Applying Natural Language Processing and Machine Learning to Develop an Interactive Sports Almanac Chatbot for the Premier League of Bosnia and Herzegovina

Nedim Bandžović
Faculty of Engineering, Natural and Medical Sciences
International Burch University
Sarajevo, Bosnia and Herzegovina
nedim.bandzovic@stu.ibu.edu.ba

Dželila Mehanović
Faculty of Engineering, Natural and Medical Sciences
International Burch University
Sarajevo, Bosnia and Herzegovina
dzelila.mehanovic@ibu.edu.ba

***Abstract*—This paper concentrates on the analysis of collected different types of data as well as processing the specific data by using different AI-related technologies. The result of this research allows the reader to learn about the most important characteristics that are related to building an interactive sports almanac chatbot whose main goal is to preserve the football history of Bosnia and Herzegovina.**

Keywords—chatbot, GPT4-Turbo, machine learning, natural language processing (NLP), FAISS, all-MiniLM-L6-v2

I. Introduction

Usage of natural language processing and machine learning techniques in different fields have made our everyday lives look impossible without using them. There are numerous fields, starting from education, law, health etc. These tools are mostly embedded in the form of AI models that allow users to generate a specific result based on a user-generated prompt. For example, one of the famous AI tools - ChatGPT can do a variety of different tasks, based on a simple (one-sentenced) prompt, starting from working on with different types of mathematical formulas and functions, performing generation of images and videos to fixing issues in important documents. The major advantage of ChatGPT is because users are able to interact with AI through a chatbot-like tool. This approach allows many AI enthusiasts to start thinking about development of specific projects that can be useful and besides the usability segment, fun. This is where this research steps in.

One of the most popular and important events in Bosnia and Herzegovina is football. Bosnia and Herzegovina is a place where many great football events have occurred and its football history is very rich which forces football enthusiasts to take specific steps in order to preserve it. The football history in the country has been written since the Austro-Hungarian period. From then, many great results have been accomplished. Some of these events include the first appearance of the Bosnian national football team on a World Cup in 2014, a Bosnian (Yugoslav back in the day) club 'FK Željezničar' reaching the semi finals of the UEFA Cup in 1985 as well as clubs like Manchester United and Arsenal playing against 'FK Sarajevo' and 'FK Željezničar' in Sarajevo in the 1960s and 1970s. However, one of fan can find out more information about a specific club, player or an event that occurred in Bosnia and Herzegovina. Even though there is the official website of the Football Federation of Bosnia and Herzegovina as well as Wikipedia / UEFA / FIFA articles and reports, there is no centralized platform where in one place, a fan can easily find specific information related to football. This project explores the application of these technologies to create an interactive chatbot that functions as a sports almanac for the Premier League of Bosnia and Herzegovina. Through the integration of historical match data, player profiles, and season statistics, the chatbot will utilize machine learning techniques to parse user queries, retrieve relevant information, and provide conversational responses. The goal is to enhance user engagement with Bosnian football history by offering an accessible and intuitive query interface.

Research questions which this paper aims to answer differ in terms of their execution and characteristics and their common goal is to assist in creation of the system that will be able to take some form of an input and then generate a final value. The questions are as follow:

- Research question 1: What are the ways in which natural language processing (NLP) can assist us in preparing raw data that will be later used for training purposes?
- Research question 2: What machine models are most efficient in working with unstructured data, especially the one that will be used for this project?

The first research question which was mentioned will be answered to in the form of a case study and it includes explanation how from unstructured data which will be fetched in form of raw text, we can build a foundation block for a model that will be relied on by the chatbot, which is the final result of this research.

The second research question is related to figuring out which of the machine learning models will be appropriate for dealing with content generated from unstructured data, which in this specific case will be fetched directly from Wikipedia and Wikidata.

II. Literature Review

The [1] is a project where the author developed a high-tech chatbot using the MERN stack (MongoDB, Express.js, React.js, Node.js) and has applied different natural language processing (NLP) methods for data filtering and formatting.

The foundation of the developed high-tech chatbot lies in the OpenAI API, which allowed the author's chatbot to achieve a high level of fluency, context and interaction between the user and the bot itself. The chatbot was delivered to a small group of test users and the research also contains information about the model's accuracy, user satisfaction as well as the general system's scalability.

