IJMRSET

e-ISSN:2582-7219

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

Volume 7, Issue 7, July 2024

ISSN
INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 7.521

| ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 7.521| Monthly, Peer Reviewed & Referred Journal|

| Volume 7, Issue 7, July 2024 |

| DOI:10.15680/IJMRSET.2024.0707084 |

VS Code-Voice Companion for Physically Disabled People using NLP

Dr. Charles Arockiaraj M, Spoorthi U

Associate Professor, Department of MCA, AMC Engineering College, Bangalore, Karnataka, India

Student, Department of MCA, AMC Engineering College, Bangalore, Karnataka, India

ABSTRACT: The rapid advancement of natural language processing (NLP) technology has opened up new possibilities for human-computer interaction, particularly in the domain of programming. This report presents the project titled "VS Code- Voice Companion for Physically Disabled People using NLP," aimed at facilitating code writing and execution through voice input and natural language understanding. The project encompasses a Flask-based web application that serves as the interface system for users to interact. Upon accessing the application, users are prompted to provide input such as desired programming language and preferred method of code generation. The available programming languages include Python, Java, and JavaScript, offering flexibility to cater to diverse developer needs. Two distinct pathways are provided for code generation: automated code generation using ChatGPT and manual code input with support for NLP techniques such as tokenization, lexer, parser, and stemming. The automated code generation leverages the power of ChatGPT API to interpret natural language descriptions of code requirements and generate corresponding code snippets. Then the manual input pathway allows users to express their code logic in natural language, which is then processed using NLP techniques to produce executable code. Key features of the project include voice input support, real-time code execution within the VS Code environment, and seamless integration of NLP capabilities for both automated and manual code generation pathways. The system aims to streamline the code writing process, reduce cognitive load for developers, and promote accessibility by enabling code composition through natural language interactions. The project represents an innovative fusion of NLP technology with software development practices, offering a novel approach to code writing and execution. The report discusses the technical implementation, challenges encountered, and future directions for enhancing the system's functionality and usability.

KEYWORDS: Natural Language Processing (NLP), Voice Commands, Automated Code Generation, ChatGPT, Flask, VS Code Integration, Machine Learning, Programming Languages, User Interface.

I. INTRODUCTION

In recent years, Natural Language Processing (NLP) technology has made significant strides in enabling computers to understand and respond to human language in increasingly sophisticated ways. This improvement only transformed how we interact with digital assistants and search engines but has also opened up different ways for enhancing productivity and accessibility in various domains, including software development. Writing and executing code traditionally requires a high level of technical proficiency and familiarity with programming languages and syntax. However, those new to coding or with limited technical background, this can present a significant barrier to entry. Additionally, even experienced developers may face challenges when expressing complex coding concepts or debugging code efficiently. To address these challenges and make programming more accessible and intuitive, we propose the project titled " VS Code- Voice Companion for Physically Disabled People using NLP." This project aims to leverage the power of NLP technology to enable users to write and execute code using natural language interactions, thereby reducing the learning curve and coding practices. The project centers around a Flask-based web application that serves as the user interface for interacting with the system. User can provide the voice input and select their desired programming language, including popular languages such as Python, Java, and JavaScript. Additionally, users have the option to choose between automated code generation using ChatGPT or manual code input with support for NLP techniques. By integrating voice input and NLP capabilities, the project aims to democratize coding by allowing individuals to express their programming logic in a manner that feels natural and intuitive. Whether through conversational interactions with ChatGPT or through manual input augmented by NLP techniques, users can articulate their coding requirements in plain language, thereby bridging the gap between human intent and machine-executable instructions.

II. PROBLEM STATEMENT

Traditional methods of writing and executing code require a steep learning curve and proficiency in programming languages, which can act as a barrier to entry for beginners or individuals with limited technical background. Expressing complex coding concepts or debugging code efficiently and also challenging for experienced developers. The Need for a more intuitive and accessible approach to coding that accommodates diverse skill levels and learning styles.

