



# AI-Powered GPT-Based University-Specific Chat Assistant: UniROBO

Serdar Budak <sup>a</sup> , Serpil ASLAN <sup>a</sup> \* 

<sup>a</sup> Malatya Turgut Ozal University, Department of Software Engineering, Malatya Türkiye – 44210

\*Corresponding author

ARTICLE INFO		ABSTRACT
Received	19.11.2024	<p>AI-powered chat applications are innovative solutions that facilitate user interaction and information access. These applications improve user experience by providing personalized and context-sensitive responses thanks to large language models and natural language processing techniques. This study examined the design and development process of UniRobo, an AI-powered chat application developed for Malatya Turgut Ozal University students and staff. UniRobo is an application that provides instant information on topics such as education, food menus, and campus events and offers personalized responses using large language models and natural language processing techniques. In the development process, based on user needs analysis, the mobile application was created with React Native, the back-end was created with Python and FastAPI, and the MongoDB database was integrated. Artificial intelligence capabilities were supported by OpenAI API and fine-tuning, thus adapting to university-specific content. Retrieval-Augmented Generation (RAG) architecture and Azure AI Search technology increased user satisfaction by providing more accurate and faster responses. As a result, UniRobo has made university life more accessible, providing users with access to fast and accurate information, and has demonstrated the potential of artificial intelligence-based solutions in the education sector.</p>
Accepted	24.12.2024	
Doi: 10.46572/naturengs.1587879		
		<p><b>Keywords:</b> Chat Asistant, GPT, Retrieval-Augmented Generation, Azure AI</p>

## 1. Introduction

Large language models (LLMs) have emerged as state-of-the-art AI systems that process and generate text [1]. LLMs are an essential part of the developments in deep learning in recent years and have revolutionized natural language processing (NLP) applications [2]. These models are typically trained with huge parameters and attempt to understand language's complex structure and relationships by learning from large datasets. LLMs demonstrate high performance in a variety of NLP tasks, such as text understanding, machine translation, summarization, and dialogue systems [3].

In recent years, artificial intelligence technologies have played an essential role in developing conversational interfaces of chatbots [4]. Chatbots are conversational artificial intelligence agents that interact with users through natural language processing techniques, and with these features, they are classified within the broader concept of 'conversational user interfaces' [5]. This term, derived from the combination of the words "chat" and "bot" (robot), is also known by other names such as dialogue system, conversational agent, virtual assistant, and personal assistant. These systems are computer

programs designed to mimic human-like conversations in the internet environment and generally function as artificial intelligence-based software that produces responses according to specific scenarios [6]. These conversations can be in text or audio format. When a user asks a question, chatbots analyze the question using artificial intelligence algorithms and produce a logical answer [7].

In today's university environments, access to fast and accurate information for students and staff has become a critical necessity to improve the efficiency of educational processes and the flow of daily life. Especially in large and complex campus structures, accessing routine information such as educational programs, food menus, and campus events is often a time-consuming and challenging process. This can lead to time management problems and productivity losses for students and university staff.

Accessing the information needed from university websites is another challenge for users. Websites' complex content and structure can make it difficult to find the section where the information is sought, which causes users to spend more time. Artificial intelligence technologies and natural language processing (NLP)

\* Corresponding author. e-mail address: [serpil.aslan@ozal.edu.tr](mailto:serpil.aslan@ozal.edu.tr)  
ORCID : [0000-0001-8009-063X](https://orcid.org/0000-0001-8009-063X)

techniques offer significant potential to facilitate access to information and provide personalized answers to users. However, research on natural language processing technologies in Turkish has been limited, and existing solutions are generally focused on English. This makes it challenging to develop knowledge-based and user-friendly artificial intelligence applications in Turkish.

In this context, UniRobo, developed to meet the need for rapid access to information for students and staff at Malatya Turgut Ozal University, aims to respond to an essential need in this area. UniRobo provides instant and personalized information on education, food menus, and campus events using large language models (LLM) and natural language processing techniques. In addition, it quickly determines which section of the university website the information users need is located, facilitating access to information and reducing time loss.

