Personal and Ubiquitous Computing (2024) 28:947–960
https://doi.org/10.1007/s00779-024-01824-6

ORIGINAL PAPER

A novel solution for the development of a sentimental analysis chatbot integrating ChatGPT

Filippo Florindi[1] · Pasquale Fedele[1] · Giovanna Maria Dimitri[1]

Received: 24 February 2024 / Accepted: 21 June 2024 / Published online: 1 July 2024

Abstract
In today's business landscape, Chatbots play a pivotal role in innovation and process optimization. In this paper, we introduced a novel advanced Emotional Chatbot AI, introducing sentiment analysis for human chatbot conversations. Adding an emotional component within the human-computer interaction, can in fact dramatically improve the quality of the final conversation between Chatbots and humans. More specifically, in our paper, we provided a practical evaluation of the EmoROBERTA software, introducing it into a novel implementation of an Emotional Chatbot. The pipeline we present is novel, and we developed it within a business context in which the use of sentimental and emotional responses can act in a significant and fundamental way toward the final success and use of the Chatbot itself. The architecture enriches user experience with real-time updates on the topic of interest, maintaining a user-centric design, toward an affective-response enhancement of the interaction established between the Chatbot and the user. The source code is fully available on GitHub: https://github.com/filippoflorindi/F-One.

Keywords Chatbots · Emotion recognition · Implementation

1 Introduction

In recent years, Chatbots have surged in popularity, emerging as indispensable tools for companies of all sizes. The widespread adoption of Chatbot-based solutions signifies their pivotal role in modern business strategies. The rapid advancements in Natural Language Processing (NLP) over the past five years, propelled by sophisticated Deep Learning architectures and algorithms [1], have catalyzed the creation of highly advanced AI Chatbots. Such progress has opened up new perspectives and achieved levels of performance that would have been unthinkable with traditional Machine Learning models. The main models employed in the context of NLP based on Deep Learning are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) neural networks, Gated Recurrent Unit (GRU) neural networks, basic Sequence to Sequence (Seq2Seq) models, Seq2Seq Models with Attention Mechanism and Transformers Models [1–6]. In this paper, we present a novel Emotional Chatbot AI–based with a focus on Question & Answer Text Generation based on Emotional Detection. The Chatbot interacts with the users, engaging in conversations and providing requested information through written messages. More specifically, we developed and implemented a framework that operates on a knowledge-based environment, therefore being able to seamlessly engage in conversations with users, delivering requested information through written responses. More specifically, we included, and embedded, in a novel way the most up-to-date engineering techniques for Chatbots developments integrating them in new AI-driven conversational interfaces. The significance of Chatbots delivering empathetic and sentimental responses lies in their capacity to establish a genuine and relatable connection with users. By understanding and responding to emotions, Chatbots, in fact, can provide a more human-like and supportive interaction. This capability not only enhances user satisfaction and engagement but also contributes to building trust and

Filippo Florindi and Giovanna Maria Dimitri contributed equally to this work.

✉ Giovanna Maria Dimitri
giovanna.dimitri@unisi.it

[1] Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, University of Siena, Via Roma 56, Siena 53100, Italy

fostering a positive user experience. Empathetic Chatbot responses are particularly valuable in scenarios where users seek emotional support, making the interaction more meaningful and aligned with human communication. For our novel implementation, we release the step-by-step implementation of the Chatbot, together with a proof of concept related to the application of the Chatbot in a Business Case scenario for Formula One. The code we release, together with the instructions is easily implementable and customizable by the users, and it represents a valid tool for the implementation of novel sentimental prototypes. As a proof of concept, we showed the performances and the implementation of it in the context of the Formula 1 regulations, showing the possibility of successfully employing it in a business scenario. The paper is organized as follows: in Section 2, we present the background in which our work lays its foundations. In Section 2.1, the state of the art of emotional Chatbots is given. Moreover, in Section 3, the design of the sentimental Chatbot is sketched. In Section 4, we give an overview of the experimental settings and results, and eventually in Section 5, we draw conclusions, limitations, and future perspectives of the work.

2 Background

Generative AI Chatbot models represent a constantly evolving field focused on creating systems capable of generating coherent and linguistically correct responses based on user interactions. These models are designed to generate new sentences, word by word, using user input as a starting point. Training generative Chatbot models require a large dataset of sentences from real conversations. During this phase, the models learn the structure, syntax, and vocabulary of sentences in order to produce appropriate and coherent responses based on the received input. Typically, Deep Learning algorithms such as encoder-decoder neural network models are used, incorporating long-term memory mechanisms. One of the most widely used models in the field of AI Chatbots is the Sequence to Sequence (Seq2Seq) [7] model. Originally developed for machine translation, Seq2Seq models have proven to be highly effective in natural language generation as well. However, Seq2Seq models hold some limitations. The main drawback is the difficulty of capturing the entire information contained in the input sentence, as it needs to be encoded into a single fixed-length context vector. Specifically, handling long-range dependencies between distant words within a sentence can be challenging. Additionally, training and inference of such models require complex sequential computations, leading to prolonged execution times. To address long-range dependencies approaches like input sentence reversal or repeated input sentence insertion can be applied. However, these methods may not universally work effectively for all languages and do not represent a general solution. Despite the challenges and limitations, Seq2Seq models remain a popular and widely used choice in the industry for dialogue generation and many other natural language processing tasks. Their end-to-end nature makes them suitable for training on different datasets and domains without requiring domain-specific knowledge. Additionally, if needed, Seq2Seq models can be adapted to work with other algorithms or techniques, allowing for greater flexibility. However, the aforementioned issues often lead Seq2Seq models to produce vague responses. Furthermore, during response generation, issues of coherence and continuity in conversations may arise. To overcome these limitations, various advanced techniques and architectures have been proposed, such as the use of Attention Mechanisms [8] to focus on relevant parts of the input and Transformer-based Generative Models. These developments continue to enhance the capabilities of Seq2Seq models in handling complex sequences, producing high-quality results.

In recent years, the introduction of Transformer models [9] has revolutionized the field of deep learning-based language models. Transformer models, which employ the Attention Mechanism, have become a preferred choice for many challenges in natural language processing. Thanks to their ability to handle long-term dependencies and enable parallel training, Transformer models have proven to be extremely effective. Pretrained Transformer models, such as BERT [10] and GPT [11–13], have been developed and can be further fine-tuned for specific applications. With the advent of Transformers, numerous Large Language Models (LLMs) [14] have been developed in recent years, many of which have been applied in Chatbot-based solutions. The progress of AI Chatbot generative models continues pushing toward new limits. Key objectives include optimizing response coherence, improving contextual understanding, and managing complex conversations. Overcoming current limitations and enhancing human-bot interaction remains a significant challenge for researchers in this rapidly evolving field.

