UNITECH SELECTED PAPERS

INTERNATIONAL SCIENTIFIC CONFERENCE
20-22 November 2025, GABROVO

WEB CONTENT BASED CHATBOT CREATION

Nedim Džanić[1], Aleksandra Kuk [2], Dragan Savić [3], Petar Milić [2*]

[1]College of Business and Technical Education in Doboj, Ozrenskih srpskih brigada br. 5A, Doboj, Bosnia and Herzegovina
[2]University of Pristina – Kosovska Mitrovica, Faculty of Technical Sciences, Knjaza Milosa br. 7, Kosovska Mitrovica, Serbia
[3]University of Economics Academy in Novi Sad, Faculty of Applied Sciences, Visegradska 47, Nis, Serbia
**Corresponding author: petar.milic@pr.ac.rs*

Abstract
Nowadays, chatbots play a key role in providing fast and efficient customer support, facilitating interaction with users through automated responses and solving their problems in real time. In this research, we will focus on developing a hybrid WebGrab Chatbot (WGC) using natural language processing (NLP) techniques to generate responses based on web page content. We will also analyze the challenges encountered in chatbot development, such as grammar and response reliability.
Keywords: WebGrab, chatbot, natural language processing, artificial intelligence, machine learning.

INTRODUCTION

Chatbots, or conversational agents, facilitate human-computer interaction using natural language, powered by natural language processing (NLP) technology. As they can replicate human conversations and automate tasks, minimizing effort, chatbots are gaining popularity across various fields, such as healthcare, customer service, education, and academic counseling.

In today's world, information is a valuable resource, and finding accurate and precise information quickly is seen as a significant achievement [1]. With the vast number of websites and sources available, it's becoming increasingly difficult to find the right information in a short amount of time. Although many websites have built-in search engines, users still face challenges in selecting the right keywords to locate specific information. The ideal solution would be if every website had support available where users could directly ask questions about the content, rather than scrolling through large amounts of text to find what they need [2]. However, this would require an enormous team of support staff to be available around the clock.

Previously mentioned challenge led us to develop a solution that can independently learn the content of a webpage and provide users with relevant answers to their questions. Throughout the paper, we will describe the WebGrab Chatbot (WGC), which, unlike most chatbots in our region, independently processes textual content on the basis of which it defines the answer to the question posed by the user. In order to achieve this, it is necessary to equip the program with possibility how to process text and understand the grammar of the language.

WEBSITE CONTENT PROCESSING SYSTEM

To build an automatic question-answer system, it is crucial to ensure that all content

doi: www.doi.org/10.70456/.....

on the page is thoroughly processed and ready before the system can be activated [3]. This means that the content needs to be prepared before users can interact with the chatbot. To optimize the page effectively, it is important to ensure that only the updated content is processed, taking into account the latest changes made [4]. A practical approach is to implement all processing functions at the points where content is stored in the database, immediately after being added, modified, or removed. Any changes made during editing are also recorded in the database.

Using the approach mentioned earlier, the speed of the chatbot's response is enhanced, ensuring that users do not perceive any background processing. Additionally, this background processing is carried out using NLP through the Classla Python library. This library provides the essential grammar documentation for the content being processed. As a result, every word or symbol in the text receives its specific grammatical meaning [5], such as tokenization, lemmatization, part-of-speech tagging, dependency parsing, named entity recognition, and more, as illustrated in Figure 1.

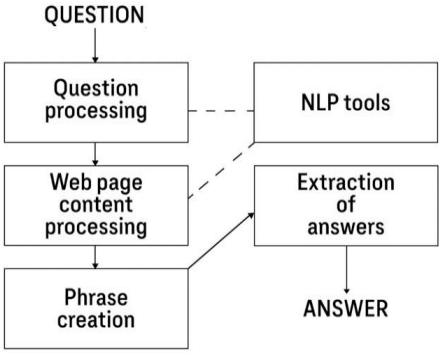

Fig. 1. *WebGrab Chatbot architecture*

Once all the required grammatical documentation for the content is obtained, we move on to the final processing, as shown in the example in Table 1. This process involves the following steps:

- reconstructing the entire sentence from the tokens
- replacing pronouns with the tokens they refer to
- segregating sentences into phrases
- storing the phrases in the database.