Within the study's scope, the development of UniRobo has created an interactive chat platform that can provide fast and accurate answers to students' and staff's daily questions, making university life significantly easier. This application contributes to developing natural language processing technologies in the Turkish language, demonstrates the potential of artificial intelligence-based solutions in the education sector, and will increase user satisfaction and efficiency at Malatya Turgut Ozal University.

## 2. Related Works

Natural language processing (NLP) is a technology field developed in artificial intelligence and computer science to understand, analyze, and produce human language. NLP enables computers to interact with human language by analyzing the meaning, structure, and context of language. This field offers many applications, such as language modeling, text analysis, sentiment analysis, and machine translation [9]. Large language models are essential in understanding texts and producing context-based responses [10]. One of the biggest challenges of NLP is the versatility of language and context-dependent meaning changes. Therefore, the accuracy of NLP systems is increased using deep learning techniques and large data sets [11]. NLP enables the development of human-like interactions by processing text and voice data, which increases the effectiveness of chatbots, virtual assistants, and other artificial intelligence-based applications. However, developments in the field of NLP have mainly been carried out in the resource-rich English language, and the performance of models at a similar level in languages supported by fewer resources, such as Turkish, is generally lower [12]. When language models explicitly developed for Turkish are examined, BERTurk models developed by Schweter [3] stand out as usable and lightweight models. These models have been trained on Turkish texts based on the BERT architecture and have achieved successful results in various natural language understanding tasks. In addition, native language processing tools such as Zemberek significantly address the deficiencies in this

area by providing language resources and tools specialized for Turkish [14].

Developments in NLP have enabled chatbots to understand language, analyze context accurately, and produce meaningful responses. Recently, chatbots have become an essential digital interaction tool in many sectors [15]. They are used in various applications, from customer service to education and healthcare. Artificial intelligence-supported chatbots increase efficiency and optimize the workforce by answering users' questions quickly and accurately [16]. In addition, they improve user satisfaction and contribute to digital transformation processes by providing personalized experiences. These developments have transformed chatbots from simple tools that only provide automatic responses to more sophisticated and interactive systems [17].

Studies and research comparing and competing chatbots with human intelligence have been ongoing for many years. The first examples of such research in the historical process were simple text-based conversational programs such as ELIZA and PARRY in the 1960s [7]. Chatbots have emerged in different periods of the history of artificial intelligence, but each represents more advanced technologies than the previous one. Essential examples in the history of chatbots show how quickly chatbot technology has evolved from the past to the present and how exciting the future in this field can be [18].

## 3. Background

### 3.1. Transformer architecture

Transformer architecture is a structure that forms the basis of large language models widely used today [19]. Introduced in 2017, the Transformer can effectively model long-term dependencies using the attention mechanism [20]. This architecture allows language models to learn the relationships of each word in a sentence with other words, producing more meaningful and contextually rich outputs. In addition, Transformer architecture enables fast and efficient training on extensive data sets thanks to its parallel processing capabilities, significantly reducing the training time of large language models. Figure 1 represents the general structure of the Transformer architecture.

