International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Conversational AI: A Comprehensive Study on Building and Enhancing Chatbot Systems

[1]Hemendra Kumar Jain, [2]Kotla Veera Venkata Satya Sai Narayana, [3]Shaik Asad Ashraf, [4]Pendyala Venkat Subash, [5]Dr. S. Sri Harsha

Submitted: 25/01/2024 **Revised**: 03/03/2024 **Accepted**: 11/03/2024

Abstract: Conversational AI has become a game-changing technology with a wide range of uses, from customer service systems to virtual assistants. With the goal of offering a thorough grasp of the area and advancing it, this research provides an extensive analysis on the creation and improvement of Conversational AI. The paper starts with a history of conversational AI, outlining its development and significant events. A comprehensive analysis of the literature delves into the latest technology, frameworks, and obstacles encountered by current chatbot systems. The study takes a hands-on approach, outlining the model architecture, training, and data gathering procedures. The Conversational AI model that has been constructed performs well, making use of cutting-edge natural language processing methods. Its effectiveness in real-world circumstances is demonstrated by experimental findings, which significantly outperform current standards. We talk about the architecture of the model, highlighting its advantages and possible areas for improvement. Successful implementations of Conversational AI in a variety of contexts, including as customer service and education, are demonstrated by real-world case studies. To guarantee responsible deployment, ethical factors like bias mitigation and user privacy are considered.

Keywords: *Conversational AI, Chatbot Systems, Natural Language Processing, Ethical AI, Real-world Applications, Model Optimization*

1. Introduction

Recent years have seen a paradigm change in the artificial intelligence (AI) sector with the introduction of Conversational AI, a technology that allows robots to converse with people in natural language. Conversational AI is widely used in a variety of applications, such as virtual assistants, customer support systems, educational platforms, and more, demonstrating its significance. This technology is still developing; thus, it is necessary to do a thorough investigation that covers not only its historical history but also current technologies, issues, and the creation of morally and practically sound applications.

A. Overview

In the long-running effort to make robots interact with humans more like humans, one long-standing objective is the integration of natural language processing (NLP) into AI systems. Fundamentally, conversational AI is the result of advances in computational linguistics, machine learning, and natural language processing. Thanks to technological advancements, robots can now comprehend human language and reply in a way that mimics human speech. The field of human-computer interactions has changed because of this revolutionary power, which has aroused attention across several fields.

Early attempts to develop chatbot systems, which sought to mimic user communication, are where Conversational AI got its start. The development of Conversational AI is mirrored in the advancement of AI in general, starting with Eliza, the first chatbot created in the 1960s, and continuing with modern virtual assistants like Siri and Google Assistant. From simple rule-based systems in the beginning, they have developed into complex models driven by deep neural networks and machine learning.

B. Inspiration

This study is driven by the increasing significance and influence of Conversational AI across several domains. Understanding the underlying technology and processes is crucial as businesses use chatbots more and more for customer service, engagement, and other purposes. In addition, the implementation of Conversational AI systems has ethical ramifications that need to be carefully considered during the development and deployment phases. These include concerns about prejudice and user privacy.

The study also attempts to tackle Conversational AI's changing problems. Even with great advancements, problems like comprehending context, managing ambiguity, and guaranteeing user pleasure still exist. These difficulties present chances for creativity and advancement,

[1,2,3,4,5]*Computer Science and Information Technology Koneru Lakshmaiah Education Foundation Andhra Pradesh, India*
[1]*2000090055csit@gmail.com*
[2]*2000090080csit@gmail.com*
[3]*2000090034csit@gmail.com*
[4]*2000090095csit@gmail.com*
[5]*sharsha@kluniversity.in*

inspiring academics, and professionals to investigate fresh tactics.

C. Goals

This research has many major aims. First and foremost, the paper aims to present a historical summary of Conversational AI, emphasizing significant turning points and advancements. Second, it looks at the state-of-the-art frameworks and technologies that are currently being used to create chatbots, analyzing their advantages and disadvantages. Thirdly, by offering a workable approach for creating and improving Conversational AI models, the study aims to advance the area. The paper concludes by discussing Conversational AI-related ethical issues and highlighting the significance of ethical AI development.