2.1 State of the art of sentimental Chatbots

The use of sentiment and emotion recognition in Chatbots has a long-standing story. If the idea of answering, trying to match the sentiment and the emotion of the interacting human agent, can be dated back to the ELIZA, the prototype of all of the existing Chatbots [15], it is also true that the latest deep learning architectures have opened the path to incredibly new and engaging solutions to be considered in the concept of sentimental solutions. The development of emotional Chatbots has significantly advanced with the integration of natural language processing (NLP), affective computing, and machine learning techniques [16,

17]. Emotional Chatbots are, in fact, designed to recognize, analyze, and appropriately respond to human emotions, thereby enhancing human-computer interactions by making them more natural and engaging. More specifically sentiment analysis tools and applications have become crucial in the development of such tools. For instance, in [18] the authors discuss the use of a deep learning model for recognizing human facial emotions, emphasizing the need for robust training datasets and addressing the challenge of expression variations. The model leverages convolutional neural networks (CNNs) and stacked autoencoders to enhance accuracy in emotion detection. Moreover in [19] the authors introduce a novel methodology employing MediaPipe and SRGAN for emotion detection, focusing on preprocessing and key landmark analysis to generate distinguishable features for classification. Additionally in [20] the authors present an emotion detection approach using EfficientNet, which incorporates data augmentation and feature extraction techniques to improve the robustness of the emotion recognition system. Finally, in [21] the authors implement a multilabel CNN for facial expression recognition and ordinal intensity estimation, highlighting the importance of overcoming interclass variation and employing advanced classification techniques. This makes such algorithms fundamental in a contextual environment where engagement and emotional response are crucial, for instance, medical context or similar contextual applications. In [22], for instance, the authors implemented a neural network-based speech emotion recognition system to classify emotion in a chatbot–a virtual assistant developed to do contact tracing during the COVID-19 pandemic. One of the critical areas of research is emotion recognition, which employs both lexicon-based methods, utilizing pre-defined dictionaries of emotional words such as the NRC Emotion Lexicon, and machine learning-based methods, which use classifiers like Support Vector Machines (SVMs), Naive Bayes, and deep learning models to predict emotions from text only. In the realm of emotion generation, in [23] the authors introduced an innovative approach using Transformer-based models and emotion-aware attention mechanisms, significantly improving response relevance and emotional expressiveness. Moreover, sentiment analysis has been effectively integrated into Seq2Seq models to enhance the sentiment appropriateness of Chatbot responses as for instance in [24]. For managing dialogue flow, both rule-based approaches, which follow predefined rules, and data-driven approaches, which leverage dialog datasets to train models, were explored. Furthermore in [25] the authors developed a multi-turn dialogue management system using reinforcement learning to maintain emotional consistency over extended interactions [25]. Additionally, in [26] the authors focused on generating empathetic responses in open-domain dialogues by fine-tuning pre-trained language models on empathetic dialogue datasets [26]. To enhance context-awareness in emotional Chatbots, in [27] the authors integrated memory networks, allowing the system to track and utilize conversation history for better emotion recognition and response generation [27]. Furthermore, in [28] provided a valuable contribution by presenting a large-scale emotion-labeled dialogue corpus, which is instrumental for training and evaluating emotional dialogue systems [28]. These advancements collectively contribute to the ongoing research focus on improving emotion recognition, response generation, and dialogue management, aiming to create more natural and emotionally engaging human-computer interactions.

3 Conceptual design of the sentimental chatbot

The innovative architecture, we present was crafted using a combination of several tools, with Python serving as the primary implementation language for the entire Chatbot. The implementation released provides the embedding and integration of several libraries for text processing, Chatbot interfacing, and developments, all embedded in our environment. We freely released the code at https://github.com/filippoflorindi/F-One.

3.1 Proof of concept context definition

In order to define a proof of concept for our novel Chatbot architecture proposed, we decided to implement it in the context of Formula 1, from here the reason why we denominated it F-One. In Section 3.1.1, we will describe the domain of expertise of the Chatbot. In Section 3.1.2, we will define the class of membership of the chatbot.

3.1.1 Chatbot domain of expertise

The specific domain in which the proof-of-concept of our novel Chatbot architecture operates is Formula 1,[1] the world's most prestigious category of auto racing. For this reason, the Chatbot developed is called F-One. The Chatbot was primarily developed to answer questions related to the Formula 1 regulations, which are a set of rules established by the Fédération Internationale de l'Automobile (FIA), to govern the Formula 1 championship. The regulations are structured into different areas that cover the sports, technical, and financial aspects of the competition. The Chatbot we developed offers a wide range of information regarding the world of Formula 1. The architecture implemented allows to answer

[1] https://www.formula1.com/

questions related to the regulations, as well as provide details on various aspects of this sport. For instance, it can offer in-depth information about the drivers, teams, circuits, and race results. Furthermore, it is able to address inquiries regarding historical events that have shaped the history of Formula 1. The Chatbot remains constantly updated with the latest news concerning Formula 1, including statistics, results, and current news. As a result, users can be assured of obtaining accurate and up-to-date information on the latest developments.

3.1.2 Chatbot class of membership

The distinction between two categories of AI-based Chatbots should be highlighted: Retrieval-based models and Text-Generation models. These two categories have different characteristics and are preferred depending on the context and objective of the Chatbot. Consequently, Chatbots are usually classified into one of these two. The F-One Chatbot, developed in this project, can be considered to belong to both categories. This is made possible by combining the features of Text-Generation models and Information-Retrieval models. This combination allows for overcoming the main limitations of both models while leveraging their capabilities. Retrieval-based models rely on predefined responses, chosen from a set of options. They ensure meaningful content, but they require manual input or selection of responses, which can be time-consuming. Such Chatbots excel in answering specific topics, making them valuable for tailored business Chatbots. On the other hand, Text-Generation-based models, like Large Language Models (LLMs), autonomously generate responses, avoiding the need for manual response dataset creation. However, LLMs lack domain-specific knowledge, potentially leading to less meaningful responses. Yet, techniques like in-context learning [29] can personalize LLM responses, enhancing their effectiveness for specific tasks or domains. In-context learning provides task-specific instructions directly through prompts, improving adaptability and maintaining a broader understanding of language. In fact, F-One uses an information retrieval system to provide the necessary information to answer the user's question, ensuring coherent and meaningful responses. At the same time, it uses LLM to understand the user's request, and the conversation context, and generate a human-like response, leveraging the information provided by the retrieval system implemented.

3.1.3 Evaluation metrics

We decided to then proceed with the evaluation of the quality of the answers produced by our Chatbot. The Question and Answering system used by F-one to provide information on the Formula 1 Regulations has been evaluated using various metrics to assess its effectiveness and accuracy. The metrics used were as follows:

- BERTScore: [30] is an evaluation metric that measures the similarity between generative sentences or automatic translations compared to a reference text using the BERT language model. The BERTScore metric assesses the quality of generation or translation by using the cosine of the similarity between sentence vectors. Unlike other evaluation metrics such as BLEU [31] and CHRF [32], BERTScore takes into account the semantic context and can provide more accurate and consistent evaluations of generated or translated sentences. It is widely used in natural language processing research to assess the quality of language generation.
- UNIEVAL: [33] UNIEVAL is an innovative and multidimensional evaluation system designed for text generation tasks. It utilizes a unified approach based on Boolean Question Answering to assess generated text from various perspectives using a single model. UNIEVAL evaluates sentences based on different linguistic qualities such as coherence, cohesion, fluency, and relevance. The intermediate learning phase of UNIEVAL incorporates tasks related to natural language evaluation, allowing for the acquisition of relevant external knowledge. The unified framework simplifies usage, promotes internal synergy among evaluation dimensions, improves the performance of pretrained language models, and demonstrates remarkable adaptability to new evaluation dimensions and tasks. Experimental results have shown that UNIEVAL outperforms other advanced evaluators, achieving a higher correlation with human judgments and significant improvements in text summarization evaluation and dialogue response generation. Additionally, UNIEVAL exhibits a strong transfer learning capability, achieving better performance than baseline metrics on unseen evaluation dimensions and tasks in a zero-shot setting. The naturalness, coherence, engagement, groundness, and understandability were implemented using the UNIEVAL assessment tool, and all of the details of the implementation can be found at.[2]
- We further evaluated also the response times of the Chatbot, and we report them in the Supplementary Material of the manuscript.

