatly saves time and enables training of very large models on massive datasets. As a result, models like BERT (Bidirectional Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer), T5 (Text-to-Text Transfer Transformer), and DialoGPT have been successfully trained.

As introduced in their study (Xue et al., 2020), mT5 (Multilingual Pre-trained Text-to-Text Transformer) is a variant of the T5 model that encompasses multiple languages. This study demonstrated that the T5 model can support multiple languages.

As introduced in the article by Devlin et al. (2018), BERT, unlike previous models at that time, is designed to learn from both the left and right contexts of a sentence simultaneously. It accomplishes this using unlabeled text. By fine-tuning only its top layer, BERT can produce state-of-the-art models and be applied in natural language processing tasks such as question answering and language translation.

GPT-2, as mentioned by Radford et al. (2019), has over an order of magnitude more parameters than GPT and is also larger than BERT. The results of this study have shown that GPT-2 achieves state-of-the-art performance on 7 out of 8 tested language modeling datasets without specific training for each task. This demonstrates that high-capacity models can learn to perform various tasks without explicit supervision by maximizing the likelihood of diverse text.

DialoGPT, trained with the architecture of GPT-2 (Radford et al., 2019), differs in that it uses broader dialogue datasets collected from Reddit discussions during training. It is hoped that this approach will lead to better understanding of source and target sentences in conversations. DialoGPT has been released as open-source for developers to use. For our thesis project, we plan to finetune DialoGPT to develop a chatbot.

5. METHODOLOGY

After discussing a bit about the history of chatbots, let's talk about the steps to develop a chatbot and particularly focus on the fine-tuning method we used in our work.

Choosing the right model is crucial when developing a chatbot. It is true that each model excels in one of the natural language processing tasks (such as text generation, question answering, etc.) and provides more accurate results. Moreover, there are specialized versions of large language models pre-trained on specific domains to enhance their performance in those domains. For example, as mentioned earlier, BERT is a widely known model. If you are working with a clinical dataset or aiming to develop a chatbot in the healthcare domain (like a medical diagnosis assistant that asks users to input symptoms and predicts their illness), it would be more suitable to choose specialized models like ClinicalBERT or Bio-ClinicalBERT, which are pre-trained on clinical data and optimized for tasks related to healthcare, making them more appropriate for developing healthcare-related chatbots.

In Ling's study (2023), BERT base model, BioClinicalBERT model, and a CNN model were compared in their ability to analyze patients' drug reviews and accurately classify their satisfaction levels as positive, neutral, or negative. The findings revealed that BioClinicalBERT outperformed both the BERT base model and CNN. Specifically, BioClinicalBERT showed a significant improvement over the BERT base model with a 0.11 increase in macro F1 score and recall results. Additionally, the CNN model demonstrated the capability to select important keywords and accurately perform sentiment analysis even in complex sentences.

After selecting the right model, the next steps involve obtaining or creating a sufficient dataset. Following this, setting up the environment and downloading necessary libraries are essential tasks. Subsequently, fine-tuning should be conducted. As previously mentioned, fine-tuning involves training only the top layer of the model with a task-specific dataset. This process allows the model to retain the language-related concepts learned during its initial training in the lower layers. By training the top layer on specific data, the model synthesizes information from both the fine-tuned top layer and the underlying layers.

In the study by Tajbakhsh et al. (2016), which focused on medical image analysis, the performance of deep fine-tuned CNNs was compared with that of pre-trained deep CNNs. The results indicated that both approaches achieved nearly equivalent scores, with the fine-tuned model even outperforming the pre-trained model when trained on limited data. Training a model from scratch is costly and time-consuming. Therefore, fine-tuning is often the most practical choice for such projects.

