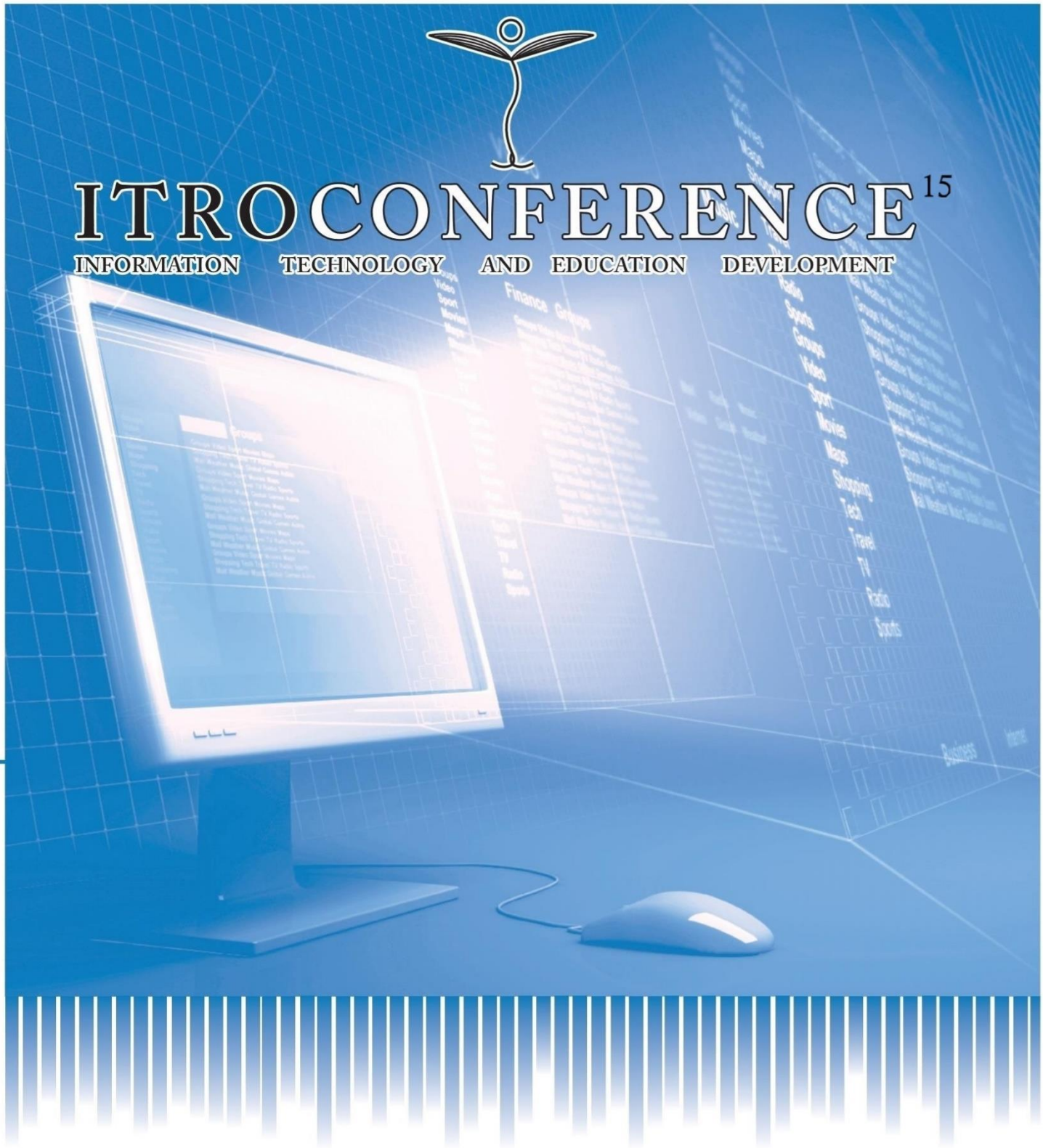
UNIVERSITY OF NOVI SAD
TECHNICAL FACULTY
"MIHAJLO PUPIN"
ZRENJANIN

ITROCONFERENCE[15]