3.2 Detailed implementation workflow description

To develop the sentimental Chatbot proposed, we combined several tools. The design phases can be divided into seven distinct steps:

[2] https://github.com/maszhongming/UniEval

1. **PDF Processing:** Extracting information from PDF files.
2. **Vector Store Creation:** Creating a store for vector representations.
3. **Development of LangChain Agent:** Use of the LangChain framework to implement the agent.
4. **Initialization of Dialogflow CX Agent:** This consists of setting up the Dialogflow CX agent for interactive conversations.
5. **Flask Application Creation:** Building a web application using the Flask framework.
6. **Website Creation:** Developing a website to interact with users.
7. **Evaluation of Response Quality:** Assessing the quality of responses generated by F-One.

4 Experiments and results

In this section, we will describe thoroughly the implementation and development of all of the steps described in Section 3.2. All of the implementation steps are freely available at: https://github.com/filippoflorindi/F-One and we will describe all of such steps in the rest of this section.

4.1 Tools used

Notable Python libraries used in this project include:

- **Pdf Plumber:** A Python library [34] specializing in extracting text, images, and tabular data from PDF files.
- **LangChain:** A Python framework [35] streamlines the incorporation of Large Language Models (LLMs) into applications. LangChain facilitates connectivity between LLMs and external data sources, offering wrapper components for popular LLMs and prompt templates for input creation.
- **Transformers:** Developed by Hugging Face [36], this library is a cornerstone in Natural Language Processing and deep learning, boasting a variety of pre-trained LLMs models and Transformers architectures, representing the fundamentals of today's NLP.
- **Flask:** A Python framework [37] designed for crafting web applications. Flask equips developers with essential tools for managing HTTP requests, routing, HTML templates, and user sessions.
- **Critique:** Developed by Inspired Cognition [38], Critique is a cutting-edge quality monitoring Python library tailored for generative AI. It simplifies the detection of potential faults by computing scores from a set of metrics to evaluate the quality of text generated by LLMs.

In conjunction with these Python libraries, Dialogflow [39], a Google-developed platform for creating Chatbots and conversational agents, was integrated. Additionally, Ngrok,[3] a web application development tool, was employed to create secure tunnels between local servers and the external world, facilitating testing and sharing of applications. Essential web development languages-HTML, CSS, and JavaScript- were also used. GitHub Pages [40], a hosting service by GitHub, facilitating easy publication and sharing of static websites.

For evaluating the quality of F-One's answers in the Question & Answering system about Formula 1 Regulations, two metrics were employed:

- **BERTScore** [30]**:** An evaluation metric measuring the similarity between generative sentences or automatic translations and a reference text, utilizing the BERT language model.
- **UNIEVAL** [41]**:** An innovative and multidimensional evaluation system designed to assess the effectiveness and accuracy of the answers produced by F-One.

In Fig. 1, we present an overall description of the architecture proposed.

4.2 PDF preprocessing

In the first phase, we processed the PDF documents related to the Formula 1 regulations to extract relevant information with the Python library PdfPlumber. To do this, we implemented an algorithm that can decompose the document text into blocks of information content having related metadata. In detail, to filter out irrelevant content like footnotes, we employed a method that involved iterative page processing, window cropping, and text line extraction. Subsequently, a classification process was implemented to categorize text lines into content, chapter, paragraph, or subparagraph, enabling us to select relevant text for the Chatbot's responses while retaining additional context. We then used the LangChain library to create a text-splitter object, which facilitated the chunking process. To adhere to constraints such as token limits and semantic search requirements, we set the maximum chunk size at 800 characters. Additionally, a 10-character overlap ensured context preservation during text splitting. The output of this phase comprises split documents ready for the next phase. The metadata associated with each spilt document are:

- The source (the title of the PDF to which it belongs).
- The page number within the PDF.
- The chapter to which it belongs (if any).
- The paragraph to which it belongs (if any).
- The sub-paragraph to which it belongs (if any).

[3] https://ngrok.com/

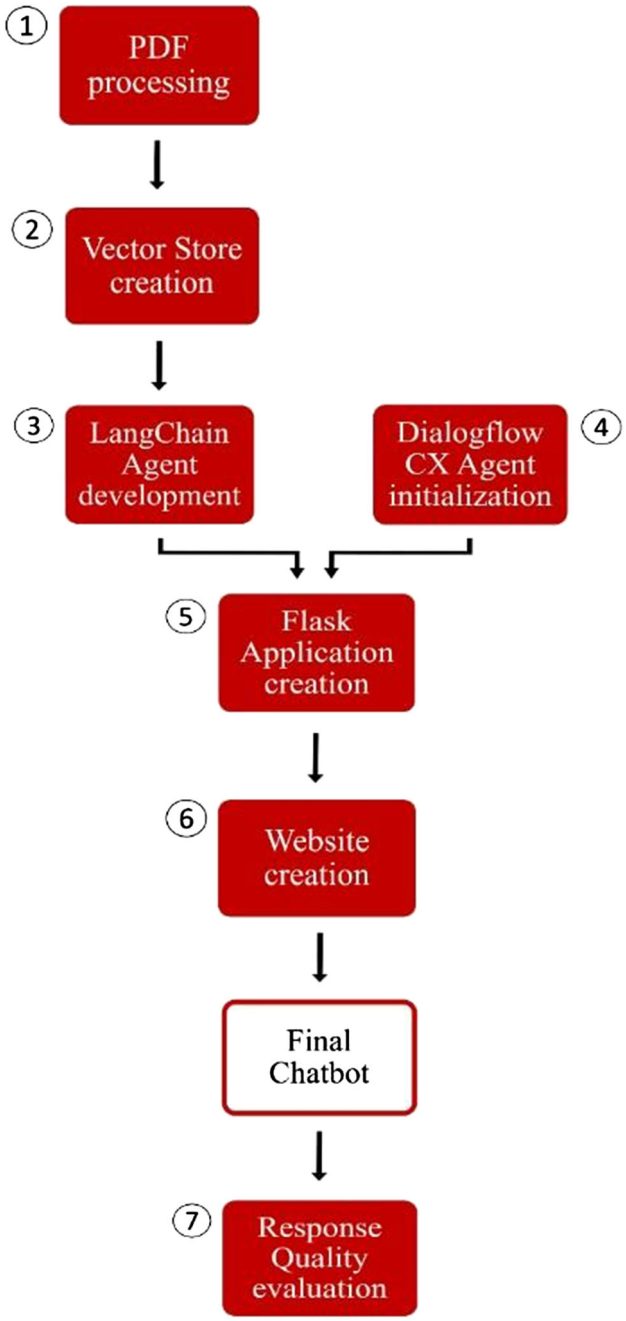

Fig. 1 Workflow of the Chatbot implemented

4.3 Vector store creation

In the second phase, we used the information extracted from the PDF documents to create a Vector Store. A Vector Store is a specialized type of database optimized for storing documents and their vector representations (embeddings), allowing the retrieval of the most relevant documents for a given query, i.e., those whose vector representations are most similar to the query's vector representation. The primary advantage of a vector store lies in its capability to conduct searches based on semantic similarity, enabling the discovery of relevant documents even if they don't contain exact matching terms. Thanks to its efficiency in calculating vector similarity, vector stores can handle large datasets swiftly and at scale, facilitating interactive and real-time searches. In the creation of the Vector Store for Formula 1 regulation information, we employed such an approach. To do so, we used the Text_Embedding-Ada-002 model from OpenAI. The vector store created for this project was used in the Information-Retrieval system. To realize the Vector Store, we used FAISS Vector Store.[4] The FAISS index used in F-One is of type Flat, which keeps the vectors unchanged.