6. IMPLEMENTATION

Since we were trying to develop a chatbot, we thought DialoGPT was the most suitable model for this NLP task. DialoGPT is a version of the GPT2 model pre-trained on conversational sentences from Reddit. We used the dialogpt-medium model trained on 147M multi-turn dialogues from Reddit discussions available on Huggingface. Following the provided instructions, we loaded DialoGPT's tokenizer and model. For finetuning the model, we utilized the dataset from Huggingface's datasets/li2017dailydialog/daily_dialog, which contains 11,118 lines of conversations, where each line consists of multiple conversational sentences. To better understand, let's examine the first line of the dataset:

{'dialog': ['Say , Jim , how about going for a few beers after dinner ? ', ' You know that is tempting but is really not good for our fitness . ', ' What do you mean ? It will help us to relax . ', " Do you really think so ? I don't . It will just make us fat and act silly . Remember last time ? ", " I guess you are right.But what shall we do ? I don't feel like sitting at home . ", ' I suggest a walk over to the gym where we can play singsong and meet some of our friends . ', " That's a good idea . I hear Mary and Sally often go there to play pingpong.Perhaps we can make a foursome with them . ", ' Sounds great to me ! If they are willing , we could ask them to go dancing with us.That is excellent exercise and fun , too . ', " Good.Let ' s go now . ", ' All right . '], 'act': [3, 4, 2, 2, 2, 3, 4, 1, 3, 4], 'emotion': [0, 0, 0, 0, 0, 0, 4, 4, 4, 4]}

We encoded the dataset with max_length=128. We used an A100 GPU and set the parameters as follows: Epoch=10, train_batch_size=16, evaluation_batch_size=64, weight_decay=0.01. To see if fine-tuning improved the model's accuracy, we printed the validation loss values before and after fine-tuning. Here are the results:

- Evaluation Results before fine-tuning : 4.766558647155762
- Evaluation Results after fine-tuning : 1.745113492012024

Finally, we decoded to finalize the model and named it as PractiChat.

After fine-tuning, we developed an algorithm to compute the difficulty score of texts written by both users and PractiChat. This algorithm takes into account the sentence length and the frequency of words in everyday usage when calculating the difficulty score. We found a dataset on Kaggle that provides word frequencies, and we ranked words from 1 to 10,000 based on their frequency of occurrence. We reasoned that less frequently used words are more challenging compared to commonly used ones. Therefore, we assigned scores based on the logarithm of the indices given to all words.

Additionally, we considered sentence length. According to research by Deveci (2019), Elsevier journal publications recommend an average of 12-17 words per sentence. Experimentally, we decided to set the threshold at 15 words per sentence. By taking the logarithm base 15 of the word count in the sentence being analyzed, we created a score related to the sentence's length.

Finally, we multiply these two scores and display them as a percentage beneath each sentence. In future work, our goal is to use this score within PractiChat to assess users' English proficiency and generate responses tailored to their level. This approach will enable users to practice with vocabulary appropriate for their proficiency level effectively.

7. RESULTS AND DISCUSSION

Here are some example conversations we conducted using PractiChat through the interface we designed for it in images 1, 2, 3, 4, 5, and 6. Difficulty scores of the sentences are also printed below each sentence.

