ODU TIP DERGİSİ/ODU MEDICAL JOURNAL (ODU MED J) **April 2024;11(1):18-32**

ARAŞTIMA MAKALES/ RESEARCH ARTICLE **DOI:** 10.56941/odutip.1403118

A Developed Graphical User Interface-Based on Different Generative Pre-trained Transformers Models

Ekrem Küçük[1,2](ID), İpek Balıkçı Çiçek[1](ID), Zeynep Küçükakçalı[1](ID), Cihan Yetiş[3](ID), Cemil Çolak[1](ID)
[1]Inonu University Faculty of Medicine, Department of Biostatistics and Medical Informatics, Malatya, Turkey
[2]Carbon Health, Senior Software Engineer, Ankara, Turkey
[3]Department of Cardiovascular Surgery, Kilis Prof. Dr. Alaeddin Yavaşca State Hospital, Kilis, Turkey

Received: 02 January 2024, Accepted: 05 April 2024, Published online: 30 April 2024

Abstract

Objective: The article investigates the integration of advanced Generative Pretrained Transformers (GPT) models into a user-friendly Graphical User Interface (GUI). The primary objective of this work is to simplify access to complex Natural Language Processing (NLP) tasks for a diverse range of users, including those with limited technical background.

Method: The development process of the GUI was comprehensive and systematic: Needs Assessment: This stage involved understanding the requirements and expectations of potential users to ensure the GUI effectively addresses their needs. Preliminary Design and Development: The initial designs were created and developed into a functional GUI, emphasizing the integration of features supporting various NLP tasks like text summarization, translation, and question-answering. Iterative Refinement: Continuous improvements were made based on user feedback, focusing on enhancing user experience, ease of navigation, and customization capabilities.

Results: The developed GUI successfully integrated GPT models, including GPT-4 Turbo and GPT-3.5, resulting in an intuitive and adaptable interface. It demonstrated efficiency in performing various NLP tasks, thereby making these advanced language processing tools accessible to a broader audience. The GUI's design, emphasizing user-friendliness and adaptability, was particularly noted for its ability to cater to both technical and non-technical users.

Conclusion: In conclusion, the article illustrates the significant impact of combining advanced GPT models with a Graphical User Interface to democratize the use of NLP tools. This integration not only makes complex language processing more accessible but also marks a pivotal step in the inclusive application of AI technology across various domains. The successful implementation of the GUI highlights the potential of AI in enhancing user interaction and broadening the scope of technology usage in everyday tasks.

Key words: Graphical User Interface (GUI), Generative Pretrained Transformers (GPT) models, Natural Language Processing (NLP), User-Friendly, Accessibility

Farklı Üretken, Önceden Eğitimli Dönüştürücüler Modellerine Dayalı Geliştirilmiş Grafik Kullanıcı Arayüzü

Özet

Amaç: Makale, gelişmiş Üretken Önceden Eğitimli Dönüştürücüler (GPT) modellerinin kullanıcı dostu Grafik Kullanıcı Arayüzü'ne (GUI) entegrasyonunu araştırmaktadır. Bu çalışmanın temel amacı, sınırlı teknik altyapıya sahip kullanıcılar da dahil olmak üzere çeşitli kullanıcıların karmaşık Doğal Dil İşleme (NLP) görevlerine erişimini kolaylaştırmaktır.

Yöntem: GUI'nin geliştirme süreci kapsamlı ve sistematikti: Gereksinim Değerlendirmesi: Bu aşama, potansiyel kullanıcıların gereksinimlerini ve beklentilerini anlamayı içeriyordu ve GUI'nin bu gereksinimleri etkili bir şekilde ele almasını sağlamak için yapıldı. Ön Tasarım ve Geliştirme: İlk tasarımlar, metin özetleme, çeviri ve soru cevaplama gibi çeşitli NLP görevlerini destekleyen özelliklerin entegrasyonunu vurgulayan işlevsel bir GUI olarak oluşturuldu ve geliştirildi. Yinelemeli İyileştirme: Kullanıcının geri bildirimlerine dayalı olarak kullanıcı deneyiminin, gezinme kolaylığının ve özelleştirme yeteneklerinin geliştirilmesine odaklanan sürekli iyileştirmeler yapıldı

Bulgular: Geliştirilen GUI, GPT-4 Turbo ve GPT-3.5 gibi GPT modellerine başarılı bir şekilde entegre edildi ve sezgisel ve uyarlanabilir bir arayüz ortaya çıkarıldı. Farklı NLP görevlerini etkili bir şekilde gerçekleştirme yeteneği göstererek, bu gelişmiş dil işleme araçlarını daha geniş bir kitleye erişilebilir hale getirdi. Kullanıcı dostu ve uyarlanabilirliği vurgulayan GUI tasarımı, teknik ve teknik olmayan kullanıcılara hitap etme yeteneği özellikle dikkat çekti.