4.4 Langchain implementation

In the third phase, we created the LangChain Agent, which uses the pre-built vector store from the previous phase. Specifically, the Agent consists of several combined components. The main components of LangChain used in the Agent are the LLMs, Tools, Memory, and Prompt. In addition to these three, there is the Emotion Detection process, which does not occur through LangChain. The LLM used in the LangChain Agent implemented in F-One is OpenAI's model, known as GPT−3.5-Turbo (ChatGPT).[5] Regarding the tools, they allow the Chatbot to have contextual information upon which it generates a response to the user's question. Our implementation is made of two tools. The first one provides information related to the regulations of Formula 1 through the Information-Retrieval system described previously. We implemented this system by creating a RetrievalQA chain. The second Tool is responsible for providing past and recent information regarding news and results about Formula 1. To implement this Tool, we used the LangChain class GoogleSearchAPIWrapper, which allows utilizing Google Search to obtain information to pass to the LLM. In the specific case of F-One, we created a custom search engine using Google Programmable Search.[6] The custom search engine used by F-One only includes the official Formula 1 website. In this way, the Chatbot is able to respond to all queries concerning Formula 1 information that is different from the regulations. Moreover, through this implementation F-One has controlled access only to correct information. The other fundamental component of the LangChain Agent is memory. As a conversational assistant, F-One needs to be able to manage a memory state that allows it to remember information communicated by the user during the conversation. LangChain allows integrating a memory system into an Agent that uses an LLM. The type of memory that we implemented in F-One

[4] https://ai.facebook.com/tools/faiss/

[5] https://openai.com/index/gpt-3-5-turbo-fine-tuning-and-api-updates/

[6] https://programmablesearchengine.google.com/about/

is called ConversationBufferWindowMemory. With this type of conversational memory, the messages exchanged between the Chatbot and the user are collected and passed as raw input to the LLM in each interaction. The buffer directly saves each interaction in the chat history. ConversationBufferWindowMemory applies a window to memory.

4.5 EmoRoBERTA model

In our Chatbot, we added, in a novel way, the emotional component through the use of the EmoRoBERTA algorithm [42]. EmoRoBERTa is a cutting-edge model designed for emotion detection in NLP, advancing the understanding of emotions in human communication. Developed using the RoBERTa framework by Facebook, it leverages pretraining and refinement on a large dataset, notably the extensive GoEmotions dataset. This dataset contains over 58,000 Reddit comments organized into 27 emotional categories, enhancing EmoRoBERTa's classification accuracy across diverse emotional contexts. The model underwent data preprocessing and training phases, exploring various configurations to optimize classification results. EmoRoBERTa showed promising performance in accurately classifying emotions, marking a significant contribution to NLP emotion detection, with potential applications in semantic analysis, propaganda studies, and advanced Chatbots. EmoRoBERTa is capable of recognizing the emotion expressed by the user by analyzing the input message sent to F-One. The Emotion Detection process with EmoRoBERTa is done separately from the LangChain Agent. The detected emotion is passed to the agent as a variable. More specifically the process which we implemented entails extracting the query from the user request, and then we passing it to EmoRoberta and afterward the sentiment defined by EmoRoberta is directly passed as a context string in the prompt used as input for the LLM. The LangChain architecture of our proposed Chatbot is presented in Fig. 2

4.6 Prompt implementation

The last component to consider is the prompt. To build our novel Chatbot, we created a custom prompt consisting of