INFORMATION TECHNOLOGY AND EDUCATION DEVELOPMENT



PROCEEDINGS

ZRENJANIN, November 2024

UNIVERSITY OF NOVI SAD
TECHNICAL FACULTY "MIHAJLO PUPIN"
ZRENJANIN
REPUBLIC OF SERBIA

XV INTERNATIONAL CONFERENCE OF

INFORMATION TECHNOLOGY AND DEVELOPMENT OF EDUCATION ITRO 2024

PROCEEDINGS OF PAPERS

XV MEĐUNARODNA KONFERENCIJA

INFORMACIONE TEHNOLOGIJE I RAZVOJ OBRAZOVANJA ITRO 2024

ZBORNIK RADOVA

ZRENJANIN, NOVEMBER 2024

Publisher and Organizer of the Conference:
University of Novi Sad, Technical faculty „Mihajlo Pupin", Zrenjanin, Republic of Serbia

For publisher:
Milan Nikolić, Ph. D, Professor,
Dean of the Technical faculty „Mihajlo Pupin", Zrenjanin, Republic of Serbia

Editor in Chief - President of OC ITRO 2024:
Jelena Stojanov, Ph. D, Associate Professor

Editors of Proceedings:
Marjana Pardanjac, Ph. D, Associate Professor
Jelena Stojanov, Ph. D, Associate Professor

Technical support:
Maja Gaborov MSc, Assistant
Nemanja Tasić MSc, Assistant
Dragica Radovanović MSc, Assistant
Katarina Vignjević MSc, Assistant

Circulation: **50**

ISBN: 978-86-7672-383-6

CIP - Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

37.01:004(082)
37.02(082)

INTERNATIONAL Conference on Information Technology and Development of Education ITRO (15 ; 2024 ; Zrenjanin)
Proceedings of papers [Elektronski izvor] / XV International Conference on Information Technology and Development of Education ITRO 2024 = Zbornik radova / XV međunarodna konferencija Informacione tehnologije i razvoj obrazovanja ITRO 2024, Zrenjanin, November 2024 ; [editors of proceedings Marjana Pardanjac, Jelena Stojanov]. - Zrenjanin : Technical Faculty "Mihajlo Pupin", 2024. - 1 elektronski optički disk (CD-ROM) ; 12 cm

Nasl. sa naslovnog ekrana. - Bibliografija uz svaki rad.

ISBN 978-86-7672-383-6

а) Информациона технологија - Образовање - Зборници б) Образовна технологија - Зборници

COBISS.SR-ID 159481865

PARTNERS INTERNATIONAL CONFERENCE

South-West University „Neofit Rilski”
Faculty of Education, Blagoevgrad,
Republic of Bulgaria

SOUTH WEST UNIVERSITY
"NEOFIT RILSKI"

Technical University of Košice
Faculty of Electrical Engineering and Informatics
Slovak Republic

University Goce Delcev Stip
Republic of Macedonia

THE SCIENCE COMMITTEE:

Milan Nikolić, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Sashko Plachkov, South-West Un. "Neofit Rilski"/Dep. of Education, Blagoevgrad, R. of Bulgaria
Nina Bijedić, Applied mathematics, Faculty of Informatics Bosnia and Herzegovina
Mirjana Kocaleva, Faculty of Informatics, University "Goce Delčev", Štip, North Macedonia
Gordana Jotanović, Univ. u Ist. Sarajevu, Saobraćajni fakultet, Doboj, Bosnia and Herzegovina
Dušan Starčević, Faculty of Organizational Sciences, Belgrade, Serbia
Mirjana Segedinac, Faculty of Science, University of Novi Sad, Serbia
Dragica Radosav, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Ivana Berković, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Dragana Glušac, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Marjana Pardanjac, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Vladimir Brtka, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Željko Stojanov, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Dalibor Dobrilović, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Zoltan Kazi, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Ljubica Kazi, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Đurđa Grijak, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Snežana Jokić, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Vesna Makitan, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Jelena Stojanov, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Đorđe Vučković, Technical Faculty “Mihajlo Pupin” Zrenjanin, University of Novi Sad, Serbia
Snežana Babić Kekez, Faculty of Sciences, University of Novi Sad, Serbia
Marina Čičin Šain, University of Rijeka, Croatia
Marta Takacs, Obuda University, John von Neumann Faculty of Informatics, Budapest, Hungary
Gordana Štasni, Faculty of Philosophy, University of Novi Sad, Serbia
Anja Žnidaršič, Faculty of Organizational Sciences, Kranj, University of Maribor, Slovenia
Janja Jerebic, Faculty of Organizational Sciences, Kranj, University of Maribor, Slovenia
Tatjana Grbić, Faculty of Technical Sciences, University of Novi Sad, Serbia
Slavica Medić, Faculty of Technical Sciences, University of Novi Sad, Serbia
Bojana Perić Prkosovački, University of Novi Sad, Serbia