III. LITERATURE SURVEY

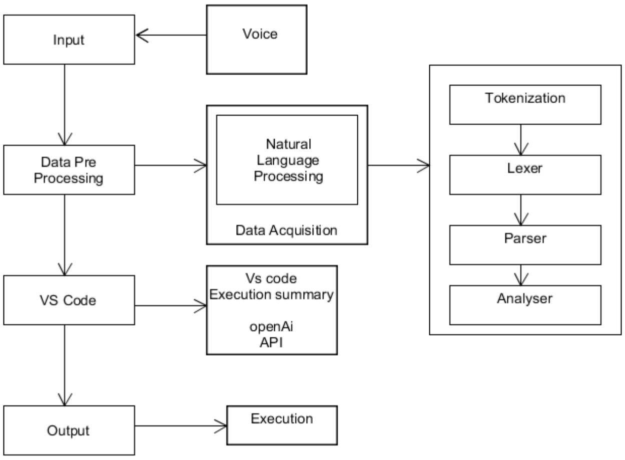
1. Analysis of Machine Code Using Natural Language Processing (N. Khurpia, 2021): This research investigates the application of NLP techniques in analysing machine code. It explores how NLP can enhance the understanding and interpretation of low-level programming constructs. This study is crucial to learn the potential implications of NLP in decoding and comprehending intricate machine-level instructions.
2. Automation using Artificial Intelligence Based Natural Language Processing (P. Mohana et al., 2022): This paper explores the synergy between AI and NLP for automation purposes. It employs AI algorithms and NLP techniques to automate tasks based on natural language instructions. This research is vital for understanding how AI, coupled with NLP, can automate various processes, offering insights into potential methodologies applicable to the proposed system.
3. Voice Assistant Using Artificial Intelligence (Indukuri Manikanta Sai Varma, Kalidindi Pranith Varma): This work focuses on the improvement of voice assistant using AI. It involves designing and training an AI model capable of processing voice commands and generating relevant responses. This research is pertinent to the proposed system's goals of incorporating voice commands for coding.
4. Code Generator Based on Voice Command for Multiple Programming Languages (S. Hossain et al., 2021): This research introduces a code generator based on voice commands supporting multiple programming languages. It involves creating a versatile code generation system capable of interpreting voice commands and producing code snippets in various programming languages.
5. Speech Coding and Speech Recognition Technologies: A Review (A. S. Spanias and F. H. Wu, 1991): This paper offers a comprehensive review of speech coding and speech recognition technologies. While not directly related to voice-based coding, it provides valuable insights into foundational speech technologies, informing the development of voice-based coding systems.
6. VSCODE – Code with Voice Using NLP (M. G. Prakash et al., 2022): This project enables coding with voice using NLP, converting speech to text and generating code snippets. It highlights the integration of NLP and voice commands in coding platforms.
7. Voice Controlled Automation System (M. S. Haleem, 2008): This project focuses on a voice-controlled automation system, providing insights into voice control technologies and methodologies applicable to the proposed system.

IV. METHODOLOGY

1. Setup Development Environment:
   a. Install Python and Flask Framework: Install Python and Flask for backend development.
   b. Set Up a Virtual Environment: Create a virtual environment to manage dependencies.
   c. Install Necessary Libraries: Install libraries such as Flask, requests, and python-dotenv.
   d. Configure Flask Application Settings: Set up Flask application settings and environment variables.
2. Create Flask Application:
   a. Create Flask Application Structure: Establish a new Flask application structure with appropriate directories.
   b. Define Routes and Endpoints: Map URL paths to view functions using decorators.
   c. Implement Error Handling: Implement basic error handling and exception handling mechanisms.
3. Voice Input Processing:
   a. Integrate Google Cloud Speech-to-Text API: Use Google Cloud's API to process voice input.
   b. Develop Modules for Audio Input: Create modules to capture audio input and send it to the API.
   c. Handle Transcription Responses: Implement logic to handle transcription responses.
4. Natural Language Processing (NLP):
   a. Implement NLP Algorithms: Use libraries such as NLTK or spacy for NLP tasks.
   b. Develop Functions for NLP: Create functions for tokenization, parsing, and interpreting user intents.

   c. Translate User Intents: Utilize NLP techniques to understand user intents and generate executable code snippets.

5. Automated Code Generation (ChatGPT):
   a. Integrate OpenAI's GPT Models: Use OpenAI's GPT models for code generation.
   b. Develop Interaction Modules: Create modules to interact with the GPT API.
   c. Format Generated Code: Handle responses from the API and format the generated code snippets.
6. Manual Code Input with NLP Techniques:
   a. Implement Manual Code Input Functionality: Allow users to input code manually.
   b. Parse and Interpret Code Descriptions: Use NLP techniques to parse and interpret user-provided code descriptions.
   c. Generate Executable Code: Develop logic to generate executable code based on the interpreted input.
7. Integration with VS Code:
   a. Create Custom VS Code Extension: Use the VS Code extension framework for real-time code execution.
   b. Develop Execution Modules: Implement modules to execute code snippets within VS Code.
   c. Capture and Display Output: Display the output within the web application interface.
8. User Interface Development:
   a. Design and Develop Web Interface: Create a user-friendly web interface using HTML, CSS, and JavaScript.
   b. Create Forms and Input Fields: Develop forms for user input.
   c. Implement Interactive Elements: Add buttons, dropdowns, and text areas for user interaction.