On the other hand, [2] is a mental health support chatbot. Before the most important of the research, which is the chatbot itself, the author emphasizes 'high-pressure workplaces, stress, anxiety and depression' to be on the rise. The chatbot that was developed is using advanced machine learning and natural language processing methods as well as OpenCV and LLMs (Langchain-integrated large language models). The chatbot works in the way where it is interpreting text prompts as well as facial expressions by using OpenCV and is trying to create a human-like, sympathetic response. Also, one of the advantages of this research is that by using LLM, the chatbot creates responses in real-time, without the need for training the model on a static dataset, like it is being done in conventional models. Also, the chatbot is embedded into a user-friendly interface, which makes interactions with it a lot simpler. The chatbot is 24/7 available to the end users.

The authors of [3] also developed a high-intelligence chatbot named 'Aros', that is able to run independently on low-value devices (authors mentioned Raspberry Pi 5 as the best example). The chatbot was developed by using a neural network which had around 1.5 billion-parameters and the dataset it was trained on had three trillion different tokens. As in [2], the authors had developed a specialized interface which allows users to use the chatbot in a more intuitive way. The users do not need internet connection in order to use the chatbot, instead, the chatbot is also available offline.

In [4], work is related to developing a chatbot that will be able to perform cascade genetic testing. Cascade genetic testing, as mentioned by the authors, is a process which offers genetic testing to blood relatives of individuals with known pathogenic mutations. Easy identification of cancer-associated pathogenic mutations can reduce incidence and mortality of cancer. In order to simplify this process, the authors developed the Cascade Chatbot, which allowed families to perform genetic-testing via specific gene-specific educational modules. In a gynecologic oncology clinic, 100 patients with pathogenic mutations in high-risk genes (e.g., BRCA1/2, MLH1, etc.) were offered the chatbot. Of the 59 patients with eligible at-risk relatives, 98% accepted the chatbot, and 76% shared it with family within two weeks. These results suggest high acceptability and potential effectiveness of chatbots in supporting cascade genetic testing, reducing the communication burden on patients, and potentially improving testing uptake among relatives.

The [5] is a research in which the authors tend to develop a smart banking chatbot. The chatbot was developed using the RASA framework which is used for creation of virtual assistant and contextual chatbots and was embodied with natural language processing (NLP) methods in order for the chatbot to easily understand user-generated prompts. The chatbot is able to check account balances, make transactions and address specific banking issues without the need for the user to contact the human-based customer service. The authors, in order to confirm that their work was successful, used different machine learning techniques (Random Forest Classification, k-Nearest neighbours, Support Vector Machines) in order to test and evaluate the accuracy of their chatbot which was, as mentioned, high.

In [6], the author has built a chatbot-based ticketing system for a museum. The chatbot is powered with TensorFlow and large language models (LLMs). The system itself includes QR-based ticketing for easy entrance to the museum, heatmaps which allow the chatbot to manage visitor flow as well as a specific SMS system which allows the user to book his ticket for the museum, reducing the need for the user to go to the museum or have any type of interaction with staff inside of the museum. What is also included in the main functionalities is the ability of the chatbot to remember previous preferences of the user and based on them, offer personalized recommendations (schedule optimization, personalized content delivery etc-) as well as to recognize the language of the user allowing it to behave in a multilingual manner. Besides using TensorFlow and LLMs, the author also used Flutter, Firebase and Go for smooth and efficient integration of all of the technologies empowered in this research.

Authors of [7] built a chatbot which assists students in choosing the right engineering college after EAMCET counselling (entrance exam) in Telangana, one of the Indian states. The chatbot tends to suggest the right engineering college based on the student's EAMCET rank, preferred courses and location. In order to do that, the authors used different natural language processing (NLP) techniques in order for the chatbot to easily understand user queries as well as machine learning techniques that were used to train a model on a dataset (official EAMCET dataset that includes information about colleges like name, branches, locations etc.) that contains the list of all suitable colleges in Telangana. The chatbot was built in Python and is publicly available as a web application thanks to the Flask framework, which allows easy deployment of Python applications.

The author of [8] developed a special chatbot named 'CollegeConnect AI'. The author emphasizes the fact that students can often face challenges in terms of accessing information that is related to college procedures and activities. This is why the author developed a chatbot that serves as a virtual assistant for college students through which students can easily get information about admission processes, examination schedules, student clubs, canteen services as well as campus events. The system was developed using Python and the Django framework for backend issues while the frontend interface was developed using the common technologies and approaches like HTML, CSS and JavaScript.