THE ORGANIZING COMMITTEE:

Jelena Stojanov, Ph.D, Associate Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia - Chairman of the Conference ITRO 2024

Dragica Radosav, Ph.D, Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Dragana Glušac, Ph.D, Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Vesna Makitan, Ph.D, Associate Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Marjana Pardanjac, Ph.D, Associate Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Snežana Jokić, Ph.D, Assistant Professor, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Nemanja Tasić, M.Sc, Assistant, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Maja Gaborov, M.Sc, Assistant, Technical Faculty “M. Pupin” Zrenjanin, UNS, Serbia
Dragica Radovanović, M.Sc., Assistant, Technical Faculty “Mihajlo Pupin” Zrenjanin, UNS, Serbia
Katarina Vignjević, M.Sc., Assistant, Technical Faculty “Mihajlo Pupin” Zrenjanin, UNS, Serbia

LIST OF REVIEWERS:

Biljana Radulović, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Bojana Perić Prkosovački, Faculty of Medicine, University of Novi Sad, Serbia
Caba Sabo, Technical University of Košice “Faculty of Electrical Engineering and Informatics”, Slovak Republic
Dalibor Dobrilović, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Darko Radovančević, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Dragan Ćoćkalo, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Dragana Glušac, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Dragica Radosav, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Đorđe Vučković, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Đurđa Grijak, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Edit Terek Stojanović, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Elena Karamazova Gelova, "Goce Delčev" University, Stip, North Macedonia
Eleonora Brtka, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Goran Jausevac, University of East Sarajevo, Bosnia and Herzegovina
Gordana Jotanović, University of East Sarajevo, Bosnia and Herzegovina
Ivan Palinkaš, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Ivana Berković, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Jelena Stojanov, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Ljubica Kazi, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Marjana Pardanjac, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Mihalj Bakator, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia

Mila Kavalić, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Mirjana Kocaleva, "Goce Delčev" Stip, University, North Macedonia
Nadežda Ljubojev, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Ninoslav Kačarić, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Snežana Jokić, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Tanja Sekulić, Technical College of Applied Sciences in Zrenjanin
Vesna Makitan, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Višnja Ognjenović, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Vladimir Brtka, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Zoltan Kazi, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia
Željko Stojanov, Technical Faculty "Mihajlo Pupin" Zrenjanin, University of Novi Sad, Serbia

The Proceedings have been published in a digital format on the Faculty web site.

INTRODUCTION

This Proceedings present the articles delivered at the international conference Information Technology and Education Development (ITRO 2024), held for the jubilee fifteenth time on November 29, 2024. This international event was conducted in a hybrid format, combining in-person and online participation. The conference continues its tradition of bridging science, professional practice, and educational experiences, with this year's focus on the conditions and perspectives of teachers' digital competencies.

The thematic fields of the conference reflect contemporary trends in education, addressing topics such as: the digitalization of education, education in crisis situations, educational challenges, theoretical and methodological issues in contemporary pedagogy, digital didactics and media, modern communication strategies in teaching, curriculum development for contemporary education, advancements in e-learning, education management practices, methodological approaches in teaching natural and technical sciences, and the integration of information and communication technologies in education.

The conference featured three plenary lectures that explored various aspects of the main topic, with the corresponding articles included at the beginning of this volume.

In total, this edition comprises 57 peer-reviewed articles, evaluated through a double-blind review process. These contributions represent the latest research and advancements in the field.