V. RESULTS AND DISCUSSION

The "VS Code-Voice Companion for Physically Disabled people using NLP" project demonstrates significant advancements in facilitating code input and execution through innovative integration of voice commands and NLP techniques. By harnessing the power of OpenAI's GPT models and Google Cloud Speech-to-Text API, the system empowers users to interact with the VS Code environment using natural language, streamlining the coding process and enhancing productivity. The development of a user-friendly web interface ensures intuitive access to the system's functionalities, while the execution module enables real-time execution of generated code snippets, providing immediate feedback to users. The project has successfully automated code generation and execution, but ongoing efforts are required to enhance the accuracy and robustness of NLP algorithms and support additional programming languages. Despite these challenges, the project represents a significant step forward in bridging the gap between natural language communication and programming, offering promising opportunities for further innovation and improvement in the future.

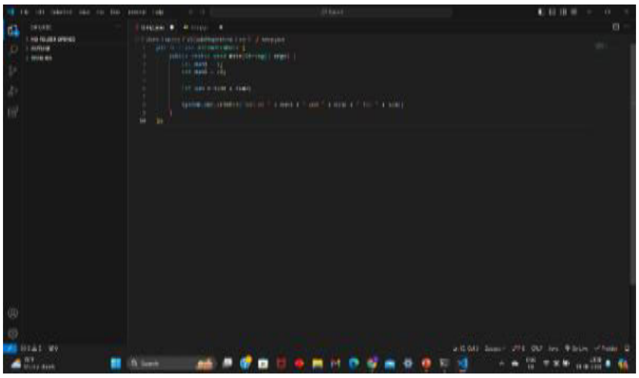


Fig – Output Screenshot

VI. CONCLUSION

The "VS Code-Voice Companion for Physically Disabled People Using NLP" used in simplifying code input and execution by integrating voice commands and NLP techniques. Leveraging OpenAI's GPT models and Google Cloud Speech-to-Text API, the system allows users to interact with the VS Code environment using natural language, thus streamlining the coding process and boosting productivity. A user-friendly web interface ensures easy availability for the system's functionalities, while the execution module provides real-time code execution and immediate feedback. Although the project has successfully automated code generation and execution, continuous efforts are necessary to enhance the accuracy and robustness of NLP algorithms and to support additional programming languages. Despite these challenges, the project marks a significant advancement in bridging the gap between natural language communication and programming, offering exciting opportunities for further innovation and improvement in the future.

REFERENCES

1. N. Khurpia, 2021 Conference "Analysis of Machine Code Using Natural Language Processing," on Intelligent Systems, Smart and Green Technologies (ICISSGT), Visakhapatnam, India, 2021, pp. 183-187, Doi: 10.1109/ICISSGT52025.2021.00046.
2. P. Mohana, M. Muthuvinayagam, P. Umasankar, and T. Muthumanickam, "Automation using Artificial Intelligence Based Natural Language Processing," 2022 6th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2022, pp. 1735-1739, doi: 10.1109/ICCMC53470.2022.9754028.
3. Indukuri Manikanta Sai Varma, Kalidindi Pranith Varma, "Voice Assistant Using Artificial Intelligence," link: https://www.ijedr.org/papers/IJEDR2202018.pdf.
4. S. Hossain, M. A. Emi, M. H. Mishu, R. Zannat, and Ohidujjaman, in 2021 "Code Generator Based on Voice Command for Multiple Programming Languages," on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 01-05, doi: 10.1109/ICCCNT51525.2021.9579880.
5. M. G. Prakash, A. Ponmalar, S. Deeba, A. Akilandeswari, S. B. M. Rasool, and P. Lavanya, "VSCODE - Code with Voice Using Natural Language Processing (NLP)," 2022 International Conference on Computer Power and Communications (ICCPC), Chennai, India, 2022, pp. 571-574, doi: 10.1109/ICCPC55978.2022.10072106
6. A. S. Spanias and F. H. Wu, "Speech Coding and Speech Recognition Technologies: A Review," 1991 IEEE International Symposium on Circuits and Systems (ISCAS), Singapore, 1991, pp. 572-577 vol.1, Doi: 10.1109/ISCAS.1991.176400.
7. M. S. Haleem, "Voice Controlled Automation System," 2008 Conference Karachi, Pakistan, pp. 508-512, Doi: 10.1109/INMIC.2008.4777791.
8. Daniel W, "A Survey of the Usages of Deep Learning for Natural Language Processing," link: https://arxiv.org/pdf/1807.10854.
9. E. d. S. Maldonado, E. Shihab, and N. Tsantalis, "Using NLP to Automatically Detect in IEEE Transactions on Software Engineering Self-Admitted Technical Debt", vol. 43, no. 11, pp. 1044-1062, 1 Nov. 2017, Doi: 10.1109/TSE.2017.2654244.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com