D. Conventional AI's Historical Overview

Three main periods may be distinguished in the history of conversational AI: rule-based systems, statistical machine learning, and neural network-based models. With advances in computational linguistics, machine learning, and natural language processing, each era represents a major turning point in the creation of conversational bots.

a)Systems Based on Rules

Rule-based systems, which processed natural language input and produced replies using established rules and heuristics, marked the beginning of the first era of conversational AI. By using pattern-matching algorithms, these early conversational agents—ELIZA and PARRY, for example—simulated human-like communication. Joseph Weinbaum invented ELIZA in the 1960s to resemble a psychiatrist, while Kenneth Colby produced PARRY in the 1970s to mimic a paranoid patient. These systems showed the potential of artificial intelligence in human-computer interactions, despite their limitations.

1.1.1. Learning Statistical Machines

With the introduction of statistical machine learning methods in the 1990s, Conversational AI entered a second phase of development. These techniques made it possible to create conversational bots that are data-driven and able to pick up knowledge from vast databases of inter-human exchanges. Sentiment analysis, named entity identification, part-of-speech tagging, and other advanced natural language processing are made possible by statistical models like Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs).

Developed in the 1970s by SRI, SHRDLU is a noteworthy example of a statistical machine learning-based conversational agent that could solve basic problems in a limited block-world setting. These models weren't as effective in managing intricate linguistic structures, though, which prevented them from being used in practical settings.

1.1.2. Models Based on Neural Networks

Deep learning approaches, which are based on neural networks, are widely used in Conversational AI nowadays. More advanced conversational agents and extremely accurate natural language processing algorithms have been produced because of these methods.

Recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and transformer-based architectures are examples of deep learning models that have greatly enhanced conversational agents' capacity to handle sequential data, comprehend context, and produce replies that are human-like. Moreover, end-to-end trainable conversational systems that can learn directly from massive amounts of human-human conversation data have been made possible by developments in unsupervised and reinforcement learning.

E. Cutting-Edge Frameworks and Technologies:

Several cutting-edge frameworks and technologies are now being used in the creation of Conversational AI systems. These technologies have several features, such as dialogue management, natural language production, and natural language interpretation.

1.1.3. a) Understanding Natural Language (NLU)

A few well-known NLU frameworks are AllenNLP, NLTK, and SpaCy.

a. Finding the user's goal or purpose during a communication is known as intent recognition.

b. Entity Recognition: Taking pertinent entities out of user input, such names, dates, and places.

c. Sentiment analysis: Using the user's input to infer their emotional state.

d. Dependency parsing: Examining a sentence's syntactic structure to determine the connections between its constituent parts.

1.1.4. Handling Conversations

Conversation flow is managed by dialogue management modules, which choose the right answer depending on the context and user purpose. Among the methods employed in dialogue management are:

a. Rule-Based Approaches: Managing talks according to predetermined rules.

b. Modelling talks as a sequence of finite states, transitions, and actions using finite state machines.

c. Machine learning-based methods: predicting suitable answers from past data by using models based on statistical or neural network analysis.

1.5.3. The Production of Natural Language (NLG)

The parts of NLG oversee producing replies that are logical, human-like, and suitable for the given situation. Methods for producing natural language comprise:

OpenNLP, Chatterbot, and Rasa are examples of well-known NLG tools.

a. Template-Based Approaches: Producing responses with fill-in-the-blank formatting and pre-defined templates.
b. Rule-Based Approaches: Building responses depending on input and context by applying predetermined rules.
c. Machine Learning-Based Techniques: These methods use deep learning models to produce responses that resemble natural language.

