Eliza:

A virtual assistant for the Ubuntu platform

# Approach

The goal is to build this piece by piece, biting off what we can chew at every step

# Project tools

* ODK Server- <http://cs2-216003532-karuhanga-odk.appspot.com/>
* Github Repository- <https://github.com/Karuhanga/eliza>

# External Tools

* Kaldi- speech recognition
* PocketSphinx- speech recognition
* pydub- audio processing
* pyaudio- audio device access
* sklearn- a bit of machine learning
* Mozilla Deep Speech

## Roadmap

### Phase 1

* Setup ODK collection server and endpoints
* Figure out a way of finding out which program is currently running
* Evaluation of speech recognition alternatives (will make use of Kaldi for the more mature parts and PocketSphinx and sklearn for quick demos)
* Setup Kaldi
* Figure out a way of generating language phonemes that represent our accent
* Demo keyboard control and automation
* Solve compatibility problems between the keyboard and pyaudio tools
* Demo sound based routine activation (this will be done by sound levels e.g clapping at this point) on select applications
* Collect keyword data based on single user (Using odk)
* Perform data cleaning and normalisation
  + Normalise amplitude(related to volume)
  + Remove leading and trailing silences
  + Normalise length
* Build speech recognition model for hotwords based on one user
* Demo hotword recognition, based on single user
* Fully automate keyboard input to allow us use hotkeys and shortcuts
* Integrate hotkey activation to hotword triggers
* Complete v1 build with three application hotword controls

### Phase 2

* Expand range of words that can be recognised by model
* Remove single user dependency- start collecting hotword data from multiple individuals
* Improve data collection methods- make these more professional and increase scale
* Improve data cleaning methodology
* Train model on new acquired data set
* Expand targeted applications
* Evaluate possibility of shifting from a control based system to a more intelligent assistant
* Passive search and trigger readiness- look into possibility of expanding scope to continuously scan the pc and build a structure that allows efficient search based on voice input/ grep it
* Customise CMU Sphinx- <https://www.quora.com/How-can-I-get-started-with-the-CMUSphinx-setup-for-building-a-new-languages-speech-recognition>

### Todo

* Scan installed programs on install and create terminal launcher shortcuts for each
* Question to action heuristic
* Add google like quick actions e.g floating lyrics
* Add quick utilities, like time
* Figure out how to open programs in non root
* Resolve action based on spacy speech similarity
* Add reminders
* Add persistence for things like reminders and todos
* Play music
* Build electron based app to help with custom actions
* Data collection with <https://voice.mozilla.org>

# References

## Kaldi Paper and Documentation

* <http://kaldi-asr.org/doc/about.html>
* <http://publications.idiap.ch/downloads/papers/2012/Povey_ASRU2011_2011.pdf>

## CMUSphinx

* <https://cmusphinx.github.io/wiki/>
* <https://cmusphinx.github.io/doc/pocketsphinx/>

Interesting projects that might be of help

* <https://spacy.io/>