Sonuç: Sonuç olarak, makale, gelişmiş GPT modellerini Grafik Kullanıcı Arayüzü ile birleştirmenin NLP araçlarının kullanımını demokratikleştirmedeki önemli etkisini göstermektedir. Bu entegrasyon, karmaşık dil işleme araçlarını sadece daha erişilebilir hale getirmekle kalmaz, aynı zamanda çeşitli alanlarda yapay zeka (AI) teknolojisinin kapsayıcı uygulamasında da bir dönüm noktası işaret eder. GUI'nin başarılı bir şekilde uygulanması, AI'nin kullanıcı etkileşimini geliştirme potansiyelini ve günlük görevlerde teknoloji kullanımının kapsamını genişletme potansiyelini vurgular.

Anahtar kelimeler: Grafik Kullanıcı Arayüzü (GUI), Üretken Önceden Eğitimli Dönüştürücüler (GPT) modelleri, Doğal Dil İşleme (NLP), Kullanıcı Dostu, Erişilebilirlik

Suggested Citation: Küçük E, Balıkçı Çiçek İ, Küçükakçalı Z, Yetiş C, Çolak C. Evaluation of Seasonal Hazelnut Workers' Emergency Department Applications in Ordu Province: A Retrospective Study ODU Med J, 2024;11(1): 18-32.

Address for correspondence/reprints:

İpek Balıkçı Çiçek

Telephone number: +90 (422) 341 06 60 / 1337

E-mail: ipek.balikci@inonu.edu.tr

INTRODUCTION

Language serves as the fundamental basis for human communication and influences our interactions with the external environment. The advent of natural language processing (NLP) has revolutionized our interaction with machines. NLP has transformed communication by enabling humans to interact with technology in a more authentic and intuitive way. The rapid increase in the amount of written information available online has driven the progress of NLP. The field of NLP has undergone substantial advancements, progressing from rudimentary rule-based systems to intricate deep learning-based models (1). Due to the intricate nature of human language, natural language interpretation and generation have historically posed significant challenges in the field of NLP, despite notable progress. Nevertheless, recent breakthroughs have paved the way for innovative strategies to address these issues. The development of the Generative Pre-Trained Transformer (GPT) is a significant achievement in the field of NLP. (2).

GPT rose to prominence with the debut of ChatGPT by OpenAI, a research firm focused on creating AI technology (3). GPT is a deep learning model that has been pre-trained on huge corpora of text data and may be fine-tuned for tasks such as language synthesis, sentiment analysis, language modeling, machine translation, and text categorization. GPT's transformer architecture is a huge step forward over earlier techniques to NLP, such as RNN and CNN. It employs a self-attention method to enable the model to consider the context of the full phrase while creating the next word, enhancing the model's capacity to understand and produce language. The decoder is in charge of producing the output text depending on the input representation (4).

GPT is capable of a wide range of NLP tasks. Natural language understanding (NLU) is one of its primary features; it can evaluate and grasp the meaning of text, including detecting entities and relationships in phrases. It's also skilled at natural language generation (NLG), which means it can generate text output such as creative material or detailed and insightful answers to inquiries. GPT may also be used as a code generator, writing computer code in languages such as Python or JavaScript.

GPT may also be used to answer questions, which means it can offer factual summaries or build tales depending on the supplied text. Furthermore, GPT may summarize material, such as delivering a concise review of a news story or research paper, and it can be used for translation, allowing text to be translated from one language to another. Overall, GPT's capacity to handle a wide range of NLP tasks with high accuracy and precision makes it a useful tool for industries such as finance, healthcare, marketing, and others. As NLP technology advances, we may expect GPT and other language models to become progressively more complex and powerful, allowing us to connect with machines more naturally and efficiently.

GPTs have revolutionized NLP, enabling rapid advancements in various industries. Despite its widespread adoption, there is a lack of comprehensive understanding of GPT's architecture and capabilities. This gap not only hampers optimization but also limits the technology's full potential. Therefore, a detailed study is essential to explore its architecture, supporting technologies, potential applications, current challenges, and future prospects. This unexplored territory and the need for a nuanced understanding served as the driving forces behind our research endeavor.

This article presents the design and implementation of a graphical user interface (GUI) that leverages the capabilities of GPT models for various natural language processing tasks.

METHODS

Objective and Scope

The primary aim of this research is to develop a user-friendly Graphical User Interface (GUI) that integrates GPT for executing a range of NLP tasks. The scope includes text summarization, translation, and question-answering functionalities.

Research Design and Methodology

Needs Assessment: Conducting surveys and interviews to identify user requirements and the NLP tasks that the GUI should support.