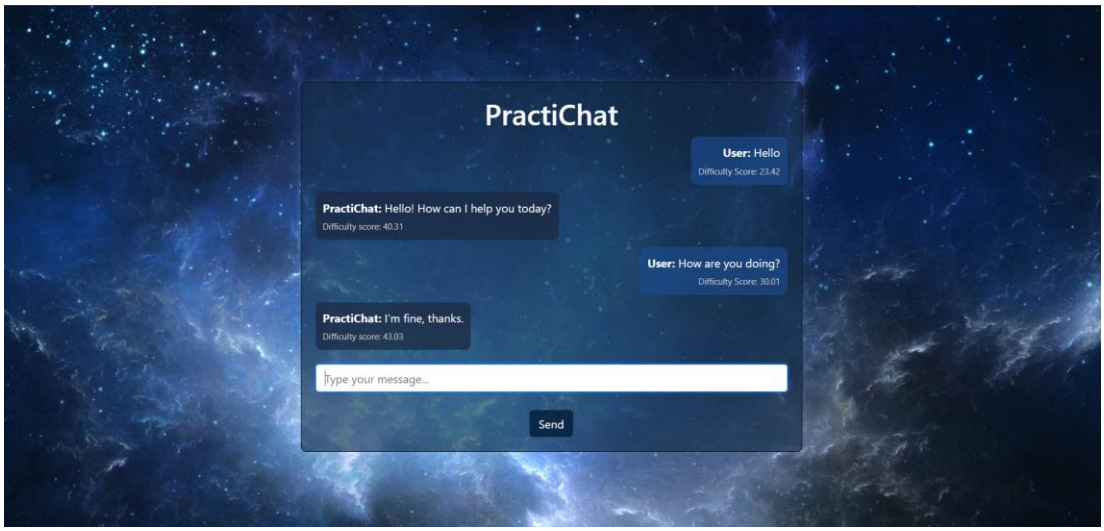

Image 1. Example of a conversation

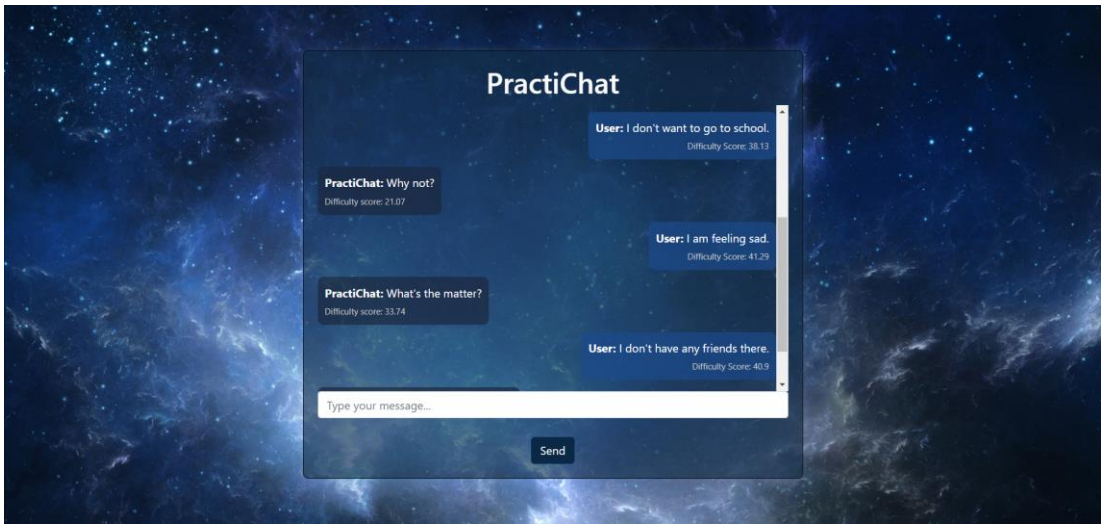

Image 2. Example of a conversation

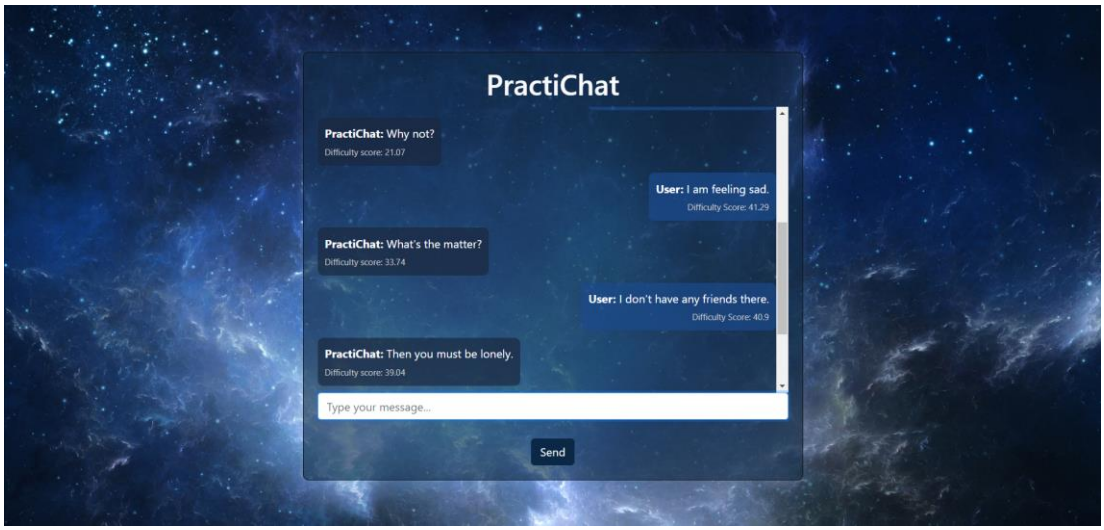

Image 3. Example of a conversation

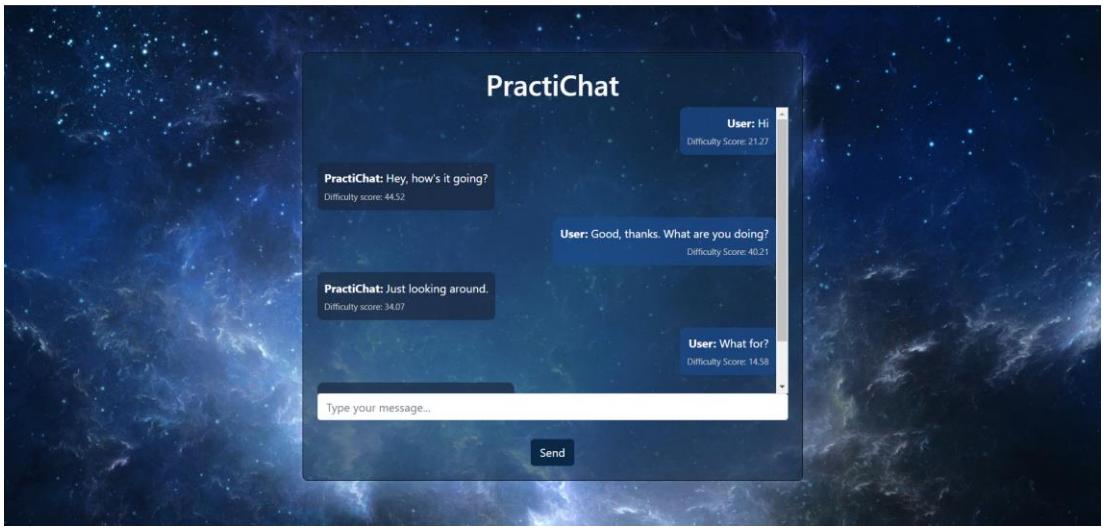

Image 4. Example of a conversation

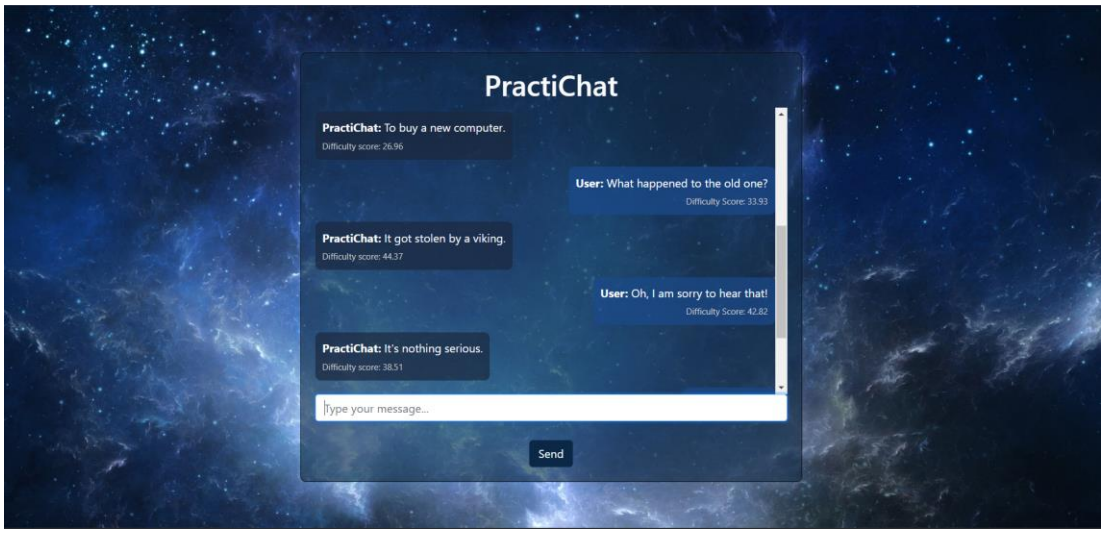

Image 5. Example of a conversation

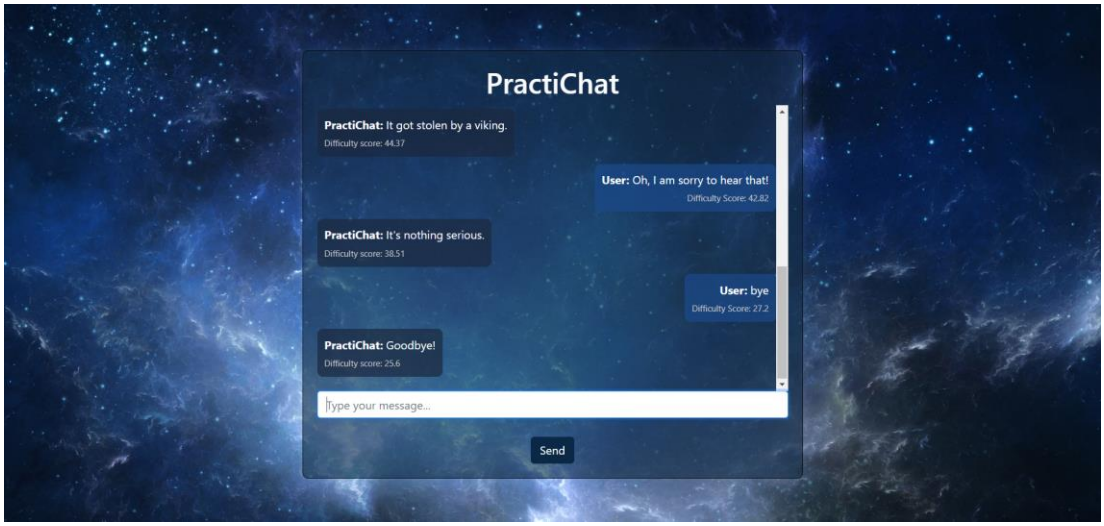

Image 6. Example of a conversation

In this study, we specifically fine-tuned the DialoGPT model, which is based on the GPT2 architecture, with a dataset consisting of dialogues to facilitate conversational interactions. We named the developed chatbot PractiChat. We shared the resources and parameters we used for this purpose. The results we obtained were quite satisfying; the model performed well in maintaining conversations, generating responses, and asking questions. Moreover, it also handles slang words and responds appropriately, such as saying "That is not nice."