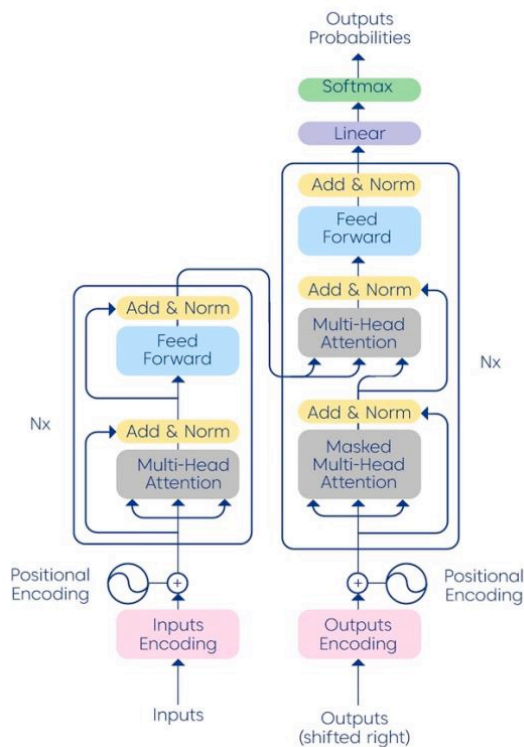


Figure 1. The transformer architecture

### 3.2. Retrieval-Augmented Generation (RAG) architecture

The Retrieval-Augmented Generation (RAG) architecture enables large language models to provide more accurate and detailed answers to knowledge-based questions [21]. RAG aims to increase the accuracy and reliability of the model's answers by combining the language model's retrieval and language generation capabilities. This architecture consists of two main components: the retriever and the language generation component. The retriever component collects relevant information from significant data sources or custom data sets and uses this information to validate and support the model's answers [22]. The language generation component produces contextually appropriate and accurate answers using the retrieved data. The retriever component usually retrieves data from pre-trained knowledge bases or search engines, and this data provides the contextual information

necessary for the model to generate its answers. The language generation component provides meaningful and accurate answers to users using the information obtained. Thanks to integrating these two components, the RAG architecture enables language models to perform more than just text generation, producing knowledge-based and more reliable answers.

The RAG architecture offers significant advantages, especially in knowledge-intensive areas and applications that require specific knowledge. For example, it is highly effective for providing accurate information on academic articles, technical documents, or specialized topics. This approach provides the language model access to a wide range of knowledge sources, significantly increasing the accuracy and reliability of the responses produced. As a result, the RAG architecture improves the capacity of language models to produce knowledge-based and contextual responses. This technology enables language models to provide more intelligent and appropriate responses to user needs, improving user experience and satisfaction.

### 3.3. Azure AI search

Azure AI Search [23] is a service capable of performing fast and accurate searches on large data sets. This technology optimizes the search experience by providing users with the most relevant information quickly and effectively. Azure AI Search provides users instant access to the necessary information through its integration with language models. This integration significantly improves user satisfaction by improving UniRobo's performance.

### 3.4. React native and mobile App development

React Native is a technology that makes it easy to create native applications for both iOS and Android platforms, enabling cross-platform development with a single codebase. This feature allows for faster and more cost-effective application development, providing a consistent experience for users across different mobile platforms. Figure 2 represents the general flow diagram of the React Native architecture. UniRobo's mobile application is developed using the React Native framework.

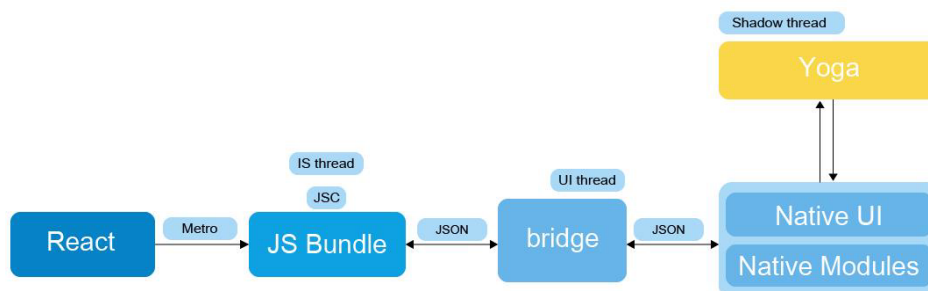


Figure 2. The React Native architecture

## 4. The Proposed Chat Assistant: UniRobo

The first step in developing UniRobo was a comprehensive analysis of user needs. Data on the information most needed by students and staff were collected through surveys and user interviews conducted

at Malatya Turgut Ozal University. During this analysis process, users' demands for fast and easy access to educational information, food menus, campus events, and other daily information were prioritized. The obtained data was essential to UniRobo's design and development processes. Figure 3 represents the flow diagram of the proposed UniRobo model based on the RAG architecture.