Preliminary Design: Sketching initial wireframes and integrating a base GPT model.

Development: Actual coding, leveraging Python rest api endpoints and Angular WEB UI for user interaction.

Evaluation: Performance metrics and user feedback are collected to assess the GUI's efficiency and usability.

Iteration: Refinements are performed based on the evaluation results.

Technical Stack

Programming Language: Python 3.x

Python Libraries: OpenAI, Flask, Flask - Cors

GUI Framework: Angular 16.2, PrimeNg 16.3, PrimeFlex 3.3.1

Deployment: Docker

Version Control: Git

Hardware Specifications

Processor: Virtual 2 Core CPU

RAM: 4GB RAM

Storage: 20GB SSD

Operating System: Debian 12.1.0

System Architecture

The system is modular, consisting of:

User Interface Layer: The User Interface (UI) Layer of our system is meticulously crafted using the Angular Framework. Angular stands out as a premier TypeScript-based open-source web application framework, spearheaded by the esteemed Angular Team at Google, with active contributions from a vibrant community of individuals and corporations alike. Renowned for its versatility and robustness, Angular offers a sophisticated platform that empowers developers in the creation of dynamic and responsive client-side web applications.

In summary, the User Interface Layer of our system, developed using the Angular Framework, embodies the epitome of modern web development practices. With its powerful features, comprehensive toolset, and thriving community, Angular empowers developers to build sophisticated, scalable, and visually stunning web applications that delight users and drive business success.

Business Logic Layer: At the heart of our system lies the Business Logic Layer, a pivotal component that orchestrates the functionality and intelligence of our application. This layer seamlessly integrates cutting-edge GPT (Generative Pre-trained Transformer) models, serving as the backbone for handling a myriad of tasks ranging from natural language processing to text generation and beyond.

Central to the functionality of the Business Logic Layer is its adept handling of various GPT tasks, leveraging the power and versatility of state-of-the-art machine learning models. Whether it's generating coherent text, summarizing documents, or engaging in conversation, the GPT models encapsulated within this layer exhibit a remarkable ability to comprehend and generate human-like responses.

To facilitate seamless interaction with the user interface, the Business Logic Layer exposes its functionality through REST API endpoints. These endpoints serve as the conduit through which data and requests are exchanged between the user interface and the underlying logic of the system. By adhering to RESTful principles, our API endpoints ensure a standardized and intuitive interface for communication, fostering interoperability and ease of integration with external systems.

Underpinning the Business Logic Layer is a robust implementation of the business service layer, meticulously developed using Python and the Flask library. Python, renowned for its simplicity, readability, and versatility, serves as the foundation for our business logic, enabling rapid development and iteration. Flask, a lightweight and extensible web framework for Python, further augments our business service layer by providing essential features for routing requests, handling HTTP methods, and managing application state.

Through the harmonious interplay of GPT models, RESTful APIs, Python, and Flask, the Business Logic Layer empowers our system with unparalleled intelligence and functionality. Whether it's automating tedious tasks, providing insightful recommendations, or engaging users in meaningful dialogue, this layer serves as the linchpin for delivering a transformative user experience. With its sophisticated capabilities and robust architecture, the Business Logic Layer lays the groundwork for a truly innovative and impactful application.

UI – Service Interaction:

A REST API, sometimes referred to as a RESTful API, is an application programming interface (API) or web API that adheres to the principles and limitations of the REST architectural style. It enables communication and interaction with RESTful web services. REST, an acronym for representational state transfer, was devised by computer scientist Roy Fielding (5).

An Application Programming Interface (API) constitutes a comprehensive set of guidelines and structures for the development and integration of application software. It functions akin to a contractual agreement between a provider of information and a user, delineating the specific data required by the user (termed as the 'call') and the corresponding output necessitated by the provider (referred to as the 'response') (6).

In this work, business logic layer provides set of API interfaces that receives calls from user interface layer, processes it, and transforms to OpenAI API model (7). Service layer has business logic to transform user inputs to Open-AI model also provides additional inputs for better NLP results (8). Figure 1 depicts workflow processes for the design of the proposed system.

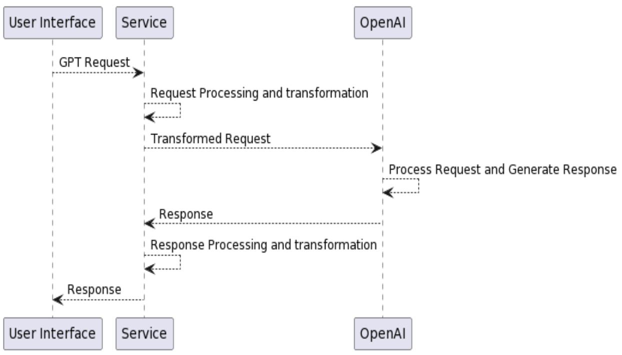

Figure 1. Workflow processes for the design of the proposed system

User Experience Design

Embedded within the core tenets of Human-Centered Design, this particular design ethos places paramount importance on the principles of simplicity and accessibility, meticulously crafted to prioritize the seamless integration of user needs and desires. With a steadfast commitment to enhancing user comfort and ease, the design intricately weaves together intuitive interactions and ergonomic features, forging a symbiotic relationship between user and interface.