In future work, we aim to compare results by fine-tuning other transformer models with the same dataset. Additionally, experimenting with a larger dataset for fine-tuning could be explored.

Future studies could leverage the difficulty score to personalize PractiChat, potentially adding features like voice interaction to further assist in English practice. Ultimately, these advancements could enhance the user experience and effectiveness of practicing English language skills.

8. CONCLUSION

In this thesis study, we embarked on a journey from the earliest chatbots in history like ELIZA and PARRY to modern-day AI assistants such as Siri and ChatGPT. We explored the evolution of technologies used in chatbot development from RNNs and LSTMs to transformers, focusing particularly on the transformers architecture. We provided a detailed explanation of how transformers work and how they differ from traditional neural networks.

Next, we outlined the step-by-step process of developing a chatbot, discussing the transformers model we chose and the reasons behind our choice. Since our goal was to develop a conversational AI, we utilized DialoGPT, which is trained on Reddit conversation threads using the GPT2 architecture. We explained what it means to fine-tune a model and shared the dataset, resources, and parameters we used for our own fine-tuning process.

In the Results and Conclusion section, we presented a conversation conducted with PractiChat, the chatbot we named. Additionally, we developed an algorithm to measure the difficulty score of sentences, calculating and displaying this score

beneath each sentence in the interface, for both PractiChat's responses and user-generated sentences.

