MEDIA PLAYER USING VOICE RECOGNITION

R Jaswanth Aakash, C Jeeth Anghesh Baalaji

1 UG Students, Department of CSE, Sri Krishna College of Technology, Coimbatore, Tamil Nadu, India

Abstract—Numerous Media Player software are available in the market, only a few are user friendly. However, they do not have any voice recognition feature incorporated in them, which forms a very user-friendly type of interface. The visually impaired also find it difficult to interface with the software. Thus, we have come up with an idea of designing a media player named “Pepper Media Player” capable of playing all type of audio and video files such as MP3, MP4, WAV, etc. The main feature of the software is to incorporate voice recognition with it. The wake word “HELLO” is used to start the communications with the software. The users can use voice commands to communicate with the software. This helps the users to control the software without the use of mouse, keyboard or any other input devices. The media player is designed as a standalone application to run in any general-purpose computers designed using ‘Java’ programming language, which is a platform-independent language and can be used in any operating system.

Keywords—Java, Media player.

I. INTRODUCTION

SOFTWARE DEVELOPMENT

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Software development is a process of writing and maintaining the source code, but in a broader sense, it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

Software can be developed for a variety of purposes, the three most common being to meet specific needs of a specific client/business, to meet a perceived need of some set of potential users (the case with commercial and open source software), or for personal use (e.g. a scientist may write software to automate a mundane task)

NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) is the ability of a computer program to understand human language as it is spoken. NLP is a component of artificial intelligence (AI). The development of NLP applications is challenging because computers traditionally require humans to "speak" to them in a programming language that is precise, unambiguous and highly structured, or through a limited number of clearly enunciated voice commands. Human speech, however, is not always precise -- it is often ambiguous and the linguistic structure can depend on many complex variables, including slang, regional dialects and social context.

SPEECH RECOGNITION

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. Rudimentary speech recognition software has a limited vocabulary of words and phrases, and it may only identify these if they are spoken very clearly. More sophisticated software has the ability to accept natural
speech. Speech recognition works using algorithms through acoustic and language modelling. Acoustic modelling represents the relationship between linguistic units of speech and audio signals; language modelling matches sounds with word sequences to help distinguish between words that sound similar. Often, hidden Markov models are used as well to recognize temporal patterns in speech to improve accuracy within the system.

UNDERSTANDING NATURAL LANGUAGE

In modelling language understanding, we must deal in an integrated way with all of the aspects of language—syntax, semantics, and inference. The system contains a parser, a recognition grammar of English, programs for semantic analysis, and a general problem-solving system. We assume that a computer cannot deal reasonably with language unless it can understand the subject it is discussing. Therefore, the program is given a detailed model of a particular domain. In addition, the system has a simple model of its own mentality. It can remember and discuss its plans and actions as well as carrying them out.

II. SOFTWARES AND HARDWARES

JAVA

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere", meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture.

NETBEANS

NetBeans IDE is the official IDE for Java 8. With its editors, code analyzers, and converters, you can quickly and smoothly upgrade your applications to use new Java 8 language constructs, such as lambdas, functional operations, and method references. Batch analyzers and converters are provided to search through multiple applications at the same time, matching patterns for conversion to new Java 8 language constructs.

MAVEN FRAMEWORK

Maven is really just a core framework for a collection of Maven Plugins. In other words, plugins are where much of the real action is performed, plugins are used to: create jar files, create war files, compile code, unit test code, create project documentation, and on and on. Almost any action that you can think of performing on a project is implemented as a Maven plugin. Plugins are the central feature of Maven that allow for the reuse of common build logic across multiple projects.

HARDWARE COMPONENTS

- General purpose Computers
- Microphone
- Speakers

III. WORKING

It listens voice input form the user using the microphone and acts passively for the wake word “HELLO”. Once the software receives the wake-word, it starts listening for the commands for the next 5 seconds. It uses IBM Watson API for passive listening of the wake-word. For listening the commands, CMU Sphinx4 application is used. The program receives and reacts to the commands by performing the corresponding actions. For the Graphical User Interface, FXML and CSS are used. In the backend, java uses file storage systems to store the playlists information and location of the files.
IV. CONCLUSION

Media player which is more reliable, robust, and user-friendly by using voice recognition to provide commands like pause, play, next and previous, thus making the usage of the product very easy. With the help of IBM Watson Speech to Text API and CMU Sphinx, the recognition of wake-word “HELLO” and passing of commands have been made more accurate and reliable.

The construction of our own language model and grammar has made the recognition of keywords and eliminating all other unnecessary words. The project has been developed with the view of making it an open source project, so that it is available for all users around the world.

REFERENCES


image sharing scheme with encrypted shadow images using optimal homomorphic encryption technique. Journal of Ambient Intelligence and Humanized Computing, 1, 13.

https://doi.org/10.1007/s12652-018-1161-0


[50] Nithya, S., Sundara Vadivel, P., Yuvaraj, D., Sivaram, M. “Intelligent based IoT smart city on traffic control system using raspberry Pi and robust


[68]Sivaram, M. "Odd And Even Point Crossover Based Tabu Ga For Data Fusion In Information Retrieval." (2014)


[74]Obulatha-II-ME-CSE, Miss O. "Position Privacy Using LocX."