Similar to [8], authors of [9] developed a chatbot named 'JavaTutorBot'. 'JavaTutorBot' is a chatbot that has been developed to guide students while they are writing their first basic Java programs. The chatbot covers different Java-related subjects, including variables, data types, arithmetic operations and the general structure of a Java program. The chatbot is available to students via a friendly user interface, which besides communication with the chatbot itself, also provides information related to the progress of the student. The chatbot is also embedded with a Java editor and online compiler, where students can pass code into it and check its correctness. Authors also mentioned that the 'JavaTutorBot' was part of a pilot evaluation in the May 2024 semester, where access to the bot was given to two different student groups in order to see whether the chatbot was effective or

not. The results indicate that students found the chatbot to be highly beneficial due to the fact that they felt that someone was there to support them while they were learning and solving Java related problems.

In [10], a chatbot whose aim is to assist breast cancer patients was developed. The author emphasizes the fact that these patients lack access to timely information and general emotional support. In order to assist these patients, the author developed an OpenAI backed chatbot which provides simplified explanation of the patient's diagnosis, treatment options as well as post-treatment care based on standard adopted in international oncology standards. Besides the official medical support, the chatbot has different tools for stress management where one of the approaches of the chatbot is to provide motivational messages and empathetic responses. In a pilot study involving 50 breast cancer patients from underserved areas, the chatbot was evaluated for usability, satisfaction, and emotional support. Results showed high user satisfaction (average score 4.7/5), with 85% of patients reporting better understanding of their condition and 70% experiencing reduced anxiety.

The author of [11] developed an asthma care chatbot. The chatbot is based on a machine learning model which was trained on a dataset that contains information related to asthma severity in order to predict the severity of symptoms. The chatbot has a user-friendly interface through which users can pass specific data which includes their symptoms as well as demographic and other personal information. The paper also includes metrics that were used in order to evaluate the accuracy of the model, and some of them include accuracy, precision, recall and F1 score.

In [12], a social chatbot named *Chatbol* was developed to handle user queries related to the Spanish football league. Designed as a Slack-based application, the chatbot processes text-based interactions and focuses on providing information about players, teams, coaches, and match fixtures. It includes a natural language understanding (NLU) module trained to detect user intents and extract relevant entities. These entities are dynamically queried from Wikidata using SPARQL, allowing real-time, up-to-date responses. In cases where the NLU module cannot provide a confident answer, a fallback retrieval-based system is used. This system leverages a database of football-related dialogues gathered from IRC football chat logs and movie subtitles via the OpenSubtitles database, enabling broader and more flexible conversational coverage within the football domain.

III. Methods and Materials

A. Dataset Overview

For the purpose of this paper, whose main goal is to create a chatbot that is focused on the Premier League of Bosnia and Herzegovina, we will not have a typical dataset as it is being used in the majority of different machine learning and natural language processing projects. The truth is that there is no actual dataset that can be downloaded and used for model training. Instead, the dataset will be derived from Wikipedia and Wikidata. The dataset will be organized in a form of two separate .json files. The first file, with the name of 'clubs_data.json' will contain narrative summaries and general metadata for each club in Bosnia and Herzegovina.

The second file that will be generated is "facts.json". This file will include factual statements derived from the 'summary' field in the 'clubs_data.json' and will be used for semantic search and retrieval that will be applied by the chatbot. The goal of this file is to have self-contained, objective and specific facts that are related to each club.

The first step that will be taken is that a list of clubs participating in the Premier League of Bosnia and Herzegovina will be generated. In order to simplify the project, the list will contain all of the Bosnian clubs that had participated in the league since 2019. After the list generation has been completed, the dataset will then be filled with common data (clubs_data.json) related to a specific club by retrieving data through Wikipedia API. After filling the first file, the process will then move to "facts.json", where through using Wikidata SPARQL Endpoints, we will be able to retrieve structured properties and statements. After that, multiple machine learning (ML) and (NLP) methods will be taken in order to prepare the collected data for human-like behaviour when it comes to user-related interaction.

B. Techniques used

The creation of the chatbot follows four structured steps. First, the project initiates by gathering and preparing domain information. The Premier League of Bosnia and Herzegovina clubs list is curated and, for each of them, initial descriptions are retrieved using the Wikipedia API. The descriptions, if available in Bosnian and secondarily in English, are written into a formatted file (clubs_data.json). To supplement this information, more formalized facts such as stadium names, city locations, titles achieved, and foundation years are retrieved by executing SPARQL queries against the Wikidata endpoint. These collected overviews and SPARQL results are normalized and rewritten as concise, factual statements and are aggregated to form the chatbot's knowledge base in facts.json.

