

Final Project Spec

Jake Seaton - jseaton@college.harvard.edu, jakeseaton13@gmail.com

Harnek Gulati - harnekgulati@college.harvard.edu, gulatiharnek@gmail.com

Overview

We want to design an algorithm that will parse user input, as part of a program that translates it into commands that are executable by the computer. We want to make it simpler to access documents and programs that you use frequently, without having to navigate directories or go through multiple hoops for relatively simple actions. The idea springs from potential improvements over existing programs such as spotlight and alfred that make the system more user friendly. The computationally significant portion of the project will be the natural language processing algorithm, the significance of which we will attempt to translate into increased user friendliness.

Feature List

Core Features

- Natural Language Processor
- Search the computer
- Calendar additions/subtractions
- Opening/Closing applications
- Searching the web
- Print
- Start/Restart/Sleep
- Open and print
- Move directories

Cool Extensions

- Convert file types
- Opens Netflix and goes to the next episode
- Study Time
- Facebook commands
- Delete things
- Weather
- Stock Lookup
- Math

User Interface

Technical Specification

The output of the parser will determine the input for the subsequent portions so we should figure those out first. Then we can break implementation of the natural language parser with the consolidation of various system commands.

Natural Language Processing

- **Parse:** Changes text into code, giving us what is necessary with the function.
- **Search File:** This runs when the parser realizes that something is being typed in that requires a file. It then uses the in-computer functionality to search through it all

Events(The following will be a module type that the rest of the functions extend from):

- **Error/Need more information:** This will run if something goes wrong and the parser needs more data before it can reliably place something into events.
- **Done:** Runs the appropriate output so that the reader knows the command is completed.

Search the Computer

- **Open file:** Opens the read file.
- **Delete file:** Deletes the read file.
- **Print file:** Prints the read file.
- **Print _ copies of:** Prints x amount of copies of a file to default printer.

Calendar Additions

- **Add:** Add calendar to the event.
- **Change:** Be able to recognize ("change tomorrow's dinner to 5 o'clock") and react accordingly.
- **Done:** Outputs with confirmation with right information
- **Warning:** Prompts for more information.

Opening/Closing applications

- **Open:** Opens an application
- **Close:** Ends a process of an application.
- **Error:** Unable to open/process application
- **Done:** Outputs with confirmation with right information

Searching the web

- **Done:** Search using the specified search engine for the input

Print

- **Error:** The document does not exist or cannot be printed. (search the web?)
- **Done:** Takes a document, opens it in the correct, program and calls the print function.

Application

- Open: Opens the application
- Close: Closes the application (perhaps just command q?)

Start/Restart/Sleep

- Shutdown: Powers off the computer
- Restart: Restarts the computer
- Lock: Locks the computer

Move directories

- Move: Takes a document or directory to be moved and a new location, and moves it. Move _ to _

Next Steps

- Jake needs to learn python
- Windows or Mac or Linux or what (Bootcamp?)
- Algorithmic analysis
- Data structures? Tries, Hash tables.
- Figure out the limitations of interacting with the hard drive and other programs (ie the print and move functions)

Useful Documentation:

https://developers.google.com/google-apps/calendar/v2/developers_guide_python

<https://pythonhosted.org/plex/>

<http://stackoverflow.com/questions/7821661/how-to-code-autocompletion-in-python>

<http://www.learnpython.org/>

<http://isites.harvard.edu/icb/icb.do?keyword=k100309> CS181!