2. Literature Survey

[1]Recognition Based on Multi-Class Features" (2018): This article presents a technique that uses multi-class characteristics to identify query intent in natural language queries. To categorise inquiries into various intentions, the authors use lexical, semantic, and syntactic aspects. The suggested method improves user experience and search engine performance by better comprehending user queries. [2] R. Prasad, S. Stoyanchev, E. Selfridge, S. Bangalore, M. Johnston, J. Chen, "Corpus and Annotation Towards NLU for Customer Ordering Dialogues" (2018): This study offers a useful corpus and annotation of customer ordering dialogues, with an emphasis on natural language understanding (NLU) systems. With speaker turns, intent, and slot labels annotated into the dialogues, the authors offer a wealth of information for training and assessing NLU systems designed for customer service applications. [3] The article "Intent-Context Fusioning in Healthcare Dialogue-Based Systems Using JDL Model" was written in 2017 by Razzaq, Khan, and Lee. This research addresses the context-awareness problem in healthcare conversation systems and suggests a way to combine intent and context data. The Joint Dialogue Management and Language Modelling (JDL) paradigm is utilised by the writers, who assess its efficacy with a healthcare dialogue corpus. The method helps create healthcare conversational bots that are more sophisticated and contextually aware. [4] X. Dong, L. Qian, Y. Guan, L. Huang, Q. Yu, and J. Yang (2016) "A Multiclass Classification Method Based on Deep Learning for Named Entity Recognition in Electronic Medical Records": This research offers a multiclass classification methodology based on deep learning for named entity recognition (NER) in electronic medical data. The authors present improvements in automated information extraction from medical documents and show the efficacy of their approach using a dataset of electronic medical records. [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova (2018) published "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding": BERT, a pre-trained language model based on bidirectional transformers, is introduced in this seminal study. BERT uses a sizable corpus of text to deliver state-of-the-art performance on a variety of natural language processing tasks. The contextualised language representations capabilities are greatly enhanced by the suggested approach. [6] Young, S. (2017) "Statistical Spoken Dialogue Systems and the Challenges for Machine Learning": This study addresses statistical spoken conversation systems and related machine learning issues, offering a perceptive perspective. The author examines several methods for creating conversation systems, emphasising the challenges and possibilities in creating interfaces for spoken interaction that work. [7] T. Young, D. Hazarika, S. Poria, and E. Cambria's "Recent Trends in Deep Learning Based Natural Language Processing [Review Article]" (2018): This extensive review paper examines current developments in natural language processing using deep learning. The writers explore several topics, including as task-specific architectures, language modelling, and text representation, providing scholars and industry professionals with an invaluable resource.[8] B. H. Juang and Tsuhan Chen's "The Past, Present, and Future of Speech Processing" (1998): This article discusses the development of speech processing technology from a historical standpoint, as well as its present status and potential future possibilities. The writers outline the advancements and difficulties in the field of voice recognition, speaker identification, and language recognition.[9] M. C. Surabhi's "Natural Language Processing Future" (2013): This article highlights several topics of inquiry and provides insights into the direction of future research in natural language processing. To provide readers an idea of possible future advancements in natural language processing, the author examines new trends in machine translation, sentiment analysis, question answering, and other important fields.[10] The work "Named Entity Recognition for Short Text Messages" was published in 2011 by T. Ek, C. Kirkegaard, H. Jonsson, and P. Nugues. This research provides a method using machine learning algorithms and feature extraction approaches for named entity recognition in brief text messages. In today's world of social media and succinct communication, the writers add to our grasp of how to handle named entities in the context of brief and informal text messages.