The conference received financial support from the Provincial Secretariat for Higher Education and Scientific Research, Novi Sad. Hosting and technical support were generously provided by the Technical Faculty "Mihajlo Pupin." We extend our sincere gratitude for this invaluable assistance.

The Organizing Committee expresses its heartfelt thanks to the authors, reviewers, and participants for their contributions, which ensure the success and continued tradition of this event.

We look forward to welcoming you to the next ITRO Conference!

On behalf of the ITRO Organizing Committee

Jelena Stojanov

CONTENTS

PLENARY PAPERS

SCIENTIFIC PAPERS

SCIENTIFIC

PAPERS

Python Chatbot: Virtual Assistant in Educational Process

Vitanova M. K.[*1], Stojkovik N.[*2], Bande C. M.[*3], Arnautova K.[*4] and Zlatanovska B.[*5]

[*]Faculty of computer science, Goce Delcev University, Stip, North Macedonia
[1]mirjana.kocaleva@ugd.edu.mk,
[2]natasa.maksimova@ugd.edu.mk,
[3]cveta.martinovska@ugd.edu.mk,
[4]kamelija.102218@student.ugd.edu.mk,
[5]biljana.zlatanovska@ugd.edu.mk

Abstract. This study explores the evolution and significance of chatbots in the context of artificial intelligence (AI). We trace the historical development of AI and the emergence of chatbots, examining their core features, functionalities, and applications across various sectors, particularly in education. The research includes a hands-on exploration of the chatbot creation process, demonstrating how these systems can be customized for diverse educational purposes. Our findings reveal that chatbots can significantly enhance human-computer interaction and improve learning experiences by providing instant support, personalized resources, and facilitating real-time feedback. We conclude with recommendations for future advancements in chatbot technology, emphasizing their potential to transform educational environments.

Keywords and phrases: Artificial Intelligence, Chatbot Development, Educational Technology, Human-Computer Interaction.

1 INTRODUCTION

Since the advent of industrialization and the first steam engine, society has shown a continuous drive to replace manual labor with mechanized solutions. In today's advanced technological landscape, chatbots have become essential tools, helping us navigate modern technological challenges. The origins of chatbots trace back to the 1960s with the development of ELIZA. Over the years, chatbots have been known by various names, including interactive agents, virtual assistants, and virtual humans—terms that, despite nuanced differences, are often used interchangeably.

Chatbots have since permeated all areas of modern life, stepping in wherever the human element can be supplemented or replaced. As AI-powered tools, chatbots leverage natural language processing (NLP) and natural language understanding (NLU) to accurately interpret user queries, adapting responses based on specific intents, while machine learning allows for real-time optimization. With AI and deep learning, chatbots continuously improve their accuracy, fostering natural and seamless interactions with minimal misunderstandings.

Historically, psychological and mathematical theories have contributed to the development of chatbots as we know them today. Unlike human employees who are bound by working hours and holidays, chatbots are available around the clock, addressing customer needs promptly. This instant availability improves customer satisfaction by reducing wait times and boosts economic efficiency by lowering operational costs. Chatbots can handle multiple users simultaneously and streamline support processes like answering calls, emails, or other online inquiries, ultimately enhancing service quality and reducing delays.

The paper by Adamopoulou and Moussiades (2020) provides a comprehensive review of chatbot design, architecture, and algorithms, discussing their functionality and evolution. It examines chatbot history, evaluation methods, and essential components such as ASR, NLP, response generation, and dialogue management. The paper also includes a case study on IBM Watson, covering security aspects and various

applications. This literature review highlights the historical progression, identifies limitations at different stages, and offers a categorization system. Additionally, it analyzes key technologies like pattern matching and machine learning, proposes a general architecture, and addresses crucial design considerations. The review explores applications, industry use cases, risks, and strategies for improvement, concluding with a forward-looking perspective on creating smarter chatbots.

Cahn's paper (2017) reviews chatbot design, architecture, and algorithms, including an analysis of chatbot functionality, history, and assessment methods. It discusses core elements like ASR, NLP, response generation, knowledge base development, dialogue management, and text-to-speech algorithms, concluding with a section on chatbot development, a case study on IBM Watson, and insights on security and applications.