At the heart of this design philosophy lies a continuous feedback loop, meticulously curated through a dual-channel approach comprising electronic mail and graphical user interfaces. This comprehensive feedback mechanism serves as the lifeblood of the design process, providing invaluable insights into user experiences and preferences. By leveraging these insights, the design team is empowered to iterate and refine the design iteratively, ensuring its ongoing alignment with evolving user expectations.

Moreover, this iterative refinement process serves as a testament to the design's adaptability and versatility, allowing it to seamlessly cater to the diverse needs of its user base. Each refinement is meticulously crafted to enhance the overall user experience, fostering a sense of inclusivity and accessibility across all touchpoints of interaction.

Furthermore, this commitment to continuous improvement extends beyond mere functionality, delving into the realm of emotional resonance and user delight. Through thoughtful iteration and refinement, the design transcends its utilitarian purpose, forging meaningful connections with users and fostering a sense of loyalty and trust.

In essence, this design philosophy encapsulates the essence of Human-Centered Design, where simplicity, accessibility, and user-centricity converge to create transformative experiences. By embracing a culture of continuous feedback and iterative refinement, this design ethos stands as a beacon of innovation and empathy, enriching the lives of users in profound and meaningful ways.

RESULTS

Figure 2 illustrates the proposed interface's settings and configuration in detail. It showcases the layout and organization of various control elements, demonstrating how users can interact with and adjust the interface. This visualization highlights the user-friendly design, emphasizing ease of navigation and customization options available within the interface. It provides a clear representation of how the interface can be tailored to meet specific user needs, displaying both the default settings and the range of adjustments possible for personalization and efficiency. The figure serves as a comprehensive guide for understanding the interface's functionality and adaptability.

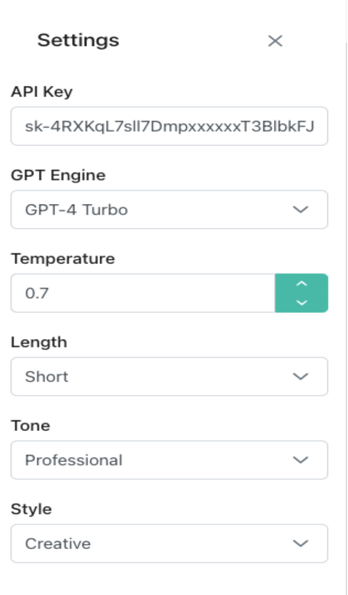

Figure 2. Settings and Configuration of the proposed interface

An API key, also known as an Application Programming Interface key, is a distinct identification utilized for the purpose of authenticating a user, developer, or calling application to an API. API keys are utilized to monitor and regulate the usage of the API, primarily to safeguard against any malicious or abusive activities. The API key serves as both a confidential authentication token and a distinctive identification.

Here API key is provided by OPENAI platform subscription.

GPT Engine: OpenAI provides set of GPT engines, each solves different use cases. The GPT models were sourced from Open-AI. GPT models listed are below in Table 1 [16]:

Table 1. GPT models and algorithms that can be used in the developed software

Models	**Description**
GPT-4 and GPT-4 Turbo	A set of models that improve on GPT-3.5 and can understand as well as generate natural language or code
GPT-3.5	A set of models that improve on GPT-3 and can understand as well as generate natural language or code
GPT base	A set of models without instruction following that can understand as well as generate natural language or code
DALL·E	A model that can generate and edit images given a natural language prompt
TTS	A set of models that can convert text into natural sounding spoken audio
Whisper	A model that can convert audio into text
Embeddings	A set of models that can convert text into a numerical form
Moderation	A fine-tuned model that can detect whether text may be sensitive or unsafe
GPT-3-Legacy	A set of models that can understand and generate natural language

In this work we have adopted “GPT-4 Turbo, GPT-4 with vision, GPT-4, GPT4-32K and GPT-3-Turbo” models.

Temperature: This parameter is used to control randomness of the output. What sampling temperature to use, between 0 and 2. Higher values like 0.8 will make the output more random, while lower values like 0.2 will make it more focused and deterministic.

Length: This parameter controls length of output. Options are:

- Long
- Medium
- Short

Tone: Defines tone of the output. Options are:

- Professional
- Causal
- Enthusiastic
- Informational
- Funny

Style: Defines style of the output. Options are:

- Creative
- Balanced
- Sensitive

GPT Prompt screen is given in Figure 3.