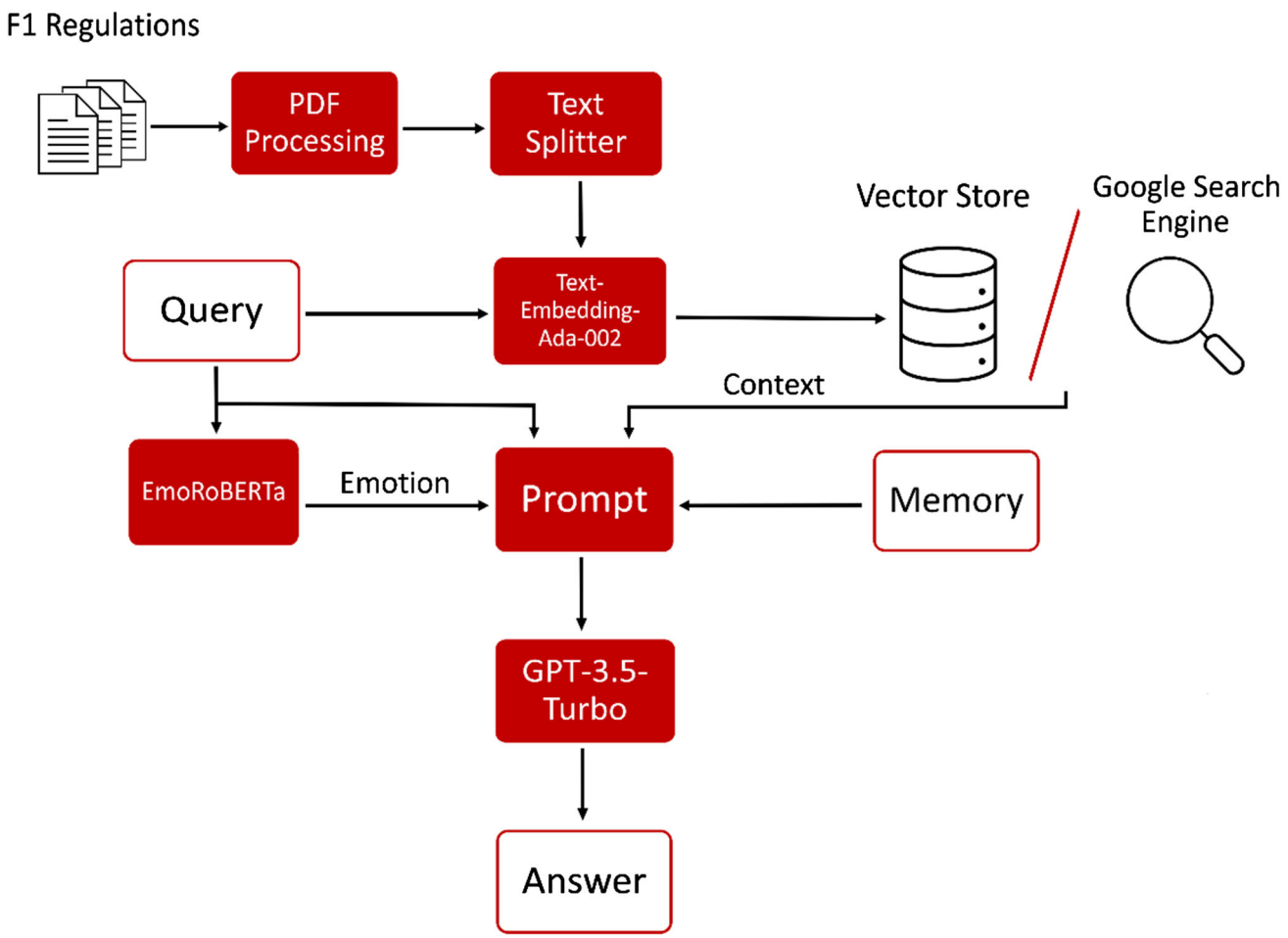

Fig. 2 LangChain agent architecture

three main components: a System Message, a Human Message, and System Variables. The System Message explains to ChatGPT its role and task. The Human Message, on the other hand, provides ChatGPT with access to the Tools, to the response formatting instructions that ChatGPT will return, to the emotion captured by EmoRoBERTa, and to the input message sent by the client to F-One in the current iteration. The memory of the previous conversation, on the other hand, will be passed to the LLM as a System Variable. All these components are combined to create F-One's Conversational Agent.

The created Agent is of the Chat-Conversational-ReAct-Description type, which means:

- Chat-Conversational: the agent application is a Chatbot with conversational memory.
- ReAct: it is a paradigm that combines reasoning and action with LLMs to solve various tasks of linguistic reasoning and decision-making, allowing models to interact with external environments while maintaining high-level action plans.
- Description: because the LLM relies on the description of the tool to decide which tool to use.

4.7 DialogFlow initialization

In this phase, we initialized the Dialogflow CX Agent. Dialogflow allows for the expansion of Chatbot functionalities and the management of various aspects. In the specific case of F-One, we modified the Start Page in the Default Start Flow, through the Dialogflow CX console. The Start Page includes the Default Welcome Intent and it also includes handlers for the sys.no-match-default and sys.no-input-default events. The first one allows setting responses that the Chatbot will provide whenever the user's input does not match any intent within the handlers' scope. The second one allows setting responses that the Chatbot will provide whenever the user's input has not been received. First, we set the response that the Chatbot will provide when the Default Welcome Intent is recognized. This way, for all welcome messages sent by a user to F-One, the Chatbot will respond as follows: "Hi, my name is F-One and I'm here to assist you! What do you want to know about Formula 1?" Next, to connect the LangChain Agent to the Dialogflow Agent and enable response generation, we created a webhook. We named the webhook as "f1-webhook." Moreover, we enabled a webhook calling in Dialogflow for both the sys.no-match-default and sys.no-input-default events. Furthermore, in fulfillment there is a field associated with the webhook, called the tag. The tag is typically used by the webhook service to identify which fulfillment is being invoked. In the Dialogflow CX Agent that we created for F-One, the tag that is used is "f1." These steps alone do not make everything work. It is in fact also necessary to implement a locally executed application to combine the two Agents. The webhook only serves as a bridge for the connection.

4.8 Flask application

In this step, we created the Flask application using Python. This application is responsible for combining the LangChain Agent with the Dialogflow CX agent through the webhook, in order to obtain the F-One Chatbot as a unified entity. In the case of this project, we created a route method named route/webhook with a POST request to handle all the webhook requests from the Dialogflow CX Agent. Therefore, whenever a fulfilment is triggered with a webhook, Dialogflow sends a POST HTTPS webhook request to this URL/webhook. The body of this request is a JSON object called WebhookRequest, containing information about the session. The function associated with the URL/webhook performs the following operations. First, it extracts the relevant session information from the JSON request object. This information includes the tag associated with the webhook, the text of the user's question message sent to the Dialogflow CX Agent, and the session name. After that, a check is performed to ensure that the tag matches "f1." Once the tag is validated, the following steps are taken. The Emotion Recognition process based on EmoRoBERTa is performed on the user's input message sent to Dialogflow. Then, the user input and the detected emotion are passed as variables to the LangChain Agent created for F-One, which executes the entire process described in phase three and returns the final response that F-One will use. The final response is inserted into a JSON response message, which will be used by the Dialogflow CX Agent to respond to the user. Furthermore, in case an error occurs during the process where a response cannot be provided, we implemented a setting within the Flask application to return the following predefined response: "I'm sorry, there's an error answering this message." The session name, on the other hand, is used within the Flask application to determine when the user interacting with F-One changes, and therefore the session changes. In this case, when the first request is made for each session, the LangChain Agent is reinitialized to allow F-One to reset the conversation memory. As long as the same session remains open, F-One will continue to maintain the conversational memory state correctly, so as not to lose any information. Once implemented, the Flask application is now ready to be run. However, since it is located on the local system, it is not accessible from the outside world. To integrate it as a webhook with Dialogflow CX, it needs to be made live/online. For this purpose, we used Ngrok. The Flask application brings together all of the components of F-One, which is now ready to be deployed on a website, and from that point on, users will be able to use it. In Fig. 3, we show the steps of Flask development. As we

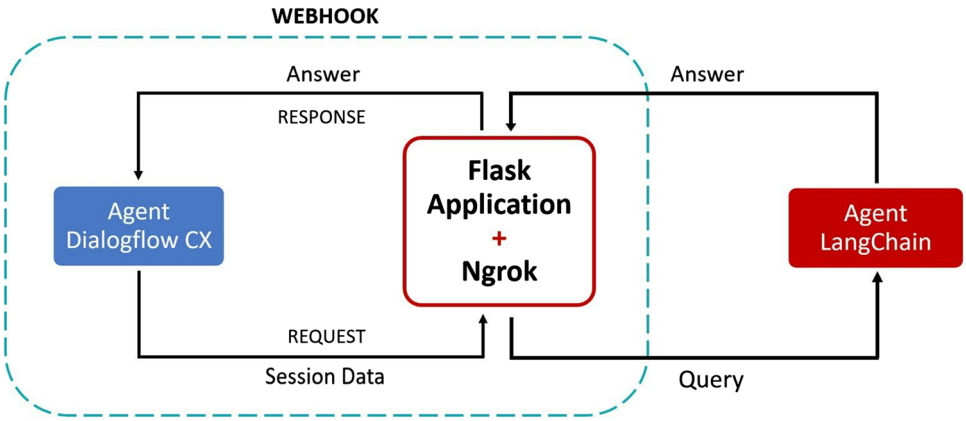

Fig. 3 Flask application architecture. In the Dashed blue set the Webhook connection between DialogFlow CX and LangChain

can see the connection between the DialogFlow X and the LangChain Agent is done through the Webhook represented in the dashed light blue set in Fig. 3.