Wei, Yu, and Fong (2018) present a framework for chatbot development, tracing AI's evolution and focusing on chatbot technology. They discuss chatbot history, features, and supportive technologies, and demonstrate the framework's viability for industrial use based on theoretical designs.

Rahman, Al Mamun, and Islam's (2017) paper examines cloud-based chatbot technologies, including IBM Watson and Microsoft Bot, while highlighting design and programming challenges due to the integration of AI, NLP, and dynamic functions. The review addresses current obstacles in chatbot development and anticipates future challenges.

Colace et al. (2018) focus on a prototype chatbot aimed at supporting university students in specific courses. Utilizing NLP and domain-specific ontologies, the chatbot interprets student queries and provides accurate answers. After implementation, an experimental study confirmed the model's efficiency and effectiveness.

Georgescu (2018) explores how conversational agents enhance daily life and education by offering accessible assistance across devices. In the educational field, chatbots present new methods for content delivery, student assessment, and real-time feedback, aligning with modern learning styles. This paper covers educational chatbot development, frameworks, benefits, challenges, and proposes a model to improve quality of life in educational contexts.

Building on the current importance of chatbots, this paper reviews their history, functionality, and modern applications, offering illustrative examples of the development process.

2 THE PROCESS OF CREATING A CHATBOT

To better explain the coded part, we will start from where the whole idea began. Initially, a file was created from which all responses would later be generated. A JSON file was structured to include metadata for training the chatbot, consisting of an intents list that contains tags, patterns, and responses. When the client inputs text, the bot can identify the relevant tag and classify the input accordingly. For instance, if the bot recognizes a greeting, it will provide a random response from a set of predefined replies. To enhance the training of the chatbot, we employ tokenization, a process that breaks down words, punctuation marks, and numbers into smaller, manageable units. For example:

"What cake flavors do you offer?"

[“What”, “cake”, “flavors”, “do”, “you”, “offer”,”?”]

Then, we will add another NLP technique that finds the root of the word (stemming). For example:

“organize”, “organizes”, “organizing”

[“organ”, “organ”, “organ”]

After applying the previously mentioned techniques, the process for handling a full sentence would be as follows.

For example, if we have the sentence: "Can I request custom cake designs?"

1.Tokenization will first split all the words: [“Can”, “I”, “request”, “custom”,” cake”,” designs”, “?”]

2.Then, all uppercase letters will be converted to lowercase: [“can”, “i”, “request”, “custom”,” cake”,” designs”, “?”]

3.Next, we will apply stemming to find the root of the words: [“can”, “i”, “request”, “custom”,” cake”,” design”, “?”]

4.After passing through the NLP techniques for tokenization and stemming, punctuation marks will be removed: [“can”, “i”, “request”, “custom”,” cake”,” design”]

5.Finally, we obtain a bag of words represented in binary values: [0, 0, 0, 1, 0, 1, 0, 1]

This binary representation indicates the presence or absence of specific words in the sentence, which helps the chatbot process and understand the input effectively.

4.1. Code Examples and Explanation

We begin by creating a file named nltk_utils.py, where we import the necessary libraries: numpy and nltk. We also import the PorterStemmer class from nltk.stem.porter. The constructor of the PorterStemmer class is then used to create an object, which is stored in the variable stemmer. This object will be used later to find the root (or "stem") of words.

```
numpy as np
import nltk
# nltk.download('punkt')
from nltk.stem.porter import PorterStemmer
stemmer = PorterStemmer()
```

The PorterStemmer is part of the Natural Language Toolkit (NLTK) and is a common technique in NLP (Natural Language Processing) for reducing words to their base or root form, making it easier to analyze text by reducing variations of words to a standard form.

```
def tokenize(sentence):
    return nltk.word_tokenize(sentence)
def stem(word):
    return stemmer.stem(word.lower())
```

The two functions above are used for tokenization and stemming of words, respectively:

tokenize(sentence):This function takes a sentence as input and splits it into individual words (tokens) using nltk.word_tokenize(). Tokenization is essential in NLP as it breaks down the input text into manageable units for further processing.

stem(word):This function stems from a given word by first converting all uppercase letters to lowercase using word.lower(). It then applies the stemmer.stem() method to reduce the word to its root form. This step is crucial for ensuring that words like "designs" and "design" are treated as the same base word.