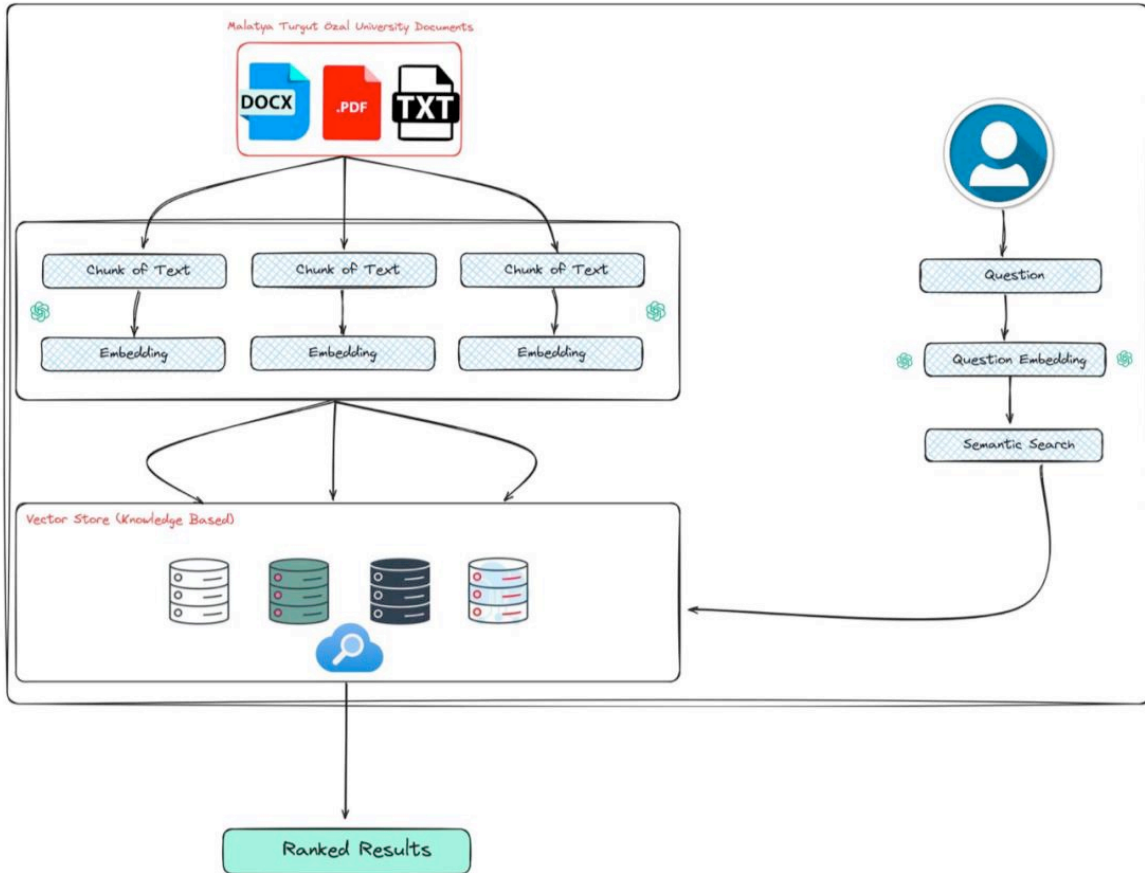


Figure 3. The RAG-based UniRobo general flow diagram

UniRobo is an AI-powered chat application developed using modern technologies:

**Mobile Application Development:** The application is developed using the React Native framework. This makes the application compatible with iOS and Android platforms, allowing cross-platform development. React Native enables faster and more cost-effective application development while providing a seamless user experience across devices.

**Back-end Development:** The back-end part of the application is built using the Python programming language and the FastAPI framework. FastAPI provides an ideal solution for creating high-performance APIs while providing a robust and scalable infrastructure combined with Python's flexibility. MongoDB is preferred for data management, as it enables secure storage and fast access to data.

**AI Integration:** UniRobo's AI component has been integrated with OpenAI's API. In addition, the model has been fine-tuned to make it more compatible with university-specific content. These processes allow the

model to provide more precise and accurate answers, improving user experience and application accuracy.

UniRobo is powered by RAG (Retrieval-Augmented Generation) architecture and Azure AI Search technology to provide users with more accurate and reliable answers. RAG enables large language models to generate informed and contextual answers by pulling data from external information sources. This architecture consists of two main components: the retriever and the language generation component. The retriever component retrieves the necessary information from significant data sources or custom data sets. In contrast, the language generation component uses this data to provide meaningful and accurate answers to the user. This integration increases the accuracy and reliability of UniRobo's answers while providing users with more comprehensive and accurate information. Azure AI Search is an advanced service that performs fast, accurate searches on large data sets. This technology provides users instant access to the information they need and increases UniRobo's performance, enhancing user satisfaction. Azure AI Search optimizes UniRobo's

search functions and quickly provides users with the most relevant information, providing a more efficient

experience. Figure 4 represents UniRobo's Azure Ai Search example.