4.9 Website interface

In the sixth phase of the project, we developed the website that hosted the F-One Chatbot for demo-showing purposes. HTML, CSS, and JavaScript were used to create the website. To integrate the F-One Chatbot into the website, we used the Integration feature of Dialogflow CX called Dialogflow Messenger. To make the website with the F-One dialogue window public, we used GitHub Pages. Figure 4 shows the homepage of the website, featuring the button and the dialogue window at the bottom right, through which F-One can be used. In Fig. 4, we show the website

4.10 Evaluation of responses quality

In the final stage, we evaluated the quality of F-One's answers with respect to questions related to information about Formula 1 regulations. This evaluation process aims to determine the accuracy of the answers provided by the Chatbot, assessing both the correctness of the context provided to the LLM and the LLM's ability to process the response by generating coherent and relevant text. The first phase of the evaluation concerns the correctness of the Question & Answer itself, which involves analyzing the effectiveness of the Information Retrieval (IR) system. The accuracy and efficiency of the IR are crucial aspects, as an ineffective information retrieval system can negatively impact the quality of the answers provided by the Chatbot. In this phase, we used BERTScore and UNIEVAL metrics. As for UNIEVAL, we used it to

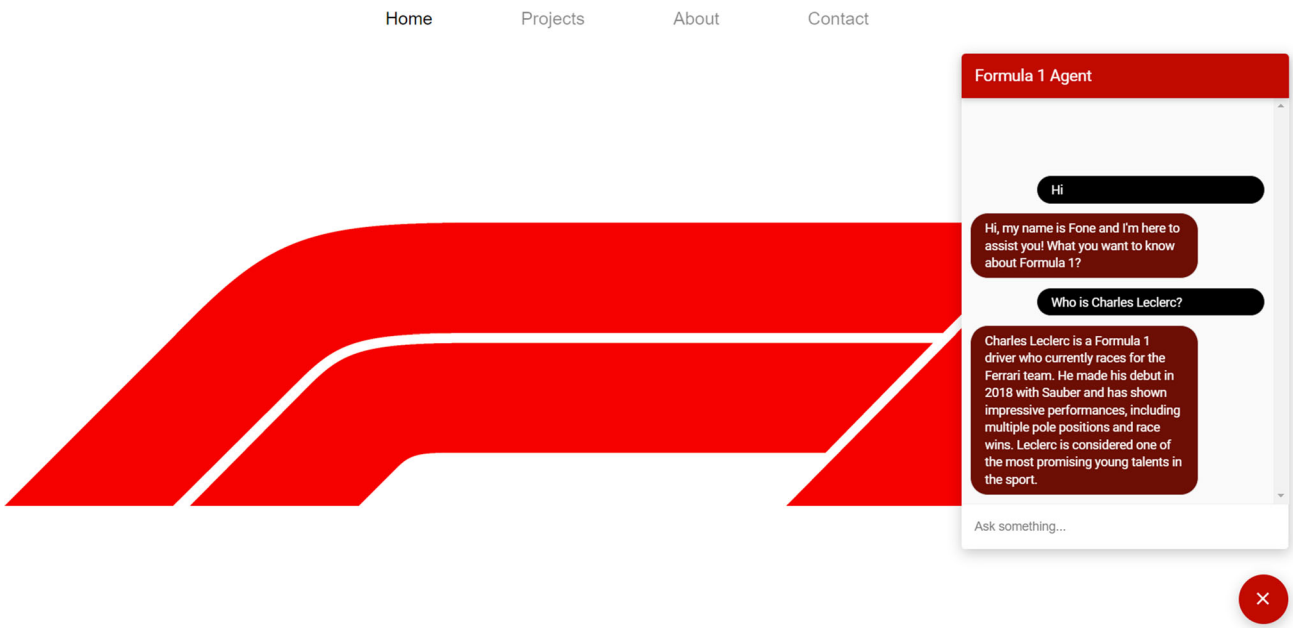

Fig. 4 Figure showing the website created together with the Chatbot dashboard

Table 1 Table showing BERT, UNIEVAL scores for the sporting, financial, and technical formula one regulations

SportBERT	SportUNI	TecBERT	TecUNI	FinBERT	FinUNI
0.6168	0.922601	0.7985	0.53526	0.8378	0.950803
1.0000	0.976769	0.6558	0.638856	0.8066	0.441136
0.6953	0.535803	0.7949	0.505142	0.8149	0.903660
0.8989	0.866939	0.8353	0.933781	0.7832	0.826500
0.7815	0.798653	0.9248	0.719562	0.8265	0.500155
0.8665	0.971138	0.8256	0.769616	0.8330	0.905605
0.7924	0.898627	0.9364	0.524970	0.8460	0.719722
0.9892	0.859361	0.7546	0.709766	0.7793	0.553241
0.7629	0.661716	0.6938	0.615149	0.7333	0.831962
0.7068	0.937588	0.7374	0.585857	0.7901	0.756445
0.8110	0.842920	0.7957	0.901	0.8051	0.79123

The last row shows the mean values obtained

obtain a factual consistency score. In fact, UNIEVAL can also serve as a high-performance one-dimensional evaluator to obtain the best correlation in factual consistency evaluation. The consistency score has a value between [0,1]. The second phase of evaluation focused on the ability of the LLM to understand the question and generate a coherent and relevant response. The LLM must be able to grasp the context of the question and produce a grammatically correct and understandable answer. Therefore, in this phase, the response generated by the LLM was assessed at the dialogue level. Also in this phase, we used the UNIEVAL metric. In this case, unlike the previous evaluation phase, we used the UNIEVAL metric to assign a score to the dialogue response generation. Specifically, we evaluated the following parameters using UNIEVAL: naturalness, coherence, engagingness, groundedness, and understandability. Engagingnes is the only dimension that uses summation scores, as it indicates the total volume of interesting facts presented in the response. Therefore, the scoring range for engagingness is between 0 and infinite, while all others are [0, 1]. Such metrics are the ones largely used in order to evaluate and assess the goodness of text produced by the generative models in the dialogical context. Since we propose a novel pipeline and implementation, we cannot directly compare it to other previously existing architecture, but we proposed metrics in order to assess the goodness of the generated text in line with those proposed and used in relevant articles [43–45] During the evaluation process, we used a test dataset, consisting of tuples, each including:

- A question about a piece of information related to Formula 1 regulations.
- A correct target answer/the correct context selected from the Formula 1 regulation PDFs.
- The response generated by F-One in relation to the question.

In this way, it was possible to compare the answers provided by the Chatbot, for a given question with the correct ones and calculate the evaluation metrics. The dataset comprises a total of 30 tuples, including 10 tuples each related to the Sporting Regulation, Technical Regulation, and Financial Regulation, respectively. In Tables 1, 2, 3, and 4, we show the evaluation of the performance indicators for the Chatbot generated responses showing the scores that we obtained for the various parameters for the 30 tuples reported in the

Table 2 Table showing naturalness, coherence, engagement, groundness, understandability, and overall scores for the sporting legislation of Formula 1 for each of the tuples tested

Nat	Cohe	Enga	Ground	Understand	Overall
0.999783	0.998975	0.998601	0.998027	0.999735	0.999024
0.999802	0.999594	0.999724	0.999272	0.999791	0.999636
0.999483	0.999747	3.990193	0.998115	0.999558	1.597419
0.999707	0.999795	2.997908	0.999584	0.999754	1.399350
0.999373	0.999783	4.996338	0.999774	0.999499	1.798953
0.999731	0.998614	0.999087	0.999274	0.999686	0.999279
0.999731	0.493846	0.998183	0.999274	0.999686	0.898144
0.999854	0.999032	1.998449	0.999698	0.999807	1.199368
0.996908	0.999564	4.994912	0.999075	0.998883	1.797868
0.99985	0.99974	2.994244	0.9998	0.999857	1.398698