In future work, we aim to use this difficulty score to personalize PractiChat according to the user's English proficiency level, thereby enabling effective English practice tailored to the user's needs.

9. REFERENCES

Neff, G., Nagy, P. (2016) Talking to Bots: Symbiotic Agency and the Case of Tay, International JournaL of Communication. 10, pp. 4915-31

Turing, A. M. (1950) Computing Machinery And Intelligence, Mind. 49, pp. 433-460.

Searle, J., 1980, ‘Minds, Brains and Programs’, Behavioral and Brain Sciences, 3: 417–57

Searle, J., 1999, ‘The Chinese Room’, in R.A. Wilson and F. Keil (eds.), The MIT Encyclopedia of the Cognitive Sciences, Cambridge, MA: MIT Press.

Cole, David, "The Chinese Room Argument", The Stanford Encyclopedia of Philosophy (Summer 2023 Edition), Edward N. Zalta & Uri Nodelman (eds.), URL = <https://plato.stanford.edu/archives/sum2023/entries/chinese-room/>.

Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Commun. ACM 9, 1 (Jan. 1966), pp. 36–45. https://doi.org/10.1145/365153.365168

Colby, K.M. Hilf, F.D. and Weber, S. (1971), ‘Artificial Paranoia’, Artificial Intelligence 2, pp. 1–25.

Saygin; Cicekli; Akman (2000), "Turing Test: 50 years later" (PDF), Minds and Machines, 10 (4): 463–518, doi:10.1023/A:1011288000451

Wallace, R. (2003). The elements of AIML style. Alice AI Foundation, 139.

Deryugina, O.V. (2010) Chatterbots. Sci. Tech.Inf. Proc. 37, 143–147. https://doi.org/10.3103/S0147688210020097

Fryer, Luke & Carpenter, Rollo. (2006). Bots as language learning tools. Language, Learning and Technology. 10. 8-14.

Alazzam, B. A., Alkhatib, M., & Shaalan, K. (2023). Artificial intelligence chatbots: a survey of classical versus deep machine learning techniques. Inf. Sci. Lett, 12(4), 1217-1233.

Mauldin, M.L. (1994) Chatterbots, tinymuds, and the turing test: Entering the loebner prize competition, in AAAI, 941994, pp. 16–21.

Vaswani, A., Shazeer, N.M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. (2017). Attention is All you Need. Neural Information Processing Systems.

Dai, Zihang & Yang, Zhilin & Yang, Yiming & Carbonell, Jaime & Le, Quoc & Salakhutdinov, Ruslan. (2019). Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. 2978-2988. 10.18653/v1/P19-1285.

Xue, Linting & Constant, Noah & Roberts, Adam & Kale, Mihir & Al-Rfou, Rami & Siddhant, Aditya & Barua, Aditya & Raffel, Colin. (2020). mT5: A massively multilingual pre-trained text-to-text transformer.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K.(2018) Bert: Pretraining of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Ling, Yue. (2023). Bio+Clinical BERT, BERT Base, and CNN Performance Comparison for Predicting Drug-Review Satisfaction.

Tajbakhsh, Nima & Shin, Jae & Gurudu, Suryakanth & Hurst, R Todd & Kendall, Christopher & Gotway, Michael & Liang, Jianming. (2016). Convolutional Neural Networks for Medical Image Analysis: Fine Tuning or Full Training?. IEEE Transactions on Medical Imaging. 35. 1-1. 10.1109/TMI.2016.2535302.

Deveci, T. (2019). Sentence Length in Education Research Articles: A Comparison between Anglophone and Turkish Authors.

Jabberwacky - About Thoughts, June 15, 2024. http://www.jabberwacky.com/j2about.

microsoft/DialoGPT-medium . Hugging Face, June 12, 2024 https://huggingface.co/microsoft/DialoGPT-medium

li2017dailydialog/daily_dialog . Datasets at Hugging Face https://huggingface.co/datasets/li2017dailydialog/daily_dialog