Table 1. *Example of creating of the phrases and replacement of the pronoun for the related token*

EXAMPLE OF PHRASE BUILDING AND THE SUBSTITUTE PRONOUN FOR THE TOKEN TO WHICH THEY REFER TO	
Original text	"Čvrnica is a solid stone on the right side of the Neretva, a harsh mountain massif, whose slopes guard over Jablanica, Drežnica, Diva Grabovica and Blidinje." "This mountain, when the winds blow, "calls" with its brothers Prenja and Vran and sister Čabulja, who protects Mostar from the ice breath from Čvrsnica."
Final editing	- Čvrsnica is a solid rock on the right side of the Neretva, a harsh mountain massif - *Čvrsnica* hillside guards over Jablanica, Drežnica, Diva Grabovica and Blidinje - *Mount Čvrsnica*, when the winds blow, "calls" with its brothers Prenja and Vrana and sister Čabulja - *Čabulja* protects Mostar from the icy breath of Čvrsnica

BUILDING A SYSTEM FOR AUTOMATIC QUESTIONS ANSWERING

When a user initiates a conversation with a chatbot, their message is sent to the backend via API. Before the system begins evaluating similarity and searching for an appropriate response, the user's message must first undergo through series of processing stages [6]. Our solution, as illustrated in Fig. 2, includes the following steps:

1. Case Sensitivity - The Jaccard similarity coefficient is case sensitive. This means that "Mountain" and "mountain" are considered different words, resulting in a

doi: www.doi.org/10.70456/.....

similarity coefficient of 0 between them. A straightforward solution to this issue is to convert all text to lowercase.

2. Stop Words Removal - To improve the Jaccard similarity coefficient, we need to eliminate frequently occurring or common words in questions. This includes removing punctuation, conjunctions, and other filler words typically used when asking a question.

3. Removing Diacritics - It's well-known that many people don't use diacritics like č, ć, đ, dž, š, and ž when typing on keyboards. To address this, we should convert all text and phrases into their non-diacritic equivalents.

4. Lemmatization - This process involves reducing words to their base or root form. For example, comparing "mountain" and "mountains" from a human perspective shows they refer to the same concept, expressed in singular and plural. However, without lemmatization, the computer would treat them as completely different words, resulting in a similarity of zero.

After message processing, the Jaccard coefficient method is applied to assess the similarity between the user's question and potential answers. In the first phase, predefined answers are checked. This means that if the user's message matches common greetings or queries such as "What's up?", "How are you?", or "Hello", the similarity is compared against a database of standard questions and answers. If the similarity fails to meet the defined threshold, the comparison continues with all stored phrases. Each phrase in the database undergoes the processing steps mentioned earlier and is then compared to the user's pre-processed message.

Although the similarity check is conducted for each phrase individually, the Jaccard coefficient is calculated and recorded. The phrase with the highest similarity score is considered the best candidate for the response, as long as its coefficient exceeds the predetermined threshold. For example, if the threshold is set to 0.20%, and the highest score falls below this, the user will receive a message indicating that no suitable answer was found.

```
Start
Query loading
Query – convert to lowercase
Query – remove diacritics

if (query = "what's up", "how are you", "Hello"...)
   Output = Search "DefaultAnswers.json"
   DefaultAnswers = convert to lowercase
   DefaultAnswers = remove diacritics
   Similarity = JaccardsSimilarityCoefficient(Query, DefaultAnswers)
   if (Similarity < Tolerance)
      Output = Search "phrases.json"
    else
      WriteOut = Output
else
   Output = Search "phrases.json"
   Query - remove stopwords
   Query - lemmatization
   Phrase - convert to lowercase
   Phrase - remove stopwords
   Phrase - remove diacritics
   Phrase - lemmatization
   Similarity = JaccardsSimilarityCoefficient(Query, Phrase)
   if (Similarity < Tolerance)
      WriteOut = "We are sorry, we don't have an answer to your question"
    else
      WriteOut = Output
    End
```

Fig. 2. *Pseudocode system for automatic answering of questions*

After each response, the user has the option to provide feedback by clicking on either the "I like" or "I don't like" button. It's important to highlight that each question, answer, and any feedback left are recorded in the database for future analysis and improvement.

The user's ability to express their opinion through "like" and "dislike" reactions serves as valuable feedback for the chatbot system [7]. These reactions can be collected and analyzed to better understand user satisfaction and assess the quality of the responses given. Applying machine learning techniques to user feedback enables the chatbot to automatically adjust its responses, enhancing its capacity to generate relevant and useful information [8]. For instance, machine learning algorithms can examine user reactions to uncover patterns and preferences, helping the chatbot deliver more personalized and satisfying answers. This ongoing user feedback can serve as a

doi: www.doi.org/10.70456/.....

foundation for the continuous improvement of the chatbot system.