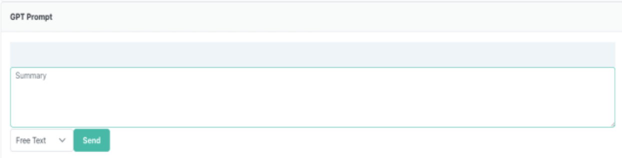

Figure 3. GPT Prompt screen

GPT Prompt screen provides user – GPT interaction. Context provides additional information related with problem.

The Transformer architecture uses contextual embeddings to represent words, meaning that the same word can have different embeddings based on its surrounding words. This allows GPT models to capture nuanced meanings and relationships between words, thereby making better predictions.

In first example, we have not provided any information about the input:

The screen showing that no information is provided regarding the login is given in Figure 4.

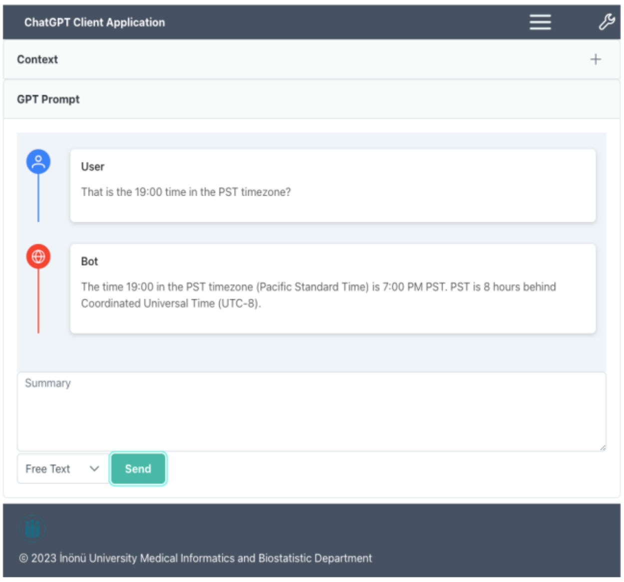

Figure 4. The screen showing that no information is provided about the login

In the second example we have provided context about the input: The screen showing that the context for the entry is provided is shown in Figure 5.

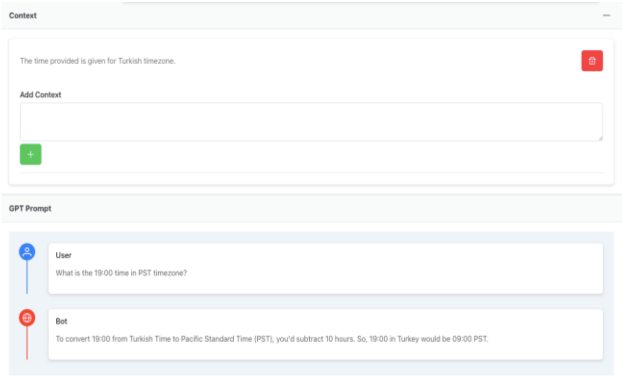

Figure 5. Screen showing that context is provided for the input

Context provides better interpretation and more accurate results for the input.

Free Text Input:

The free text entry screen is shown in Figure 6 below.

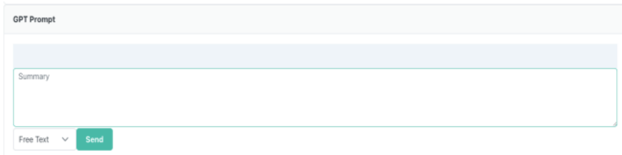

Figure 6. Free text input screen

Free text input provides chat-GPT interaction with user. System sends settings parameters and context to server to support different use cases and optimize responses. User's input text screen is shown in Figure 7 below.

Input:

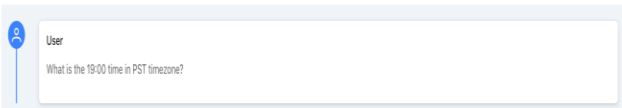

Figure 7. User's input text

Output - 1:

Model: GPT-4 Turbo

Temperature: 0,7

Length: Long

Tone: Funny

Style: Creative

A screenshot of an example of a long response from the system to the user is given in Figure 8 below.

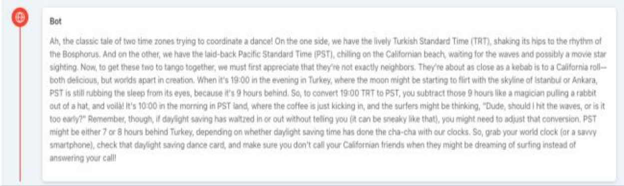

Figure 8. Example of long response from the system to the user

Output - 2:

Model: GPT-4 Turbo

Temperature: 0,7

Length: Medium

Tone: Informational

Style: Balanced

A screenshot of the shorter response sample given by the system to the user is given in Figure 9.