Table 3 Table showing naturalness, coherence, engagement, groundness, understandability, and overall scores for the technical legislation of Formula 1 for each of the tuples tested

Nat	Cohe	Enga	Ground	Understand	Overall
0.999791	0.999642	1.998116	0.999153	0.999772	1.199295
0.998755	0.999756	2.996543	0.998717	0.998904	1.398535
0.998676	0.999373	7.989337	0.999812	0.999291	2.397298
0.999573	0.997502	0.999014	0.998881	0.999501	0.998894
0.999251	0.999199	5.978598	0.998864	0.99941	1.995064
0.999588	0.999638	4.997285	0.999381	0.999678	1.799114
0.99978	0.999603	0.999921	0.999805	0.999764	0.999775
0.999625	0.999819	2.998616	0.998462	0.999607	1.399226
0.999771	0.99982	1.998906	0.999861	0.999749	1.199621
0.999804	0.99971	4.991663	0.994522	0.999811	1.797102

Supplementary Material. From the scores, it is possible to observe that overall, the Information-Retrieval system has achieved high values for BERTScore, all above 0.6. The average value revolves around 0.8 for this case. The scores of UNIEVAL FACTUAL CONSISTENCY mostly confirm the BERTScore scores. However, some UNIEVAL scores have values below 0.6. The average of UNIEVAL FACTUAL CONSISTENCY is around 0.74. These results highlight the correct functioning of the Information-Retrieval system. The performance is indeed high. Furthermore, if we consider the tuples related to these scores, it is possible to observe that the answers are correct in terms of content and relevant to the requested context. Therefore, some low values in UNIEVAL CONSISTENCY FACTUAL may be due to the fact that sometimes F-One generates a final response that not only answers by providing the correct context but also adds additional relevant information. This aspect may not be taken into consideration by the metric in calculating the score. At the same time, sometimes F-One is unable to provide all the necessary information to fully answer but only returns some of it. This aspect could be improved by making modifications at the Information-Retrieval system level. The complete set of tuples, including questions and answers are all included in the Supplementary Material provided.

We further validated also the response times of the Chatbot in the different regulation trials we performed. For the three regulations, we reported on average a response time of 7 seconds, with a variety of length and difficulty of the response proposed. We report in the Supplementary Material the complete list of the questions and answers together with the response times for each of the responses generated by the Chatbot.

5 Conclusions and discussions

Today, Chatbots represent an extremely important tool for businesses. Digital assistants offer a tangible opportunity for innovation, as companies choose to provide an additional service to their customers and optimize processes such as customer support and information extraction. In this context, in fact, and through the use of such tools reasoning and actions could be affected in an important way. For instance, the use of a sentimental Chatbot could imply a change of direction of a certain flow of conversation, implying a change in the final decision for a customer (if we are in a business context) or in general for the other person involved in the dialog. In this regard, we developed a novel Chatbot workflow with the

Table 4 Table showing naturalness, coherence, engagement, groundness, understandability, and overall scores for the financial legislation of Formula 1 for each of the tuples tested

Nat	Cohe	Enga	Ground	Understand	Overall
0.999911	0.999794	0.999926	0.999914	0.999898	0.999889
0.999859	0.999806	0.99991	0.999837	0.999866	0.999856
0.999695	0.99975	1.999524	0.998675	0.999692	1.199467
0.999786	0.999553	3.998335	0.999536	0.999827	1.599407
0.999561	0.999668	3.997861	0.999706	0.99964	1.599287
0.999781	0.999032	0.999792	0.999722	0.999765	0.999618
0.999901	0.99958	1.998612	0.997154	0.999892	1.199028
0.999538	0.999845	2.997952	0.999476	0.999546	1.399271
0.999844	0.999685	1.998364	0.999642	0.999841	1.199475
0.999678	0.999572	0.999799	0.999746	0.999639	0.999687

intention of combining different approaches and technologies that offer a solution capable of achieving satisfactory performance while optimizing the Chatbot implementation process. We present this novel solution through the fusion of two frameworks, Dialogflow and LangChain. The former allows the Chatbot to handle specific information securely and with the assurance of the highest quality. However, this aspect is limited to a subset of topics because as the volume of data increases, the complexity and implementation time of the digital assistant also increases significantly. On the other hand, LangChain is based on Prompt Engineering techniques that leverage the power of LLMs like ChatGPT. ChatGPT, in fact, automates the Chatbot's response process by generating text that closely resembles human-generated text. Using LLMs, we, therefore, also introduce the reasoning part, due to the chain-of-thoughts integrated and embedded within the LLMs way of working. However, this type of Artificial Intelligence method must be used correctly as it alone does not adapt well to the implementation of a Chatbot that needs to respond to a particular domain of interest. There are multiple challenges in this case, from the lack of knowledge of the LLM to limited control over the generated text. However, to overcome these limitations, the Information Retrieval system implemented in F-One allows ChatGPT to be a particularly useful and efficient tool within the context of a custom Chatbot. This is because F-One has knowledge of information present within PDF documents. For many companies, this solution could represent an extremely important resource. In fact, the majority of corporate documents, and not only, are in PDF format. The application of a Chatbot like F-One is, therefore, not limited to the context of a Chatbot that offers an additional service to customers but also extends to the context of an enterprise Chatbot that can provide information requested by employees, for example. Furthermore, and most importantly, the novel Chatbot architecture proposed was designed with the intent of becoming a "sentimental Chatbot", with the aim of "understanding" the emotions expressed by the user it interacts with and responding accordingly in an empathetic and sensitive manner. Such an emotional attitude of a Chatbot can in fact become crucial to help customers and to provide support and assistance. In addition to this, F-One has access to up-to-date information and events. F-One's ability to provide recent news during conversations with messages similar to those of a human being makes it particularly innovative. The architecture of F-One is designed to optimize implementation costs in terms of time and money while maximizing the quality of the resources used. As for F-One's performance, the results that we obtained from the evaluation process show that F-One is able to provide the right information requested by the user. Such a guarantee of correctness in the content of the responses is crucial, especially regarding a complex and detailed topic such as Formula 1 regulations. F-One responds with human-like dialogue. We are aware that some limitations of our Chatbot might be due to the presence of a technology that is every day changing. Moreover, there might be the need of adding further text processing steps, to consider a wider variety of possible different use cases and contexts. Future improvements may include further data analysis implementation in the dashboard, as well as multilingual support for a larger audience. Moreover, we plan on enhancing the Chatbot's emotional intelligence by integrating more sophisticated affective computing techniques which could significantly improve its ability to understand and respond to users' emotions accurately. Additionally, exploring multi-modal approaches that incorporate not only text but also audio and visual cues could provide a richer understanding of users' emotional states, leading to more contextually appropriate responses. Furthermore, investigating personalized adaptation mechanisms based on users' individual emotional profiles could further tailor the Chatbot's interactions to meet users' specific emotional needs. Moreover, thinking also about the integration of reasoning and action, during the dynamic flow of conversation so obtained, the Chatbot could also integrate further human-like aspects, such as for instance empathetic conversation, in order to meet conversational needs, whenever there is the feel that the other person in the conversation might be irritated or similar. Finally, exploring the ethical implications of deploying sentimental Chatbots in various contexts, such as mental health support or educational settings, is crucial to ensure responsible and engaging interactions.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00779-024-01824-6.