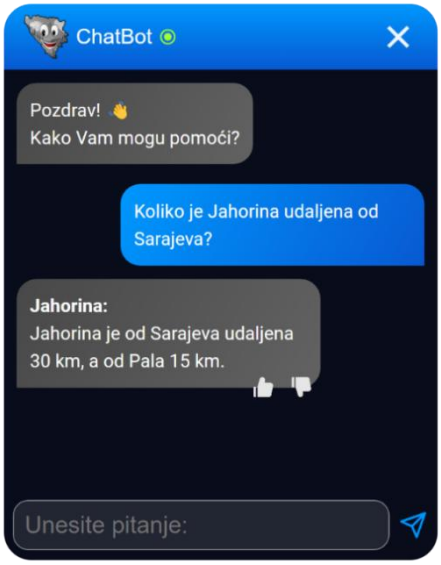

Fig. 3. *WGC chatbot*

COMPARATIVE ANALYSIS WITH EXISTING SOLUTIONS

Chatbots offer a smoother and more efficient browsing experience in the areas where they are implemented. Rather than sifting through numerous pages or searching for information manually, users can simply type their queries into a chatbot interface and receive real-time responses. This not only saves time but also boosts user engagement by providing instant support. In this section of the paper, we compare our solution with existing alternatives, highlighting its advantages and disadvantages. Additionally, Table 2 presents a comparison of features across three different types of chatbots.

Table 2. *Comparison of features between three different types of chatbots*

Characteristic	WGC Chatbot	GPT Chatbot	VPTŠ Chatbot
Chatbot type	Hybrid	GPT	Rule-based
Using rules and patterns for answer generation	Yes	No	Yes
Accuracy of response	It depends on the implemented rules and language processing	High	It depends on the exact match of the input with the content
Ability to process complex and ambiguous queries	Relatively good, but depends on rules and language processing	High	Limited, requires exact matching with rules
Generating new answers based on user questions	No	Yes	No
Ability to understand context and follow up on questions	Limited	No	Yes

GPT CHATBOT

The GPT chatbot[1] is currently one of the most advanced and widely used chatbots, gaining popularity for its ability to adapt to various domains and user needs. Based on the GPT model (Generative Pre-trained Transformer), which is one of the leading models in natural language processing, the GPT chatbot is a generative, advanced system. It leverages deep learning as its core method for processing data and can generate entirely new responses based on user queries [9]. When a user submits a question, the chatbot analyzes the input text and draws from its prior knowledge to generate a response. The model can understand context and consider previous parts of the conversation, ensuring it provides the most

[1] https://chatgpt.com/

doi: www.doi.org/10.70456/.....

relevant answer. In Table 3, we provide a comparison between the WGC and GPT chatbots through various examples.

Table 3. *Comparison of the WGC and GPT chatbot's through examples*

TYPE	TEXT	ANSWER
QUESTION: Koliko je Jahorina udaljena od Sarajeva?		
GPT Chatbot	X	Jahorina se nalazi oko 28 kilometara jugoistočno od Sarajeva.
WGC Chatbot	Jahorina je planina u Bosni i Hercegovini koja pripada Dinarskom planinskom sustavu. Najviši vrh je Ogorjelica s 1.916 m nadmorske visine. Jahorina je od Sarajeva udaljena 30 km, a od Pala 15 km. Ljeti je prekrivena gustom zelenom travom, a zimi i do 3 m visokim snijegom. Jahorina se prostire od 43°39' do 43°47' sjeverne zemljopisne širine i od 18°31' do 18°43 istočne zemljopisne dužine.	Jahorina je od Sarajeva udaljena 30 km, a od Pala 15 km.
QUESTION: A od Pala?		
GPT Chatbot	X	Jahorina je udaljena od Pala oko 22 kilometra, također jugoistočno od Pala.
WGC Chatbot	-‖-	Žao mi je, nisam našao odgovor na Vaše postavljeno pitanje...
QUESTION: Koliko je Jahorina udaljena od Pala i koji je najviši vrh?		
GPT Chatbot	X	Jahorina je udaljena od Pala oko 22 kilometra, također jugoistočno od Pala. Najviši vrh na planini Jahorina je vrh Ogorjelica, koji se nalazi na nadmorskoj visini od 1.916 metara.
WGC Chatbot	-‖-	Jahorina je od Sarajeva udaljena 30 km, a od Pala 15 km.