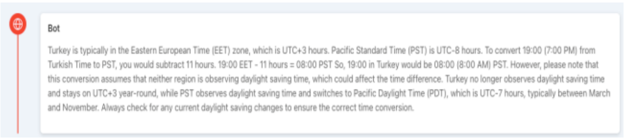

Figure 9. Example of a shorter response given by the system to the user

Text Translate:

Predefined context by software provides information about the input and the transformation to be performed. The results with same transformation will be applied all inputs provided. The image of the screen where the user enters text for text translation is given in Figure 10. The text entered by the user is translated and given as output.

Input:

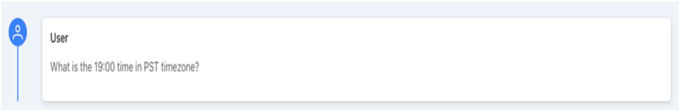

Figure 10. User's input text

File Translate

File translate provides file upload utility to profile external file content as input. Predefined context by software provides information about the input and the transformation to be performed. The transformation is applied to file content provided and result can be viewed with transformed input. The file upload screen for file translation is shown in Figure 11. The file that needs to be translated is selected and uploaded to the system, and the file translation output is obtained from the system.

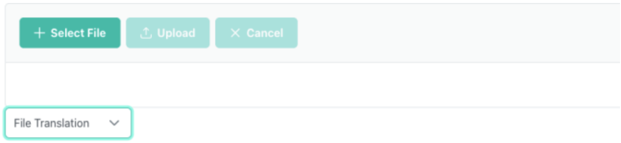

Figure 11. File upload screen for file translate

Cardiovascular Surgery File Translate:

This tool provides pre-defined context and back-end service reads uploaded data, transforms it and uses Open-AI API endpoints with context and content to perform NLP operations. An example of the text that the user wants translated about cardiovascular disease is given in Figure 12.

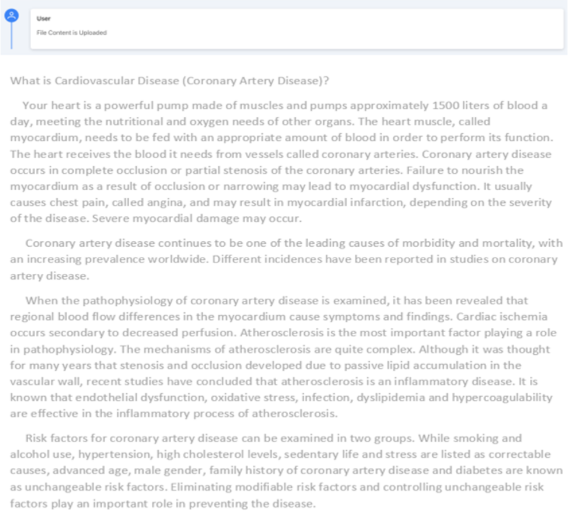

Figure 12. Example of user-uploaded text about cardiovascular disease

Cardiovascular Surgery Paraphrase:

Cardiovascular Surgery Paraphrase provides paraphrasing utility to paraphrase inputs in Cardiovascular Surgery domain. Pre-defined context has been provided by system to control NLP engine. An example of the output where the system comments on the entry after the user enters information about Cardiovascular surgery is shown in Figure 13.

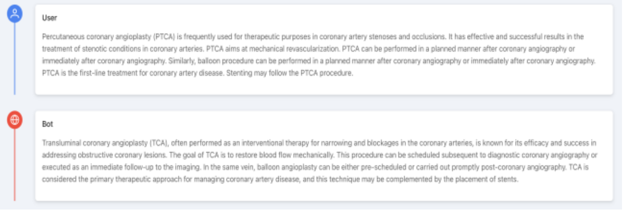

Figure 13. Example of output after the user enters information about Cardiovascular surgery and the system provides a comment on the entry.

DISCUSSION

This work has significant implications for the broader adoption of NLP technologies, particularly for users without a technical background. It opens up new avenues for research and development in the field of user-friendly interfaces for complex machine-learning modelsGPT is an AI language model created by OpenAI, Inc. GPT, an abbreviation for Generative Pre-trained Transformer, is an artificial intelligence system that use a Transformer architecture to generate original material by using a pre-existing dataset during its training phase. GPT was especially designed for NLP tasks that encompass two fundamental elements: (1) comprehending and discerning the semantic meaning of a phrase (natural language comprehension), and (2) producing new phrases based on the inputs (natural language generation). Classical NLP models are rule-based and offer restricted responses by adhering to a predetermined set of encoded rules. As a result, they lack adaptability and struggle to accommodate the dynamic and diverse characteristics of language (9).