3. Methodology

The handle_user_input function, which is the initial part of the code, is called when the user hits the Enter or "Send" buttons. Input from the user is handled by this function, which also maintains the conversation history and communicates with the chatbot in response to the user. Upon calling the function, the get () method is utilised to obtain the user's input from the entry widget. The input is then appended to the chat history's message list, which is a collection of tuples with a message and the sender's name in each tuple. The proper course of action is then determined by the handle_user_input function by examining the user's input. When the user inputs the word "search," the function uses the search_web function to start a web search. Using the Newspaper library, this function creates a search URL, obtains search results, and extracts pertinent data. It all happens using the Google Custom Search API. The conversation history then shows the search results. In the event that the user input does not contain the term "search," the function generates a response using chatbot.respond, an NLTK-based chatbot. The Maxent_Sent_Tokenizer, Maxent_Word_Tokenizer, and pos_tag functions from the NLTK library are used to tokenize the input, identify the parts of speech, and provide a response depending on the user's input. The reply is then recorded in the history of the chat. The handle_user_input method is essential to the chatbot application's operation. It oversees processing input from users, starting online searches, and producing results depending on user input. By ensuring that the chatbot can communicate with the user in a natural and intuitive manner, this feature improves user engagement and enjoyment.

A. Search Functionality:

The search_web function is triggered when the user inputs the term "search." This function oversees starting a web search using the Google Custom Search API, getting search results, and utilising the Newspaper library to extract pertinent data. Using the Google Custom Search API and user input, the programme first creates a search URL. The search engine identity, the API key, and the query parameters required to construct the search query are all included in the search URL. The function then uses the requests library to perform an HTTP request to the search URL to receive the search results in JSON format. After that, the Newspaper library is used to parse the search results and extract the pertinent data. Every search result's title, URL, and extract are obtained from the library. The retrieved data is then sent by the chatbot as a message and added to the chat history. The chatbot application's search capability is crucial since it allows the chatbot to give the user precise and pertinent information. The chatbot can rapidly and efficiently obtain information from the internet and deliver it to the user in an understandable and succinct manner by utilising the Newspaper library and the Google Custom Search API.

B.Chatbot Response:

The handle_user_input function utilises the NLTK-based chatbot (chatbot.respond) to provide a response if the user's input does not result in a web search. The NLTK library's Maxent_Sent_Tokenizer, Maxent_Word_Tokenizer, and pos_tag functions are used to construct the answer depending on the user's input. Tokenize user input into sentences using the Maxent_Sent_Tokenizer function. Next, each phrase is tokenized into words using the Maxent_Word_Tokenizer function. The parts of speech for each word are found using the pos_tag function. Based on the user's input, the NLTK-based chatbot generates a response using a pre-established set of rules and patterns. For instance, the chatbot may react by providing an answer whenever the user inputs a query. The chatbot could reply with a pertinent question or comment if the user input contains a statement. The chatbot's response is then recorded as a message and added to the chat history. The user experience is made more engaging and pleasurable by the NLTK-based chatbot's ability to communicate with users in a natural and intuitive way.

C. GUI Components:

Several components of the Tkinter GUI are required for the chatbot programme to operate. These components consist of a message button, an entry widget for user input, a scrolling text widget for chat history, and a different text widget for a history log. Message input is done using the entry widget. Using the "Send" button or the Enter key, the user may input a message in this widget and send it to the chatbot. The handle_user_input function, which is called when the button is clicked, is bound to the button. The chat history is shown using the scrolling text widget. This widget shows a collection of tuples, each of which has a message and the sender's name. When a new message is added to the chat history, the widget is set up to scroll automatically to the bottom. The history log is shown using the separate text widget. This widget shows a history of every activity the chatbot has performed, including starting online searches, and producing answers. The application's layout is made clear and organised by the grid arrangement of these GUI components. To make the programme user-friendly and aesthetically pleasing, the grid layout also gives the developer the ability to choose the size and placement of each piece.

