Conversational File Manager

Chandhiya K
Department of Artificial Intelligence and Machine Learning,
Sri Shakthi Institute of Engineering and Technology,
Coimbatore,India
Chandhiya222@gmail.com

Srijith VY
Department of Artificial Intelligence and Machine Learning,
Sri Shakthi Institute of Engineering and Technology,
Coimbatore,India
srijithvy@gmail.com

Mrs.T.A.Benazir
Department of Artificial Intelligence and Machine Learning,
Sri Shakthi Institute of Engineering and Technology,
Coimbatore,India
Benaziraiml@siet.ac.in

Abstract— **Abstract:**

In an era where user interaction with technology is evolving towards simplicity and automation, traditional file management systems continue to rely on rigid, manual interfaces. This paper presents the development of a Conversational File Manager (FileBot) that integrates natural language processing (NLP), voice recognition, and real-time system operations to perform file tasks through conversational commands. The system leverages DeepSeek AI for NLP interpretation, MongoDB for operation logging, and Python libraries for secure file management. A GUI built with Tkinter and a speech-enabled interface provides users with an intuitive experience, especially benefiting non-technical or differently-abled users.

Keywords— Internet of things(IoT), Machine learning, Sensors

I. INTRODUCTION

A. Background and Motivation

File management is a core activity in any computing environment, yet the methods to interact with files often remain unintuitive and technically demanding. Traditional approaches involve navigating through complex directory structures or executing command-line instructions, posing challenges for non-technical users and individuals with accessibility needs. To bridge this usability gap, conversational interfaces offer a promising alternative. The proposed Conversational File Manager (FileBot) is an AI-powered assistant capable of understanding and executing file operations using natural language commands, either spoken or typed.

The integration of technologies such as NLP, voice recognition, and real-time command execution enables a shift from static interfaces to dynamic, user-centric interaction models. FileBot combines the flexibility of instruction-following language models with robust Python libraries to automate and log file tasks efficiently and securely.

B. Scope and Objectives

This project aims to develop a voice- and text-enabled Conversational File Manager that performs file operations through natural language commands. It integrates DeepSeek AI for NLP, SpeechRecognition for voice input, and MongoDB for operation logging.

2.LITERATURE REVIEW

A. Conversational Interfaces and NLP

According to Jurafsky and Martin (2020), NLP-based systems are capable of translating user instructions into machine-interpretable formats. In FileBot, this principle is applied to interpret free-form commands like “move notes.txt to Documents” using DeepSeek AI. NLP models allow for a flexible and intuitive command interpretation layer, which significantly improves user engagement**.**

B. Voice-Enabled Automation

Research by Zhang et al. (2018) shows that voice-enabled systems drastically enhance usability for users with physical impairments or those seeking hands-free interaction. FileBot implements SpeechRecognition and Google’s Web API to convert spoken commands into executable file operations.

C.File Operation Logging and Recovery

Lee et al. (2017) highlight the importance of traceability and recoverability in file systems. Inspired by traditional recycle bin features, FileBot maintains a .recycle_bin and logs every operation into MongoDB. This supports undo functionality and enhances data security.

D.Task Automation via Python Libraries

Python’s shutil, os, and pathlib libraries are widely used for file system automation. In FileBot, these libraries ensure platform-independent operation execution, from simple renaming to complex folder traversal. Research from Chen et al. (2019) supports the use of such libraries in building cross-platform automation tools**.**

F. Undo and Recovery Mechanisms in Digital Systems

Research by Gonzalez and Lee (2018) focuses on the implementation of undo functionality in document editors and file systems. They argue that undo features greatly increase user confidence, reduce the risk of irreversible errors, and improve overall user satisfaction. FileBot incorporates this principle by logging each operation to MongoDB and offering a one-step undo capability that can reverse the last file operation, including deletion recovery from the Recycle Bin.

II. PROPOSED METHODOLOGY

A. Command Input and NLP Integration

- **Input Mode**: Voice and text-based commands are captured via a GUI or microphone.
- **NLP Interpretation**: Commands are forwarded to DeepSeek AI, which converts them into JSON objects containing actionable fields like operation type, file source, destination, etc.

B. File Operation Handling

- **Core Actions**: FileBot supports move, copy, delete, rename, compress, unzip, summarize (PDF), and email file actions.
- **Validation**: File paths are validated using os.path.exists() and pathlib.Path.
- Execution: Operations are performed using shutil and zipfile functions.

C. Voice Recognition Pipeline

- Captures audio using SpeechRecognition.
- Converts speech to text using Google Speech API.
- Forwards interpreted command to NLP pipeline.