The origins of NLP models may be traced back to 1949, with the introduction of Weaver's memorandum which laid the foundation for the notion of machine translation (MT). In the early stages of NLP, the primary focus of algorithms was on machine translation (MT), however there were other models developed with more extensive capabilities (10). Before 1990, systems were predominantly rule-based and heavily influenced by language ideas. The early 1990s witnessed a significant breakthrough with the implementation of statistical models, which was subsequently accompanied by a transformative move towards machine learning. The rise of deep learning in the early 2000s established the foundation for present-day NLP models. The presentation of the pioneer neural language model by Bengio et al. in 2003 was a significant early advancement (11). This model was a feed-forward neural network with a single hidden layer, and it is thought to be the first model that employed the word embedding technique. The 2010s witnessed a significant advancement in technology due to the rapid growth in computer processing capabilities and the collection of extensive datasets. This advancement resulted in the successful implementation of recurrent neural networks (RNN) and long short-term memory (LSTM) models. These sophisticated network architectures offer significant benefits in forecasting or classifying sequential data. Consequently, two innovations have laid the groundwork for GPT. Initially, novel algorithms were proposed, including sequence-to-sequence learning (2014) (12), attention (2015), and self-attention (2017) (13), which significantly enhanced the effectiveness of generative natural language processing (NLP) models (referred to as "G" in GPT). The second significant development was the introduction of innovative 'word embedding' methods. These methods include representing words as numerical data that captures their importance, frequency of use, and user-defined meanings. This allows words with comparable meanings to have similar numerical values. Word2Vec12 pioneered these techniques in 2013 (14). These methodologies proved to be more effective than previous methods and enabled training on far larger data sets. The concept of large pre-trained language models was introduced in 2016 (15). (The letter "P" in GPT) The Transformer, which is the primary architect of GPT, was launched in 2017, alongside significant technological and conceptual advancements in NLP (16). This innovative architecture facilitated the collection and processing of the semantic significance of words within sentences by NLP models. The first Transformer design has both an encoder and a decoder. The encoder takes inputs for processing and transforms them into a sequence of uninterrupted representations using six identical layers. Afterwards, the decoder sends these representations through six further layers that are comparable in order to produce outputs. Both the encoder and decoder layers comprise a sublayer known as 'multi-head self-attention' and another sublayer known as a 'feed-forward' network, which is fully coupled. Furthermore, within each decoder layer, there exists a sublayer known as 'masked self-attention' that exclusively utilizes preceding words within a phrase to forecast words at a certain place (auto-regression). The GPT model undergoes a biphasic training phase (17). The initial phase entails the unsupervised training of the model using an extensive collection of unannotated textual input. The model's 'pre-training' involves autonomously acquiring knowledge of linguistic patterns and representations. With a substantial rise in size, GPT showcased its capacity to efficiently acquire new skills with minimal task-specific input, a phenomenon referred to as 'few-shot' learning. As the dimensions of GPT expanded, it exhibited the capacity to proficiently acquire knowledge in new tasks with less task-specific data, a phenomenon referred to as 'few-shot' learning (18).

This article shows how we can implement different use cases with customized inputs for chat-GPT NLP models. In the realm of artificial intelligence, the introduction of OpenAI's API endpoints for ChatGPT represents a significant advancement in democratizing access to sophisticated language models. By providing well-documented and accessible API endpoints, OpenAI has expanded the utility of ChatGPT beyond the confines of research laboratories and specialized applications to a broader audience, including developers, entrepreneurs, and educators. This shift facilitates a more extensive and diverse application of the technology, enabling the incorporation of advanced natural language processing capabilities into a variety of software applications and services. The API's accessibility encourages innovation and experimentation across various fields, fostering a collaborative environment where developers can tailor ChatGPT's capabilities to meet specific needs. This expansion is crucial in bridging the gap between cutting-edge AI research and practical, real-world applications, thereby accelerating the adoption and integration of AI into everyday technology solutions (19).

In academic discourse on artificial intelligence, the optimization of output in GPT for chat completions through context and parameters emerges as a pivotal subject. The efficacy of these language models, hinges significantly on their capacity to interpret and respond to user input within a given context. By meticulously setting parameters – including tone, style, and content specificity – users can significantly refine the model's responses to align more closely with the intended application. This adaptability is crucial in diverse fields ranging from education, where the model can be tuned to provide age-appropriate responses, to customer service, where it can be calibrated for empathetic and solution-oriented interactions. Furthermore, the incorporation of relevant context into prompts enables the model to generate responses that are not only syntactically and semantically accurate but also contextually appropriate, enhancing the overall user experience. These customizable features underscore the importance of fine-tuning AI models to specific use cases, thereby maximizing their potential and utility in real-world applications (19, 20).