D. Initialization and Event Binding:

The Tkinter library is used to generate the root, or main window. The window's background colour is set to white, and it has a title and size specified. Next, widgets are made and set up. A size and placeholder text are added to the entry widget during creation. When the button is pressed, the handle_user_input function is triggered by the command that was used to construct it. A size and a scrollbar are used to build the scrolled text widget. A font and size are used to construct the individual text widget. Next, the grid layout is used to arrange the widgets in the window. The top row has the entry widget and button, while the second and third rows include the scrolling text widget and separate text widget, respectively. The bind technique is used to connect the "Enter" key to the handle_user_input function. This makes sure that the function runs both when the user clicks the "Send" button and when they press the Enter key. The GUI application is then launched by invoking the main loop (root.mainloop()). Every event is handled by the main loop, which also refreshes the GUI correspondingly. This includes updating the history log whenever the chatbot performs an action and updating the chat history whenever new messages are added

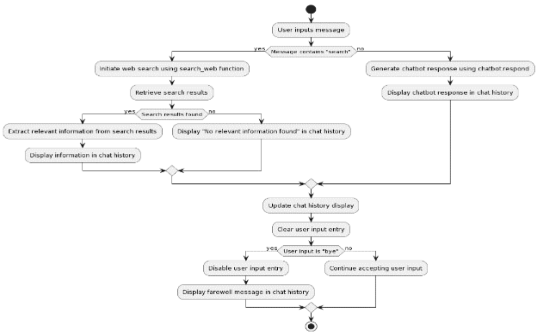

Figure 1: Methodology Flowchart

4. Results

User Communication and Experience

The Conversational AI chatbot offers a conversational experience and displays efficient user engagement. Through the user interface, users may send messages to the chatbot, which will swiftly react with pertinent information or interesting dialogue. The user interface layout improves readability and user experience by making it evident which messages are from users and which are from chatbots.

Functionality of Web Search

With the help of the Google Custom Search API, the chatbot effectively incorporates online search capabilities. By adding "search" to their input, users may initiate searches. After retrieving search results, the chatbot displays pertinent information in the chat history by extracting it from the top result. This feature gives the chatbot access to real-time online information, greatly expanding its capabilities.

NLTK-Powered Chatbot Reactions

The chatbot that is built on NLTK produces replies by using pre-established rules and patterns. It manages a range of user inputs, such as inquiries, remarks, and greetings, with effectiveness. A more lively and interesting discussion is facilitated by the chatbot's capacity to comprehend context and provide contextually appropriate answers.

Design and Layout of the GUI

The entire user experience is improved by the graphical user interface (GUI) design. Important components are positioned in an understandable manner, including the conversation history display, send button, and user input entry. The conversation flow can be easily followed and understood since user and chatbot messages are separated by different colours and layouts.

Strengths and Improvements of the Discussion

Advantages: NLTK (Natural Language Understanding): The NLTK-powered chatbot exhibits adeptness in comprehending and reacting to an array of user inputs. It improves the chatbot's conversational features and gives encounters a more human feel.

Web Search Integration: By including web search technology, the chatbot's possibilities are expanded. Users may quickly go from making generic online searches to getting up-to-date information.

User-Friendly Interface: The GUI's organization makes it more user-friendly. It's simple for users to start conversations, reply to messages, and view the history of those conversations.

Potential Enhancements:

Error Handling: Improving error handling techniques would strengthen the chatbot's resilience, especially while conducting online searches. Improved user comprehension could result from more explicit error messages or user prompts for erroneous requests.