Funding Open access funding provided by Università degli Studi di Siena within the CRUI-CARE Agreement.

Data Availability Code and data have been publicly released at the following GitHub: https://github.com/filippoflorindi/F-One.

Declarations

Ethics Approval and Consent to Participate No ethical Approval was needed for the studies conducted in this manuscript.

Conflict of Interest The authors declare no competing interests.

References

1. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
2. Spiga O, Cicaloni V, Dimitri GM, Pettini F, Braconi D, Bernini A, Santucci A (2021) Machine learning application for patient stratification and phenotype/genotype investigation in a rare disease. Briefings Bioinfo 22(5):434
3. Vozzi F, Pedrelli L, Dimitri GM, Micheli A, Persiani E, Piacenti M, Rossi A, Solarino G, Pieragnoli P, Checchi L et al (2024) Echo state networks for the recognition of type 1 brugada syndrome from conventional 12-lead ecg. Heliyon 10(3)
4. Flach P (2012) Machine learning: the art and science of algorithms that make sense of data
5. Depraetere I, Cappelle B, Hilpert M, De Cuypere L, Dehouck M, Denis P, Flach S, Grabar N, Grandin C, Hamon T et al (2023) Models of modals: from pragmatics and corpus linguistics to machine learning 110
6. Dimitri GM, Spasov S, Duggento A, Passamonti L, Toschi N et al (2020) Unsupervised stratification in neuroimaging through deep latent embeddings. In: 2020 42nd Annual international conference of the IEEE engineering in medicine & biology society (EMBC), IEEE pp 1568–1571
7. Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. arXiv:1409.3215
8. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems 30
9. Vaswani A, Shazeer NM, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. NIPS
10. Devlin J, Chang M-W, Lee K, Toutanova K (2019) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
11. Radford A, Narasimhan K, Salimans T, Sutskever I Improving language understanding by generative pre-training. OpenAI
12. Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I Language models are unsupervised multitask learners. OpenAI
13. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan TJ, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. arXiv:2005.14165
14. Zhao WX, Zhou K, Li J, Tang T, Wang X, Hou Y, Min Y, Zhang B, Zhang J, Dong Z, Du Y, Yang C, Chen Y, Chen Z, Jiang J, Ren R, Li Y, Tang X, Liu Z, Liu P, Nie J, Wen J-r (2023) A survey of large language models. arXiv:2303.18223
15. Weizenbaum J (1966) Eliza-a computer program for the study of natural language communication between man and machine. Commun ACM 9(1):36–45
16. Dimitri GM, Spasov S, Duggento A, Passamonti L, Lió P, Toschi N (2022) Multimodal and multicontrast image fusion via deep generative models. Info Fusion 88:146–160
17. Safi Z, Abd-Alrazaq A, Khalifa M, Househ M (2020) Technical aspects of developing chatbots for medical applications: scoping review. J Med Int Res 22(12):19127
18. Gao Xea (2023) Performance evaluation of machine learning for recognizing human facial emotions. Comput & Security 103476. https://doi.org/10.1016/j.cose.2023.103476
19. Patel Aea (2023) Deploying machine learning techniques for human emotion detection. Math Comput Appl 100508. https://doi.org/10.1016/j.mlwa.2023.100508
20. Wang Yea (2023) A novel lightweight deep convolutional neural network model for human emotions recognition in diverse environments. Front Psychol 1190326. https://doi.org/10.3389/fpsyg.2023.1190326
21. Ekundayo Tea (2023) Multilabel convolutional neural network for facial expression recognition and ordinal intensity estimation. J Appl Res Technol 34. https://doi.org/10.18280/ria.340304
22. Pucci F, Fedele P, Dimitri GM (2023) Speech emotion recognition with artificial intelligence for contact tracing in the COVID-19 pandemic. Cognit Comput Syst 5(1):71–85
23. Zhong P, Wang D, Miao C (2019) An affect-rich neural conversational model with biased attention and weighted cross-entropy loss. Proceedings of the AAAI Conference on Artificial Intelligence, 1–8
24. Colnerič N, Smailovič J (2018) Emotionally relevant dialogue generation. 1–9 arXiv:1806.08312
25. Zhou L, Small K, Kautz H, Prasad R (2020) Multi-turn response selection for chatbots with deep attention matching network. In: Proceedings of the annual meeting of the association for computational linguistics, 1–10
26. Rashkin H, Smith EM, Li M, Boureau Y-L (2019) Towards empathetic open-domain conversation models: a new benchmark and dataset. In: Proceedings of the annual meeting of the association for computational linguistics, 1–11
27. Majumder N, Poria S, Gelbukh A, Cambria E (2020) Dialogue: context-aware conversational models with memory networks. 1–7 arXiv:1806.08313
28. Liu Z, Shen S, Quan X, Hu W, Qin B, Liu T (2017) Emotionlines: an emotion corpus of multi-party conversations. In: Proceedings of the annual meeting of the association for computational linguistics, 1–9
29. Dong Q, Li L, Dai D, Zheng C, Wu Z, Chang B, Sun X, Xu J, Sui Z (2022) A survey on in-context learning. arXiv:2301.00234
30. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019) Bertscore: evaluating text generation with Bert. arXiv:1904.09675
31. Zhang T, Kishore V, Wu F, Weinberger KQ, Artzi Y (2019) Bertscore: evaluating text generation with Bert. arXiv:1904.09675
32. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics, pp 311–318
33. Popović M (2015) chrf: character n-gram f-score for automatic mt evaluation. In: Proceedings of the tenth workshop on statistical machine translation, Lisbon, Portugal, Association for Computational Linguistics, pp 392–395
34. Pdf plumber documentation. Available via. https://pypi.org/project/pdfplumber/
35. LangChain Python documentation. Available via. https://python.LangChain.com/en/latest/index.html
36. Hugging face transformers documentation. Available via. https://huggingface.co/docs/transformers/index
37. Flask documentation. Available via. https://flask.palletsprojects.com/en/2.3.x/
38. Critique documentation. Available via. https://docs.inspiredco.ai/critique/
39. Dialogflow documentation. Available via. https://cloud.google.com/dialogflow/docs
40. GitHub pages. Available via. https://pages.github.com/
41. Zhong M, Liu Y, Yin D, Mao Y, Jiao Y, Liu P, Zhu C, Ji H, Han J (2022) Towards a unified multi-dimensional evaluator for text generation. In: Conference on empirical methods in natural language processing

42. Kamath R, Ghoshal A, Eswaran S, Honnavalli P (2022) An enhanced context-based emotion detection model using Roberta. In: 2022 IEEE International conference on electronics, computing and communication technologies (CONECCT), IEEE pp 1–6
43. Chang Y, Wang X, Wang J, Wu Y, Yang L, Zhu K, Chen H, Yi X, Wang C, Wang Y et al (2024) A survey on evaluation of large language models. ACM Transactions on Intelligent Systems and Technology 15(3):1–45
44. Mao R, Chen G, Zhang X, Guerin F, Cambria E (2023) Gpteval: a survey on assessments of chatgpt and gpt-4. arXiv:2308.12488
45. Liusie A, Manakul P, Gales M (2024) Llm comparative assessment: zero-shot nlg evaluation through pairwise comparisons using large language models. In: Proceedings of the 18th conference of the European chapter of the association for computational linguistics (vol 1: Long Papers), pp 139–151