

Initial Plan -

Majel: Voice Assistant Style Interface For Command Line Terminal

Project Description

Majel is a program that allows users to speak command line inputs aloud into a computer connected microphone and have them interpreted and executed by the system. It will support the input of preexisting commands, as well as commands involving user defined filenames and scripts. It will also feature a form of 'natural language' interface, where non explicit inputs that do not relate exactly to commands in the system will be able to be interpreted and executed.

It will be written in Python, using the SpeechRecognition and PyAudio modules as well as the CMUSphinx speech recognition system in the form of PocketSphinx. It will support the BASH shell (and BASH like shells such as fish), and will primarily be designed to work on Linux systems.

The overall goal of the project is to increase the ease at which complex and verbose command line inputs can be made. This will be useful for new users, who might find use of command line daunting or difficult at first and also for experienced users who might need to regularly run many complex commands that can be more concisely expressed as speech. If a 'natural language' input is given, the actual command that has been interpreted will be displayed; this will help inexperienced users learn commands, and clarify to experienced users what command will be executed.

Project Aims and Objectives

- Allow for the successful execution of any program installed on the system, given that it's name, flags and user defined inputs are said correctly.
- Majel will be able to be launched via a 'wake word' at any time.
- Majel will work entirely offline, with no calls to online APIs
- Some form of natural language interface will be implemented, with the user not having to explicitly state the name of the program, only what they want to have happen.
- Processing of input will be very responsive, with the command being executed as quickly as possible.
- The input will be checked and the user prompted if the input command could have been misinterpreted, especially when moving, copying and deleting files, or other critical commands.
- Majel itself will have a command line interface, and will be usable in BASH scripts.
- Majel will have an optional GUI interface, indicating its current status
- User will have the option of extending the set of commands Majel will recognise by importing their command line history, can be done automatically on a schedule
- Majel will be lightweight, requiring only the installation of a small number of Python packages to function on a standard Ubuntu based distribution

Work Plan

Week 1 (27/01/2020)

- Complete Initial Plan
- Continue work on prototype - no natural language at this point
- Research existing Voice Control systems
- Research building of language model
- Research grammar formats

Week 2 (03/02/2020)

- Submit Initial Plan (03/02/2020)
- Obtain data to build language model
- Research problem of user defined inputs (file, directory names)
- Research natural language interface
- Create test input data files
- Complete and test prototype
- Meet with supervisor - discuss prototype

Week 3 (10/02/2020)

- Note any issues that arise from testing
- Fix issues from testing
- Implement checking and user confirmation dialog for potentially dangerous commands
- Start on NL implementation
- Create test input data for NL
- Meet with supervisor to discuss NL implementation

Week 4 (17/02/2020)

- Finish up basic implementation
- Research implementation of GUI
- Meet with supervisor

Week 5 (24/02/2020)

- Continue work on NL
- Start GUI implementation
- Research implementation of 'wake word' feature

Week 6 (02/03/2020)

- Continue work on NL, GUI implementation
- Finish GUI and test
- Meet with supervisor

Week 7 (09/03/2020)

- Complete NL implementation
- Start on testing of NL

Week 8 (16/03/2020)

- Implement feature of importing custom command importing
- Fix issues raised in NL testing

Week 9 (23/03/2020)

- Finish up work on implementation

Week 10 (30/03/2020)

- Final testing

Week 11 (6/04/2020)

- Start work on final report
- Meet supervisor to discuss final report

Week 12 (13/04/2020)

- Continue work on final report

Week 13 (20/04/2020)

- Continue work on final report

Week 14 (27/04/2020)

- Finish Final report

Week 15(04/05/2020)

- Submit final report (07/05/2020)

*Author: George Close 1736823 closeg@cardiff.ac.uk
Supervisor: Professor David Marshall marshallad@cardiff.ac.uk
Module: CM3203 One Semester Individual Project (40 credits)*