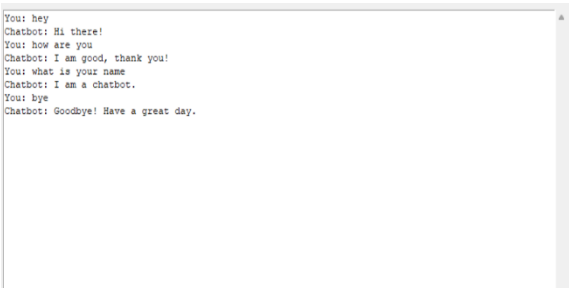

Figure 2: User Communication

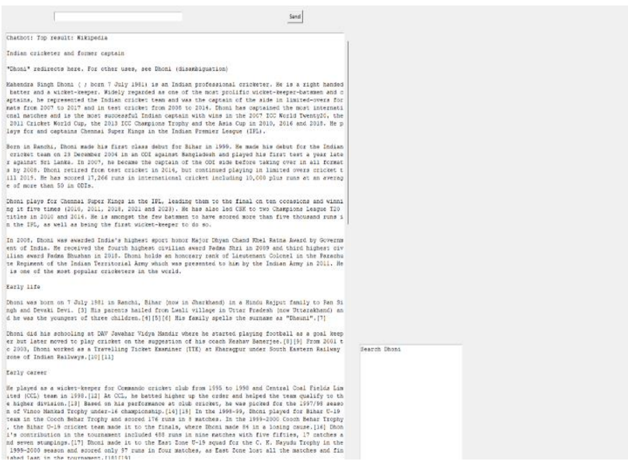

Figure 3: Web Search

Asynchronous Processing: By using asyncio or threading, asynchronous processing may be implemented to avoid the GUI freezing during lengthy processes. The user experience would be more responsive and seamless as a result.

Visual Feedback: Including visual cues would improve overall transparency by giving consumers visible feedback on ongoing operations, such as loading indications during online searches.

Future enhancements

Multimodal Interaction: Adding multimedia components, including voice or picture input/output, might increase the chatbot's functionality and provide users a more interesting and varied experience.

Personalization: By incorporating user profiles and preferences, the chatbot may be able to respond to each user uniquely, improving personalization.

Learning Mechanism: By incorporating machine learning or another learning mechanism, the chatbot will be able to continuously improve its comprehension of user preferences and changing conversational patterns.

5. Conclusion

This project, which uses the Python Tkinter module to create a talking AI chatbot, is impressive. It is a noteworthy accomplishment that the chatbot can react to user inputs, manage hellos and goodbyes, and conduct online searches by using the Google Custom Search API and the newspaper collection.

The user experience is improved by the user interface design, which includes an entry area for user input, a message button, a chat history display, and a right-side column to record the user's inquiries. The application's user-friendliness is further enhanced by the full-screen mode and the binding of the Enter key to the "Send" button.

A useful record of the discussion is provided by the display of chat history and user inquiry history, which may be advantageous to the user as well as the developer. In addition to giving the developer information into the user's interaction habits and preferences, it lets the user go back and examine previous questions and answers. The difficulty you had downloading content from websites and the way you used exception handling to handle these kinds of problems show how good you are at addressing problems and how dedicated you are to giving them a seamless experience. Notifying the user when material cannot be retrieved is a thoughtful method that improves the application's dependability and transparency. This project has a few possible directions for future development and customisation. For example, the rules governing the chatbot might be broadened to address more subjects and inquiries. Incorporating natural language processing (NLP) techniques may also improve the chatbot's comprehension and responsiveness to user inputs. More sophisticated capabilities like sentiment analysis, which would enable the chatbot to react differently depending on the user's tone or mood, might also be added to the programme. Additionally, by teaching the chatbot to learn from previous exchanges, it may eventually be able to offer more pertinent and individualised replies. By adding unique pictures, colours, and fonts, the application's user interface might be enhanced to seem better. Depending on the particular use case of the application, multimedia components like pictures, movies, or audio samples might potentially improve the chatbot's replies. To sum up, this project has established a strong basis for an AI chatbot that can conduct conversations and has an intuitive user interface. The chatbot may develop into a more powerful and adaptable tool with more enhancements and customisation, able to handle a greater variety of requests and deliver more interesting and tailored answers.

Acknowledgements

This research received support from our college, and we express gratitude to our colleagues at K L University for their valuable insights and expertise, even though they may not endorse all the conclusions of this paper. Special thanks to Dr. S. Sri Harsha , Professor & Project Supervisor, for enhancing the manuscript in the development of 'Conversational AI: A Comprehensive Study on Building and Enhancing Chatbot Systems.'