Figure 4. UniRobo's Azure Ai Search example

#### 4.1. Fine tuning process

For UniRobo to provide more compatible and accurate answers to university-specific content, a fine-tuning process was performed on the model using custom instructions. The fine-tuning process means retraining a previously trained large language model to make it more suitable for a specific task or context. At this stage, the model's learning process is enriched with customized data in line with the university's needs, and the model's outputs are shaped based on the specific content within the university. This fine-tuning process using custom instructions increases the model's linguistic accuracy. It enables it to produce more compatible and original answers with the university's information in educational information, food menus, and campus events. Thus, it has become possible for UniRobo to provide its users with more meaningful, contextually rich, and accurate answers based on university-specific content. This process allows the user to access the information they are looking for faster while also standing out as a feature that increases the model's reliability and user satisfaction.

##### 4.1.1. Determining user instructions

The first step is to determine specific instructions for how the model should respond to the user. These instructions are designed to increase the accuracy and consistency of the model's responses. Here are some sample instructions:

- Only respond in Turkish.
- Focus on the requested information when responding; do not provide additional information.
- Approach the user in a friendly manner.
- Ask the user to ask the question more meaningfully in response to irrelevant or poorly formatted questions.
- Only respond to university-related questions; do not consider university information when responding to independent questions.

The above instructions are integrated into the model's response process. Below is a sample code that shows how the instructions are used in the model's response process, as shown in Figure 5.



```

def process_results_with_chatgpt(search_result: dict, user_prompt: str) → str:
    system_prompt = {
        "role": "system",
        "content": (
            "You are a Malatya Turgut Özal Üniversitesi assistant. Use the following details to assist the student:\n"
            f"{search_result}\n"
            "Only speak in Turkish language.\n"
            "If you know the answer, send me only the result as an answer and don't write me anything else.\n"
            "Never ask the user additional questions.\n"
            "Your name is UniRobo and you are a virtual assistant.\n"
            "If the question is meaningless and poorly formatted, ask the user to ask the question in a meaningful way.\n"
            "Understand the information and present it well.\n"
            "Never give an answer to keep the conversation going.\n"
            "Talk to him in such a way that he feels like a friend.\n"
            "If you are asked something independent of the school, answer it independently, ignoring the information you have received.\n"
            "If the information received is not relevant to the question asked, respond with something like Unfortunately, I could not find the information you asked for, you can search for detailed information on our university website and never mention that you received the information."
        )
    }
    user_message = {
        "role": "user",
        "content": user_prompt
    }

    completion = openai.ChatCompletion.create(
        model="gpt-4",
        messages=[system_prompt, user_message]
    )

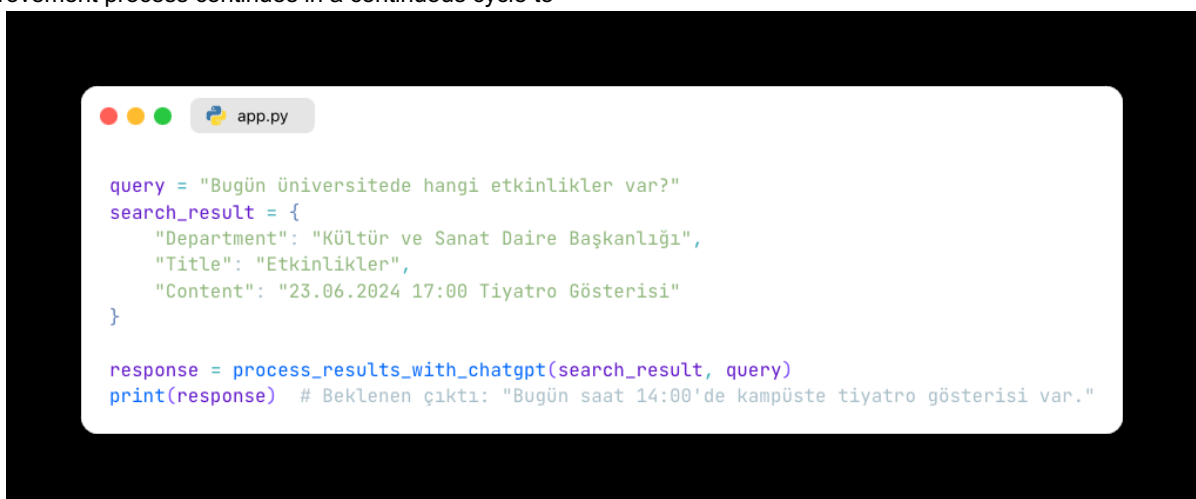
    return completion.choices[0].message.content