The integration of OpenAI's API endpoints into various applications exemplifies a paradigm shift in the utilization of natural language processing (NLP) techniques, transcending traditional boundaries. By offering a streamlined and accessible gateway to advanced NLP capabilities, these endpoints enable a wide range of applications to harness the power of sophisticated language models like ChatGPT. This integration facilitates diverse functionalities, from semantic analysis and sentiment detection in social media platforms to automated customer support and personalized content creation in digital marketing. Moreover, the adaptability of the API allows for the tailoring of NLP features to specific industry needs, be it in legal text analysis, healthcare communication, or educational content generation. This democratization of NLP technology not only fosters innovation across sectors but also accelerates the development of more intuitive and human-centric AI interfaces. As a result, the OpenAI API endpoints serve as a catalyst, transforming the theoretical potential of NLP into practical, impactful solutions across an array of industries (21).

In conclusion, the article illustrates the significant impact of combining advanced GPT models with a Graphical User Interface to democratize the use of NLP tools. This integration not only makes complex language processing more accessible but also marks a pivotal step in the inclusive application of AI technology across various domains. The successful implementation of the GUI highlights the potential of AI in enhancing user interaction and broadening the scope of technology usage in everyday tasks.

Ethics Committee Approval: Ethics committee approval is not required in this study.

Peer-review: Externally peer-reviewed

Author Contributions: Concept: EK, ibç, Design: IBÇ, ZK, CÇ, Literature search: İBÇ, ZK, CY, CÇ, Data Collection and Processing: EK, İBÇ, CY, Analysis or Interpretation: İBÇ, Writing: EK, İBÇ, ZK, CÇ, CY

Conflict of Interest: The authors declared no conflict of interest.

Financial Disclosure: The authors declared that this study has not received no financial support.

REFERENCES

1. Han X, Zhang Z, Ding N, Gu Y, Liu X, Huo Y, et al. Pre-trained models: Past, present and future. AI Open. 2021; 2: 225-50.
2. Yenduri G, Srivastava G, Maddikunta PKR, Jhaveri RH, Wang W, Vasilakos AV, et al. Generative Pre-trained Transformer: A Comprehensive Review on Enabling

Technologies, Potential Applications, Emerging Challenges, and Future Directions. arXiv preprint arXiv:230510435. 2023.
3. Brockman G, Sutskever I. The OpenAI team,“. Introducing OpenAI. 2015; 11.
4. Dong L, Xu S, Xu B, editors. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP); 2018: IEEE.
5. Kim M, Corradini D, Sinha S, Orso A, Pasqua M, Tzoref-Brill R, et al., editors. Enhancing REST API Testing with NLP Techniques. Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis; 2023.
6. Ball CJ. Hacking APIs: Breaking Web Application Programming Interfaces: No Starch Press; 2022.
7. Hat R. What is a REST API? 2021. URL: https://www redhat com/en/topics/api/what-is-a-rest-api (visited on 08/06/2021).
8. Lin J, Pradeep R, Teofili T, Xian J. Vector search with OpenAI embeddings: Lucene is all you need. arXiv preprint arXiv:230814963. 2023.
9. Cheng SW, Chang CW, Chang WJ, Wang HW, Liang CS, Kishimoto T, et al. The Now and Future of ChatGPT and GPT in Psychiatry. Psychiatry and Clinical Neurosciences. 2023.
10. Lennon B. Machine translation: A tale of two cultures. A companion to translation studies. 2014; 133-46.
11. Bengio Y, Ducharme R, Vincent P. A neural probabilistic language model. Advances in neural information processing systems. 2000; 13.
12. Sutskever I, Vinyals O, Le QV. Sequence to sequence learning with neural networks. Advances in neural information processing systems. 2014; 27.
13. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:14090473. 2014.
14. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.
15. Józefowicz R, Vinyals O, Schuster M, Shazeer N, Wu Y. Exploring the limits of language modeling. arXiv [Preprint](2016). arXiv preprint arXiv:160202410.
16. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. Advances in neural information processing systems. 2017; 30.
17. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Advances in Neural

Information Processing Systems. 2022; 35: 27730-44.

18. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020; 33: 1877-901.
19. Kim TW. Application of artificial intelligence chatbot, including ChatGPT in education, scholarly work, programming, and content generation and its prospects: a narrative review. Journal of Educational Evaluation for Health Professions. 2023; 20: 38-45.
20. Dai D, Sun Y, Dong L, Hao Y, Ma S, Sui Z, et al., editors. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. Findings of the Association for Computational Linguistics: ACL 2023; 4005-4019.
21. Masood A, Hashmi A. Cognitive Computing Recipes: Artificial Intelligence Solutions Using Microsoft Cognitive Services and TensorFlow: Apress; 2019.