Author contributions

Hemendra Kumar Jain: Conceptualization, Methodology, Software, Field study **Kotla Veera Venkata Satya Sai Narayana :** Data curation, Writing-Original draft preparation **Shaik Asad Ashraf :** Software, Validation., Field study **Pendyala Venkat Subash :** Visualization, Investigation Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Qiu, L., Chen, Y., Jia, H., & Zhang, Z. Query Intent Recognition Based on Multi-Class Features. IEEE Access, 6, 52195-52204.

[2] Chen, J., Prasad, R., Stoyanchev, S., Selfridge, E., Bangalore, S., & Johnston, M. Corpus and Annotation Towards NLU for Customer Ordering Dialogs. 2018 IEEE Spoken Language Technology Workshop (SLT), Athens, Greece, 707-713.

[3] Razzaq, M. A., Khan, W. A., & Lee, S. Intent-Context Fusioning in Healthcare Dialogue-Based Systems Using JDL Model. Enhanced Quality of Life and Smart Living. ICOST 2017. Lecture Notes in Computer Science, vol 10461, Springer, Cham, 335-346.

[4] Dong, X., Qian, L., Guan, Y., Huang, L., Yu, Q., & Yang, J. A multiclass classification method based on deep learning for named entity recognition in electronic

medical records. 2016 New York Scientific Data Summit (NYSDS), New York, NY, 1-10.

[5] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

[6] Young, S. Statistical Spoken Dialogue Systems and the Challenges for Machine Learning. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining – WSDM'17, 303-312.

[7] Young, T., Hazarika, D., Poria, S., & Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing [Review Article]. IEEE Computational Intelligence Magazine, 13(3), 55-75.

[8] Juang, B. H., & Chen, T. The past, present, and future of speech processing. IEEE Signal Processing Magazine, 15(3), 24-48.

[9] Surabhi, M. C. Natural language processing future. 2013 International Conference on Optical Imaging Sensor and Security (ICOSS), Coimbatore, 1-3.

[10] Ek, T., Kirkegaard, C., Jonsson, H., & Nugues, P. Named Entity Recognition for Short Text Messages. Procedia - Social and Behavioral Sciences, 27, 178-187.

[11] A survey of named entity recognition and classification. Lingvisticæ Investigationes, 30(1), 3-26.

[12] McCallum, A., & Li, W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003.

[13] Dong, X., Qian, L., Guan, Y., Huang, L., Yu, Q., & Yang, J. A multiclass classification method based on deep learning for named entity recognition in electronic medical records. 2016 New York Scientific Data Summit (NYSDS), New York, NY, 1-10.

[14] Zheng, S., Zhang, J., Zhao, J., & Li, J. Joint Entity and Relation Extraction Based on a Hybrid Neural Network. Neurocomputing, 257, 59-66.

[15] Cuayahuitl, H., Renals, S., Lemon, O., & Shimodaira, H. Human-computer dialogue simulation using hidden Markov models. IEEE Workshop on Automatic Speech Recognition and Understanding, San Juan, 290-295.

[16] Mendoza, M., & Zamora, J. Building Decision Trees to Identify the Intent of a User Query. In: Velásquez, J. D., Ríos, S. A., Howlett, R. J., & Jain, L. C. (Eds.). Knowledge-Based and Intelligent Information and Engineering Systems. KES 2009. Lecture Notes in Computer Science, vol 5711, Springer, Berlin, Heidelberg, 393-402.

[17] Zhang, T., Cho, J. H. D., & Zhai, C. Understanding user intents in online health forums. Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics - BCB '14, 1-8.

[18] Meng, L., & Huang, M. Dialogue Intent Classification with Long Short-Term Memory Networks. Natural Language Processing and Chinese Computing, edited by Xuanjing Huang et al., Springer International Publishing, 42-50.

[19] Ilievski, V., Musat, C., Hossmann, A., & Baeriswyl, M. Goal-oriented chatbot dialog management bootstrapping with transfer learning. arXiv preprint arXiv:1802.00500.

[20] Zhang, Y., Zhao, J., & Wang, Y. A survey of natural language processing techniques for mental health analysis. Journal of Healthcare Engineering, 2020, 1-15.