VPTŠ CHATBOT

The VPTŠ Chatbot[2] is a rule-based chatbot developed by the College of Business and Technical Education in Doboj. It operates by using predefined rules and patterns to generate responses to user inquiries. These chatbots work on a system designed to identify patterns in the user's input and provide a corresponding response based on those patterns [10]. This chatbot is available on the Viber social network,

[2] viber://pa?chatURI=vptsdo

doi: www.doi.org/10.70456/.....

making it more accessible for students of the institution to easily obtain relevant information during their studies.

One of the key advantages of this chatbot is the accuracy of its responses, as they are precisely as expected. However, its primary limitation lies in its inability to handle complex or unclear queries. Since it relies solely on predefined rules, it requires an exact match with those rules in order to generate a response.

CONCLUSION

The WGC Chatbot adopts a hybrid approach, merging the features of rule-based systems with natural language processing (NLP) techniques and elements of artificial intelligence. This combination enables the chatbot to deliver accurate answers while enhancing user engagement.

Its ability to process content from a webpage allows it to generate responses that are directly relevant to that content. Whether in e-commerce, travel, education, or other sectors, the chatbot is designed to be adaptive and responsive, providing users with the information they require.

A standout feature of WGC is its automatic learning capability. As the content on the webpage evolves, the chatbot updates its knowledge to reflect the new information. This ensures that users always receive the most current and accurate responses, even as content changes over time.

Given the significant role of artificial intelligence in chatbot development, we focus on integrating machine learning techniques to continually enhance system performance. By analyzing user feedback on our responses, we aim to improve the chatbot system iteratively, enabling it to learn from every interaction.

This approach allows us to gradually refine the chatbot, optimizing its responses and tailoring them to meet users' unique needs and preferences. Through ongoing learning from user interactions, we strive to create a chatbot that offers a personalized experience while effectively addressing users' queries and requests.

Acknowledgments: The authors would like to thank the Ministry of Science, Technological Development and Innovation of the Republic of Serbia for funding the scientific research work, contract no. 451-03-18/2025-03/200155, realized by the Faculty of Technical Sciences in Kosovska Mitrovica, University of Pristina.

REFERENCE

[1] Ibrihich, S., Oussous, A., Ibrihich, O., & Esghir, M. (2022). A Review on recent research in information retrieval. Procedia Computer Science, 201, 777-782.

[2] Guo, K., Defretiere, C., Diefenbach, D., Gravier, C., & Gourru, A. (2022, April). Qanswer: Towards question answering search over websites. In Companion Proceedings of the Web Conference 2022 (pp. 252-255).

[3] Deutsch, D., Bedrax-Weiss, T., & Roth, D. (2021). Towards question-answering as an automatic metric for evaluating the content quality of a summary. Transactions of the Association for Computational Linguistics, 9, 774-789.

[4] Kuk, K., Rančić, D., Pronić-Rančić, O., & Ranđelović, D. (2016). Intelligent agents and game-based learning modules in a learning management system. In Agent and Multi-Agent Systems: Technology and Applications: 10th KES International Conference, KES-AMSTA 2016 Puerto de la Cruz, Tenerife, Spain, June 2016 Proceedings (pp. 233-245). Springer International Publishing.

[5] Khandare, A., Agarwal, N., Bodhankar, A., Kulkarni, A., & Mane, I. (2023). Study of Python libraries for NLP. International Journal of Data Analysis Techniques and Strategies, 15(1-2), 116-128.

[6] Park, D. M., Jeong, S. S., & Seo, Y. S. (2022). Systematic review on chatbot techniques and applications. Journal of Information Processing Systems, 18(1), 26-47.

[7] Belda-Medina, J., & Calvo-Ferrer, J. R. (2022). Using chatbots as AI conversational partners in language learning. Applied Sciences, 12(17), 8427.

[8] Skrebeca, J., Kalniete, P., Goldbergs, J., Pitkevica, L., Tihomirova, D., & Romanovs,

doi: www.doi.org/10.70456/.....

A. (2021, October). Modern development trends of chatbots using artificial intelligence (ai). In 2021 62nd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS) (pp. 1-6). IEEE.

[9] Holderried, F., Stegemann–Philipps, C., Herschbach, L., Moldt, J. A., Nevins, A., Griewatz, J., & Mahling, M. (2024). A generative pretrained transformer (GPT)–powered chatbot as a simulated patient to practice history taking: prospective, mixed methods study. JMIR medical education, 10(1), e53961.

[10] 10. Solomon, E., & Tilahun, S. L. (2024). Rule based chatbot design methods: A review. Journal of Computational Science and Data Analytics, 1(01), 75-84.

doi: www.doi.org/10.70456/.....