```

**Figure 5.** Fine Tuning sample instruction structure

Once the instructions are integrated, rigorous testing of the model's responses is necessary. This phase is critical to assess how well UniRobo responds to user needs and how effectively it responds. The responses to users' various questions are systematically reviewed and analyzed to show that the specific instructions identified are being implemented correctly. After evaluating each response's accuracy, contextual relevance, and consistency, improvements are made if necessary. This improvement process continues in a continuous cycle to

optimize the model's performance and the user experience. The instructions identified and tested are activated within the application, aiming to provide accurate and reliable responses for every user interaction. Below, an example usage scenario of how UniRobo responds to a user question is presented. This scenario provides a concrete example of how the system functions and is customized to user needs, as shown in Figure 6.



```

query = "Bugün Üniversitede hangi etkinlikler var?"
search_result = {
    "Department": "Kültür ve Sanat Daire Başkanlığı",
    "Title": "Etkinlikler",
    "Content": "23.06.2024 17:00 Tiyatro Gösterisi"
}

response = process_results_with_chatgpt(search_result, query)
print(response) # Beklenen çıktı: "Bugün saat 14:00'de kampüste tiyatro gösterisi var."

```

**Figure 6.** Instructions usage scenario

This process optimizes the model according to specific guidelines, enabling users to get more contextually accurate and consistent answers that are more relevant to their needs. Fine-tuning aligns the model's pre-trained structures to university-specific content and user

interactions. This way, users access the correct information and interact with more personalized and meaningful answers. Fine-tuning also improves the model's performance, improving response accuracy, speed, and reliability. This improved accuracy and

consistency increases UniRobo's user satisfaction, as users can access the information they need with less effort. This maximizes UniRobo's overall effectiveness, strengthening the user experience and the application's long-term success.

## 5. Conclusions

In this study, the design and development of UniRobo, an artificial intelligence-supported chat application developed for students and staff of Malatya Turgut Ozal University, is discussed in detail. UniRobo provides fast and accurate information to its users by providing instant and personalized answers on various topics such as educational content, food menus, and campus events. Within the study's scope, university users' needs were analyzed in detail, and the functionality and efficiency of the application were increased. With the integration of modern technologies, the fine-tuning process performed through special instructions ensures the model's adaptation to university-specific content and increases the accuracy of the answers. UniRobo offers an interactive platform that can effectively respond to the daily questions of students and staff by utilizing large language models and natural language processing (NLP) techniques. The use of RAG architecture and Azure AI Search technologies has improved the user experience by increasing the accuracy and reliability of the answers. In addition, modern software technologies such as React Native, Python, FastAPI, and MongoDB have optimized the performance of the application and ensured cross-platform compatibility. As a result, UniRobo has facilitated the education and daily life processes at Malatya Turgut Ozal University, enabling students and staff to access information quickly and accurately. This study has significantly contributed to developing natural language processing technologies in Turkey. It has revealed the potential of artificial intelligence-based solutions in the education sector.