These functions help preprocess the input by normalizing and simplifying the words, making further analysis more effective.

```
def bag_of_words(tokenized_sentence, words):
    # stem each word
sentence_words = [stem(word) for word in tokenized_sentence]
    # initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
```

```
            if w in sentence_words:
                bag[idx] = 1
        return bag
```

The bag_of_words function is used to convert a tokenized sentence into a numerical representation known as a "bag of words." This representation is crucial for processing text data for machine learning models. Here's how it works:

1. Tokenization and Stemming:sentence_words contain the stemmed versions of the words from the tokenized sentence. This ensures consistency in word forms (e.g., "design" for "designs").
2. Initialization:bag is initialized as a numpy array of zeros with a length equal to the number of words in the vocabulary. Each element in this array represents whether a specific word from the vocabulary is present in the input sentence.
3. Updating the Bag:The function iterates through the vocabulary (words). For each word in the vocabulary, if it appears in the sentence_words, the corresponding index in the bag is set to 1. This binary representation indicates the presence or absence of each vocabulary word in the sentence.

This function helps convert text data into a format suitable for machine learning algorithms, enabling the chatbot to classify and respond to different inputs effectively.

4.2. Training the Chatbot Data

The chatbot is trained using the train.py file. Here’s how the process works:

1. Loading and Reading Data:The script begins by opening and reading the JSON file, which contains the responses and associated metadata. This file includes the intents, tags, patterns, and responses used to train the chatbot.
2. Creating Lists:Several lists are created to store the "bag of words," tags, and patterns. These lists are populated using the data from the intents key in the JSON file.The previously defined functions for tokenization and stemming are used to process the input sentences, ensuring they are in a format suitable for training.
3. Indexing and Searching:A class with a constructor is created to enable indexing and searching by index during the training process. This class helps manage and retrieve data efficiently.
4. Model Creation:A model is created to train the chatbot using a neural network. The model includesInput Layer: Accepts the bag of words representation of the input sentence.Hidden Layers: Several hidden layers process the input and extract relevant features.Output Layer: Provides a probability distribution over possible class (i.e., intents).
5. Training the Model:The model is trained using the prepared data. Each input sentence is processed through the network, producing output probabilities that represent the likelihood of each intent class. This helps the chatbot determine the most appropriate response based on the input.
6. Forward Propagation:The forward function is used to activate all layers of the neural network during training and inference. This function computes the output by passing input data through the network layers, updating weights, and generating the final class probabilities.
7. Adding the Model and GPU Support:Returning to the train.py file, we incorporate the model and check for GPU support. The training process includes optimization and loss calculation to ensure accurate responses. Finally, the data is saved.
8. Creating chat.py:In the chat.py file, necessary libraries are loaded. GPU support is checked, the JSON file and the previously saved training data are loaded, and the model’s status is assessed.The chatbot interface is developed, including naming the bot and displaying introductory text. A while loop is used to import sentences, with an if condition to end the conversation.Sentences are tokenized and converted into the bag of words format, which the bot can recognize. A function is defined to generate responses with a 75% accuracy rate, based on prediction probabilities.For example, if "Hi" is entered, the bot searches for tags that might include "Hi" as a greeting. Once the appropriate tag is identified, the bot selects a response from predefined options.
9. Routing and Flask Application:A routing file is created to read the basic HTML file, enabling communication with the chatbot. This Flask application handles the exchange of sentences between the

bot and the user, modifying text into JSON format.When the code is activated, it opens a window and handles interactions like opening, closing, and clicking the SEND button through event-driven functions.A function selects the text input area, sends the sentences via POST requests to the prediction function, converts the text into a string, and specifies the JSON file format.After sending a POST request, the response is extracted and used to update the chat box. Error handling is also included.

10. Training Script Example:The examples below show how to train the model by calling the train.py script, with detailed information about the tags and the time required to find them.