Stage 2 involves that all factual sentences are embedded with the all-MiniLM-L6-v2 model of Sentence-Transformers library, projecting words into high-dimensional semantic space. The embeddings are stored in Facebook's FAISS library to speed up similarity-based retrieval. The FAISS index and accompanying metadata are stored as embeddings.index and embeddings.pkl.

The third stage consists of building a light-weight command-line chatbot interface using Python. Whenever a query is submitted by a user, the query is inserted into the same transformer model and compared against the indexed facts via cosine similarity. The top-k most relevant facts are retrieved and passed along with the original query to a language model such as GPT-4-turbo or a locally run LLaMA model. The model returns a fact-grounded, context-sensitive answer, which is output to the terminal.

Finally, the system is evaluated with a set of example football-related questions to ensure its correctness. This loop keeps the chatbot accurately up-to-date and continuously enhances its ability.

The system employs the all-MiniLM-L6-v2 model from the Sentence-Transformers library to convert user queries as well as statements of fact into dense vector representations for semantic lookup. The model is based on Microsoft's MiniLM architecture, an efficient transformer with the aim to attain much of BERT's performance while dramatically reducing computational requirements [13]. all-MiniLM-L6-v2 has approximately 22 million parameters and outputs 384-

dimensional embeddings, making it optimal for low-latency use cases such as document similarity, semantic search, and information retrieval. It was fine-tuned with a contrastive learning objective on large-scale Natural Language Inference (NLI) and paraphrase pairs and can thus map semantically close text to nearby points in vector space. This feature allows the chatbot to retrieve the most precise facts by carrying out cosine similarity comparison between the fact vectors precomputed and embedded user question. Sentence-Transformers models are simple to implement and very efficient, and all-MiniLM-L6-v2 is particularly suited for CPU-based platforms and applications with lower computational complexity [14] . Its compact size and high performance make it a suitable and practical solution for semantic search in this project.

To generate fluent, context-dependent responses from the facts retrieved, the project employs GPT-4-turbo, a highly performing large language model developed by OpenAI. GPT-4-turbo is an optimized variant of GPT-4 with faster inference and lower computational cost, without compromising the language understanding and generation of the original model. It is based on the transformer architecture, with dense self-attention layers and large-scale pretraining over a broad corpus of publicly available and licensed data. GPT-4-turbo, as a generative model, performs particularly well when paired with a retrieval-based pipeline, where retrieved facts are used to ground its output and constrain hallucinations. OpenAI did not disclose the exact architecture, model size, or training data for GPT-4 or GPT-4-turbo publicly, but both performance metrics and public usage confirm its elevated accuracy, coherence, and contextual understanding on a wide range of tasks [15]. In this project, GPT-4-turbo is used in a prompt-engineering paradigm, where it accepts the user's question and the top-k retrieved facts as input and produces a natural language response that includes the factual content. This retrieve-and-generate paradigm—commonly referred to as Retrieval-Augmented Generation (RAG) [16]—enables responses to be both factually accurate and linguistically fluent [17].

IV. Results

The result of this research is a functional chatbot that can be easily launched and interacted with. Even though in this stage of the project there is no specific user interface which can be empowered in the form of a web application, by simply using the Python console, the user can easily interact with the chatbot and immediately get the response needed.

Due to the fact that this research is dealing with a chatbot, there are different techniques that can be used in order to determine the validity of data that the chatbot is returning after we send some user input to it. However, the best approach that can be taken in order to test the chatbot is to give specific prompts to it and then to analyze its answers. This will be done by having a questionnaire which is consisted of a number questions about different clubs that are currently participating in the Premier League of Bosnia and Herzegovina, in order to determine and check how our chatbot is dealing with different types of information that is related to different clubs (not only related to the Bosnian Big Four - Željezničar, Sarajevo, Zrinjski and Borac) but also to smaller clubs who were or are currently participating, like Igman Konjic or HŠK Posušje.