## References

- [1] Arcas, B. A. (2022). Do large language models understand us?. *Daedalus*, 151(2), 183-197.
- [2] Zhang, J., Krishna, R., Awadallah, A. H., & Wang, C. (2023). Ecoassistant: Using llm assistant more affordably and accurately. *arXiv preprint arXiv:2310.03046*.
- [3] Wu, C., Lin, Z., Fang, W., & Huang, Y. (2023, October). A medical diagnostic assistant based on llm. In *China Health Information Processing Conference* (pp. 135-147). Singapore: Springer Nature Singapore.
- [4] Guan, Y., Wang, D., Chu, Z., Wang, S., Ni, F., Song, R., & Zhuang, C. (2024, August). Intelligent agents with llm-based process automation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 5018-5027).
- [5] Luo, B., Lau, R. Y., Li, C., & Si, Y. W. (2022). A critical review of state-of-the-art chatbot designs and applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 12(1), e1434.
- [6] Wei, R., Li, K., & Lan, J. (2024, March). Improving Collaborative Learning Performance Based on LLM Virtual Assistant. In *2024 13th International Conference on Educational and Information Technology (ICEIT)* (pp. 1-6). IEEE.
- [7] Adamopoulou, E., & Moussiades, L. (2020). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373-383). Springer, Cham.
- [8] Kim, J. K., Chua, M., Rickard, M., & Lorenzo, A. (2023). ChatGPT and large language model (LLM) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine. *Journal of Pediatric Urology*, 19(5), 598-604.
- [9] Fanni, S. C., Febi, M., Aghakhanyan, G., & Neri, E. (2023). Natural language processing. In *Introduction to Artificial Intelligence* (pp. 87-99). Cham: Springer International Publishing.
- [10] Bharadiya, J. (2023). A comprehensive survey of deep learning techniques natural language processing. *European Journal of Technology*, 7(1), 58-66.
- [11] Wang, Y., Sun, Y., Fu, Y., Zhu, D., & Tian, Z. (2024). Spectrum-BERT: pre-training of deep bidirectional transformers for spectral classification of Chinese liquors. *IEEE Transactions on Instrumentation and Measurement*.
- [12] Çöltekin, Ç., Doğruöz, A. S., & Çetinoğlu, Ö. (2023). Resources for Turkish natural language processing: A critical survey. *Language Resources and Evaluation*, 57(1), 449-488.
- [13] Schweter, S. (2020). BERTurk-BERT models for Turkish. *Zenodo*, 2020, 3770924.
- [14] Koçak, S., İç, Y. T., Sert, M., & Dengiz, B. (2023). Ar-Ge projelerinin sınıflandırılması için doğal Türkçe dil işleme tabanlı yöntem. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, 38(3), 1375-1388.
- [15] Cao, L. (2023). Diaggpt: An llm-based chatbot with automatic topic management for task-oriented dialogue. *arXiv preprint arXiv:2308.08043*.
- [16] Gunawan, J. (2023). Exploring the future of nursing: Insights from the ChatGPT model. *Belitung Nursing Journal*, 9(1), 1.
- [17] Deng, X., & Yu, Z. (2023). A meta-analysis and systematic review of the effect of chatbot technology use in sustainable education. *Sustainability*, 15(4), 2940.
- [18] King, M. R. (2023). The future of AI in medicine: a perspective from a Chatbot. *Annals of Biomedical Engineering*, 51(2), 291-295.
- [19] Li, Y., Cao, J., Xu, Y., Zhu, L., & Dong, Z. Y. (2024). Deep learning based on Transformer architecture for power system short-term voltage stability assessment with class imbalance. *Renewable and Sustainable Energy Reviews*, 189, 113913.
- [20] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
- [21] Ardic, O., Ozturk, M. U., Demirtas, I., & Arslan, S. (2024, May). Information Extraction from Sustainability Reports in Turkish through RAG Approach. In *2024 32nd Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
- [22] Wiratunga, N., Abeyratne, R., Jayawardena, L., Martin, K., Massie, S., Nkisi-Orji, I., ... & Fleisch, B. (2024, June). CBR-RAG: case-based reasoning for retrieval augmented generation in LLMs for legal question answering. In *International Conference on Case-Based Reasoning* (pp. 445-460). Cham: Springer Nature Switzerland.
- [23] Fukizi, K. Y. (2023, August). Collaborative Decision-Making Assistant for Healthcare Professionals: A Human-Centered AI Prototype Powered by Azure Open AI. In *Proceedings of the 6th ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies* (pp. 118-119).