4.3. Python-based chatbot as a virtual assistant for the educational process

Creating a Python-based chatbot as a virtual assistant for the educational process can greatly enhance both teaching and learning experiences. Here's an outline of how a Python chatbot can function as an educational assistant and some key elements it should include:

- Student Q&A Support: The bot can be designed to answer student queries about course materials, assignments, schedules, and other frequently asked questions (FAQs). This feature helps students get instant support, saving time for both students and educators.
- Resource Recommendations: Based on the student's queries or topics of interest, the bot can recommend relevant resources such as articles, videos, or practice exercises to help them deepen their understanding.
- Study Reminders and Notifications: The chatbot can remind students of upcoming deadlines, exams, and scheduled classes or events, helping them stay organized and manage their time effectively.
- Tutoring and Concept Reinforcement: By integrating natural language processing (NLP) and machine learning, the bot can tutor students on certain topics by providing explanations, examples, and quizzes for practice, reinforcing learning outside of class hours.
- Assessment and Feedback: The bot can conduct simple assessments through quizzes or flashcards, giving instant feedback to help students identify strengths and areas for improvement.
- Administrative Assistance: Beyond academic support, the bot can assist with administrative tasks, such as providing information on enrollment processes, campus facilities, or links to relevant forms and documents.

Some of the essential building blocks of a Python-based educational chatbot include:

- Natural Language Processing (NLP): NLP enables the chatbot to understand and interpret user input effectively, facilitating smooth interaction.
- Intent Recognition and Response Generation: This component helps the bot identify user intents (e.g., requesting resources, asking questions) and generate appropriate responses.
- Integration with Educational Resources: Connecting the bot to an educational knowledge base allows it to offer relevant resources, such as articles, videos, or course materials, to students.
- User Interface: An accessible interface, whether for desktop or mobile devices, is crucial for ease of use and convenience.
- Personalization and User Profiles: The chatbot can be programmed to recognize individual students, track their progress, and provide tailored responses, thereby enhancing the learning experience.

3 THE APPLICATION OF CHATBOTS IN EDUCATIONAL PROCESS

The application of chatbots in education enhances learning by providing instant, accessible support, helping students with information retrieval, lesson reviews, and practice exercises. Chatbots can deliver personalized assistance, answer questions, and engage students in interactive learning activities. They also aid teachers by automating routine tasks like grading and attendance, allowing more focus on student engagement. By simulating human-like conversations, educational chatbots foster a dynamic learning environment that complements traditional teaching methods, supports self-paced learning, and promotes continuous assessment and feedback.

A developed chatbot can be used as an assistant or learning aid in various ways to enhance the educational experience for both students and teachers. Here are some possible applications:

1. Answering Student Queries: The chatbot can serve as an on-demand resource, answering frequently asked questions (FAQs) related to course material, assignments, deadlines, or administrative issues. This frees up teacher time and ensures students get immediate support.
2. Tutoring and Concept Reinforcement: By guiding students through specific topics or exercises, the chatbot can help reinforce learning. It can provide explanations, examples, or even quizzes to ensure students understand concepts effectively.
3. Personalized Study Plans: Based on students' progress, the chatbot could offer personalized study plans, recommending resources, practice problems, or even adjusting topics based on students' understanding and performance in assessments.
4. Feedback and Progress Tracking: The chatbot can keep track of students' learning progress and provide personalized feedback on their performance in assignments, quizzes, or assessments. This helps students identify areas for improvement in real time.
5. Interactive Study Companion: Through quizzes, flashcards, or mini challenges, the chatbot can help students review and memorize information, making the learning process more engaging and interactive.
6. 24/7 Support and Assistance: Unlike human educators, chatbots are available around the clock, providing support at any time. This availability allows students to get help with their studies outside of regular class hours, especially beneficial in different time zones or for students studying remotely.
7. Automated Assessment and Grading: The chatbot can assist in grading quizzes and assignments, especially for objective-type questions. This automation can reduce grading workload and provide instant feedback to students.
8. Class Reminders and Notifications: It can remind students of upcoming deadlines, exams, or class schedules, keeping them organized and on track.
9. Language Practice and Improvement: For language courses, the chatbot can engage students in conversation practice, helping them improve vocabulary, grammar, and fluency. It can correct mistakes and suggest improvements in real-time.
10. Group Project Coordination: The chatbot can facilitate communication and coordination among group members, assign tasks, and set deadlines, helping students manage collaborative projects effectively.
11. Resource Recommendations: Based on a student’s learning needs or topics of interest, the chatbot can suggest additional learning resources like articles, videos, or exercises, expanding students' knowledge outside of class materials.
12. Exam Preparation and Mock Tests: To aid students in exam preparation, the chatbot could offer mock tests, track results, and provide tailored feedback on how students can improve.
13. Teacher Assistance: For teachers, the chatbot can assist by providing quick information about student progress, helping create reports, and tracking attendance. It can also alert teachers to students who may need extra help based on chatbot interactions.
14. Motivation and Encouragement: The chatbot can send motivational messages or study tips, encouraging students to stay engaged and productive, which can be especially helpful in long-distance or online learning environments.
15. Real-Time Polling and Feedback Collection: During live classes, the chatbot can be used to conduct polls, gather feedback, or engage students in discussions, providing teachers with instant insights into students' understanding and preferences.