TABLE I. RESULTS PROVIDED BY THE CHATBOT

QUESTION	ANSWER
What is the name of the stadium where FK Željezničar is playing its games?	Stadion Grbavica
Is Igman Konjic currently playing in the Premier League of Bosnia and Herzegovina?	No, Igman Konjic is not currently playing in the Premier League of Bosnia and Herzegovina. They play in the First League of the Federation of Bosnia and Herzegovina.
When was FK Sarajevo established and how?	FK Sarajevo was established on 24 October 1946, as the result of a merger between local Sarajevo football clubs Udarnik (Vanguard) and Sloboda (Liberty). The meeting to form the club was held in the main hall of the FIS building.
What is the full name of HŠK Zrinjski Mostar?	The full name of HŠK Zrinjski Mostar is Hrvatski športski klub Zrinjski Mostar.
What are the biggest rivalries in the league?	The biggest rivalries in the league are the Mostar derby between the two Mostar teams and the Sarajevo derby between FK Sarajevo and their biggest rival.
What is the name of the supporters of FK Velež?	The supporters of FK Velež are called the Red Army Mostar.
What is the capacity of Stadium Grbavica?	The capacity of Stadium Grbavica is 30,121.
What is the full name of the stadium where FK Sarajevo is playing?	The full name of the stadium where FK Sarajevo plays is Stadion "Asim Ferhatović Hase".
When did FK Željezničar win its first title? What happened to FK Velež in Bosnian war?	FK Željezničar won its first title in the 1971–72 season. During the Bosnian War, FK Velež was affected by the conflict and the political changes in Bosnia and Herzegovina. As Bosnia voted to leave Yugoslavia in an independence referendum, Velež, which was part of the Yugoslav First League, found itself on the way out of the league. The war and the subsequent independence of Bosnia led to significant changes in the football landscape, including the refounding of clubs and shifts in club identities and support bases. After the war, Velež became a symbol of the Bosniak side of Mostar, representing the east side of the city.
Who played the Marshal	Željezničar and Velež Mostar played in the Marshal Tito Cup final in the 1980-1981 season.

Tito Cup in the season 1980-1981?	
Who was the winner of the Yugoslav cup in 1988?	The winner of the Yugoslav Cup in 1988 was Borac.
Who was the manager when Borac won the Cup in 1988?	The manager when Borac won the Cup in 1988 was Husnija Fazlić.

In summary, as visible in Table 1, the chatbot demonstrates a strong grasp of factual knowledge related to football in Bosnia and Herzegovina, particularly concerning clubs like FK Željezničar, FK Sarajevo, HŠK Zrinjski Mostar , FK Velež and others. It provides accurate and contextually relevant answers, correctly identifying stadium names, club histories, rivalries, and historical cup events. The responses show coherence, reliability, and a clear alignment with verified sources, making the chatbot a potentially valuable tool for users seeking concise and trustworthy information on the Premier League of Bosnia and Herzegovina. However, it is notable that it is making mistakes in terms of numerical data, which means that more training steps should be included and implemented.

V. Conclusion

This research set out to build a chatbot capable of providing clear, fact-based information about football clubs in the Premier League of Bosnia and Herzegovina.

At its core, this project uses the power of natural language processing and machine learning to simulate conversation while returning accurate and relevant answers. Although still in a foundational phase without a web-based user interface, the result is a functional chatbot that can be easily interacted with through a Python console. Users can pose direct questions and receive immediate answers, making the system both accessible and practical for testing and feedback. The evaluation of the chatbot was guided by a structured set of questions covering a wide range of clubs—from the Bosnian Big Four (Željezničar, Sarajevo, Zrinjski, and Borac) to less prominent teams. Each response was assessed for factual accuracy and supported with references where available. The findings indicate that the chatbot demonstrates a strong command of historical and contextual football knowledge, effectively returning reliable information about stadiums, club origins, rivalries, and key sporting events. Nonetheless, some minor inaccuracies (particularly in numerical data) highlight areas for further improvement. The main limitation of this project is the actual data that is being collected. In order to avoid any type of language miscommunication and lack of concurrency, the main source that was being used was Wikipedia with English-written articles. However, when we talk about the Premier League of Bosnia and Herzegovina, it is quite understandable that articles related to the Bosnian Big Four contain more data in comparison to articles that are related to minor clubs. One of the reasons is that there are fans and fan bases that are regularly updating Wikipedia articles for their clubs. Also, an option where club officials are also modifying and updating their Wikipedia articles is also not turned off.