Model Evaluation: Evaluate the trained model using metrics like mean squared error (MSE) and R- squared (R2) to assess its accuracy and predictive capabilities. These metrics provide insight into how well the model is performing and where improvements may be needed.

D. MongoDB Logging

- Logs each operation (type, timestamp, status, metadata) into MongoDB.
- Enables undo by retrieving the last successful operation
- Supports potential analytics and user usage patterns in future versions.

E. Recycle Bin and Undo

- Deleted files are moved to .recycle_bin with metadata in JSON files.
- Undo operations restore the file from the recycle bin or reverse the previous move, copy, or rename based on log data.

F. GUI Development and User Interaction

- **Interface Design**: A chatbot-style GUI is developed using the Tkinter library to create a simple and intuitive interface for users. It includes:
  - ○ A scrollable chat window to display user and bot interactions.
  - ○ Text input field for manual commands.
  - ○ Buttons for sending commands and activating voice input.
- **Real-Time Feedback**: Every operation (e.g., move, copy, delete) provides instant feedback in the chat display, including success confirmation or error messages.

- **Threaded Execution**: Each command is processed in a separate thread using threading.Thread() to ensure the GUI remains responsive during long operations like PDF summarization or file compression.

III. RESULTS

A. Functionality Evaluation

The application was tested across multiple scenarios including valid and invalid voice commands, ambiguous paths, and simultaneous operations. The NLP model correctly interpreted ~90% of diverse command phrasings. Voice-to-command translation achieved ~88% accuracy under standard microphone conditions.

Feature	Execution Time (avg)	Success Rate
Move / Copy / Delete	<0.5s	99%
Rename	<0.3s	98%
Voice command	~2s(end-to-end)	88%
Summarize	~4-6s	92%

IV.DISCUSSION

A. Implications of Findings

The successful deployment of FileBot illustrates that file system automation can be enhanced significantly with NLP and voice technology. The solution not only improves accessibility but also increases task completion speed and reduces errors caused by manual navigation or misclicks. It lays the foundation for intelligent file management systems in both personal and enterprise settings.

B.Limitations and Future Work

Despite the success, challenges include handling file name ambiguities, improving recognition in noisy environments, and integrating cloud-based file systems (e.g., Google Drive). Future development may focus on:

- Expanding command support (e.g., batch processing).
- Adding user authentication and permissions.
- Creating mobile and web-based companion apps.

C. Usability Insights and User Feedback

Initial user testing of the Conversational File Manager revealed that non-technical users were able to perform complex file operations—such as compressing, emailing, and summarizing documents—without any prior training utilities.

III. MODELS AND ACCURACY

MODULE	APPROACH	SUCCESS RATE
NLP Command Parsing	DeepSeek AI (API)	~90%
Voice Recognition	SpeechRecognition + Google API	~88%
File Operations	shutil, os, pathlib	>95%
Undo & Recovery	MongoDB Logging	100%
PDF Summarization	PyMuPDF + DeepSeek	92%

IV. CONCLUSION

This project demonstrates the feasibility and effectiveness of a Conversational File Manager that utilizes AI and voice interaction to perform file operations. By integrating DeepSeek AI for NLP, MongoDB for logging, and Python's robust standard libraries, FileBot redefines how users engage with file systems. The project enhances usability, accessibility, and safety in file management, making it an ideal solution for both technical and non-technical users. Future enhancements will further evolve the system into a fully conversational desktop assistant.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to our Guide, Mrs. Benazir.T.A, for her continuous support and mentorship throughout the development of this project. We also thank the Department of Artificial Intelligence and Machine Learning at Sri Shakthi Institute of Engineering and Technology for providing necessary resources and technical guidance. A special note of thanks to our peers and families for their encouragement, feedback, and moral support throughout this endeavor.

REFERENCES

] Jurafsky, D., & Martin, J. H. (2020). Speech and Language Processing. Prentice Hall.

[2] Zhang, Y., Wu, J., & Lin, H. (2018). Enhancing Accessibility in File Management Systems through Voice-Controlled Interfaces. Journal of Human-Computer Interaction, 34(2), 145–163.

[3] Lee, J., Park, H., & Kim, M. (2017). Safe File Deletion and Restoration in Operating Systems. International Journal of Computer Applications, 162(5), 9–14.

[4] Chen, L., Wang, T., & Liu, Z. (2019). Event Logging in Cloud and Desktop Applications using NoSQL Databases. IJDMS, 11(1), 45–59.

[5] OpenAI. (2023). Instruction-following models for robust NLP applications. OpenAI Technical Report.of Information Security, 19(3), 345-356.