By integrating a chatbot into the learning process, educational institutions can create a more interactive, responsive, and supportive environment that aligns with modern student needs and teaching practices.

4 Conclusion

In today's digital and highly interconnected educational environment, Python-based chatbots provide significant opportunities to enhance learning experiences by automating support, offering real-time resources, and facilitating self-paced learning. Starting from foundational concepts like Natural Language Processing (NLP), intent recognition, and personalized user profiling, chatbots in education represent a shift toward more responsive and flexible educational support systems. By leveraging essential chatbot building blocks, such as a user-friendly interface, educational resource integration, and customized student interactions, these tools offer tailored learning assistance that adapts to individual student needs and learning styles.

The implementation of educational chatbots can transform the teaching and learning process, providing accessible, round-the-clock support while allowing educators to focus on more complex pedagogical tasks. Through applications like tutoring, progress tracking, and interactive quizzes, chatbots can assist students in mastering content and reinforcing concepts outside traditional classroom hours. Moreover, by automating routine administrative tasks and facilitating class reminders and group coordination, chatbots contribute to streamlined educational processes and improved overall efficiency.

As AI and machine learning capabilities continue to advance, the role of educational chatbots will likely expand, offering even more sophisticated functionalities. By fostering dynamic, student-centered interactions, educational chatbots can create a more engaging and productive learning environment, thus enhancing both student satisfaction and educational outcomes. With continued development and adaptation to meet evolving academic needs, chatbots are poised to become an integral component of the modern educational landscape.

References

Adamopoulou, E., &Moussiades, L. (2020). An overview of chatbot technology. In IFIP international conference on artificial intelligence applications and innovations (pp. 373-383). Springer, Cham.

Adamopoulou, E., &Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with applications*, *2*, 100006.

Cahn, J. (2017). CHATBOT: Architecture, design, & development. *University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science*.

Colace, F., De Santo, M., Lombardi, M., Pascale, F., Pietrosanto, A., & Lemma, S. (2018). Chatbot for e-learning: A case of study. *International Journal of Mechanical Engineering and Robotics Research*, *7*(5), 528-533.

Černý, M. (2022). The history of chatbots: the journey from psychological experiment to educational object. *Journal of Applied Technical and Educational Sciences*, *12*(3), 322-322.

Georgescu, A. A. (2018). Chatbots for education–trends, benefits and challenges. In *Conference proceedings of» eLearning and Software for Education «(eLSE)* (Vol. 14, No. 02, pp. 195-200). Carol I National Defence University Publishing House.

Rahman, A. M., Al Mamun, A., & Islam, A. (2017). Programming challenges of chatbot: Current and future prospective. In *2017 IEEE region 10 humanitarian technology conference (R10-HTC)* (pp. 75-78). IEEE.

Wei, C., Yu, Z., & Fong, S. (2018, February). How to build a chatbot: chatbot framework and its capabilities. In *Proceedings of the 2018 10th international conference on machine learning and computing* (pp. 369-373).