In regards to the first research question, it was shown that Natural language processing (NLP) plays a key role in preparing raw data for this project by cleaning and normalizing text from Wikipedia articles, extracting structured facts from narrative summaries, handling language differences between Bosnian and English content, and generating vector embeddings for semantic similarity search using models like MiniLM. It also assists in identifying and linking entities such as club names, stadiums, and historical events to ensure consistent and accurate data representation.

The second research question was also proven that abovementioned previously empowered models work efficiently with this unstructured data; the project uses two main models. The first is all-MiniLM-L6-v2, a lightweight and fast transformer model that creates dense vector embeddings for semantic retrieval, making it ideal for comparing user queries with stored facts. The second is GPT-4-Turbo, a generative language model used to produce fluent, fact-based responses by combining retrieved information with user input. This retrieval-augmented generation approach ensures both accuracy and contextual relevance in chatbot interactions.

These findings suggest the potential for additional training, data enrichment, and eventual deployment in a user-friendly web application powered by frameworks such as Flask. Ultimately, this chatbot represents a promising step toward creating intelligent, domain-specific assistants that make relevant information more accessible to a broader audience as well as an additional step whose main goal is to preserve the history of Bosnia and Herzegovina and its sports for all sport generations.

References

[1] P. Pooja, "Intelligent conversational chatbot," *Int. J. Sci. Res. Eng. Manag.*, vol. 9, pp. 1–9, 2025, doi: 10.55041/IJSREM47310.

[2] A. Singh, "Mental health support chatbot," *Int. Sci. J. Eng. Manag.*, vol. 4, pp. 1–7, 2025, doi: 10.55041/ISJEM03966.

[3] O. Karande, R. Dhavale, A. Bhagat, and S. Sapate, "AROS: Local AI chatbot and server," *Int. Res. J. Modernization Eng. Technol. Sci.*, vol. 7, 2025.

[4] L. Rivera *et al.*, "Cascade conversations: Empowering cancer genetic testing through cascade chatbots," *J. Clin. Oncol.*, vol. 43, 2025, doi: 10.1200/JCO.2025.43.16_suppl.e22621.

[5] S. Thota, S. Bongoni, P. Reddy, and M. Parameswar, "Smart banking chatbot," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, pp. 2319–2323, 2025, doi: 10.22214/ijraset.2025.67801.

[6] A. Singh, "Chatbot ticketing system," *Int. J. Sci. Res. Eng. Manag.*, vol. 9, pp. 1–9, 2025, doi: 10.55041/IJSREM47242.

[7] K. Padmavathi, M. Prasanna, K. Prathap, and T. Madhu, "Chatbot deployment for college recommendation," *Int. J. Innov. Sci. Res. Technol.*, pp. 3381–3386, 2025, doi: 10.38124/ijisrt/25may2088.

[8] G. Mahant, "CollegeConnect AI: An intelligent chatbot for student support," *Int. J. Sci. Res. Eng. Manag.*, vol. 9, pp. 1–9, 2025, doi: 10.55041/IJSREM47592.

[9] N. Subramaniam, S. Raghavan, and A. Awang, "JavaTutorBot: An interactive learning chatbot with generative AI for programming," in *Proc. Int. Conf. Advances in AI*, 2025, doi: 10.1007/978-3-031-80388-8_4.

[10] S. Yadav, "Integrating cancer support and technology in breast cancer care: Developing a chatbot for underserved populations," *J. Clin. Oncol.*, vol. 43, 2025, doi: 10.1200/JCO.2025.43.16_suppl.e13545.

[11] P. Satpute, "Asthma care chatbot," *Int. J. Sci. Res. Eng. Manag.*, vol. 8, pp. 1–11, 2024, doi: 10.55041/IJSREM37837.

[12] C. Segura, À. Palau, J. Luque, M. R. Costa-Jussà, and R. E. Banchs, "Chatbol, a chatbot for the Spanish 'La Liga'," in *Proc. 9th Int. Workshop Spoken Dialogue Syst. Technol.*, LNEE, vol. 579, Springer, Singapore, 2019, doi: 10.1007/978-981-13-9443-0_28.

[13] W. Wang *et al.*, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *arXiv preprint* arXiv:2002.10957, 2020.

[14] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," *arXiv preprint* arXiv:1908.10084, 2019.

[15] OpenAI, "GPT-4 technical report," 2023.

[16] T. Brown *et al.*, "Language models are few-shot learners," *arXiv preprint* arXiv:2005.14165, 2020.

[17] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *arXiv preprint* arXiv:2005.11401, 2020.