Overview and Motivation

From Amazon's Alexa to MicroSoft's Cortana, these AI assistants rely on the ability to translate natural language instructions to machine commands, e.g launch itunes, search for the weather. The performance of these commercially available AI assistants are great, but not satisfactory enough. Although this project will not aim at surpassing Alexa or Cortana, it strives to come up with an AI program that can assist users of Unix systems by *translating English instructions to Unix commands*. It will especially benefit users who are unfamiliar with or unaware of the command line utilities and would like to exploit such utilities. It will be a good aid to developers working in Unix environment, or simply serve as a shortcut to regular users.

Input & Output

In the scope of this project, our ***input*** will mainly be human instructions in English describing tasks involving: finding and processing files, setting timer on system (delayed action), shutting down, downloading and uncompressing things from the web, and launching certain programs. The ***input*** will come in text form from the terminal, speech recognition is not included in this project as it has already been well-developed. The ***output*** will be the corresponding Unix commands that could perform the tasks in the instruction in the correct order.

One simple example of the input would be: "can you shut down in 10 minutes?"
The corresponding output would be "`shutdown -h +15`".

One complex input would be: "Download this random_file.tar.gz from www.cs221.edu/resource, and can you wait for the download to finish and then just uncompress it? And, shut down afterwards."
The corresponding output would be "`wget http://www.cs221.edu/resource/random_file.tar.gz && tar xvzf random_file.tar.gz && shutdown -h`"

Training Data

To get a decent amount of training data, I will first create several simple inputs that should be translated to one command only (their outputs). I will then use Amazon Mechanical Turks or Upwork to crowdsource people to help me create equivalent expressions of those simple English inputs/instructions, e.g "shut down in 10 minutes" == "turn off in 10 minutes".

To form complex inputs/instructions, I will explore various structures of logically joining simple instructions in English, maybe also with the help of crowdsourcing platforms like Amazon Turks. Then, I can extrapolate these structures to sets of multiple simple instructions and generate many sensible complex instructions and their corresponding output.

Evaluation Metric

Since the output will contain a sequence of Unix commands, I can measure the performance of my AI program by the following 2 metrics.

1	On average, the overlapping componential commands of my output with the oracle's output **divided by** the number of commands in the oracle's output. E.g: when mine is (A, B, C) and oracle gives (A, B, D), accuracy is ⅔ = 66.7%	Messures translation accuracy
3	The percentage of output sequences that has the correct ordering. For example, if the command sequence of the oracle is (A, B, C, D), our output (A, B, F, D) will be considered as having the right ordering.	Measures logic accuracy

The project will be considered successful if both metrics measure > 85% on test set.

Baseline and Oracles:

Baseline: We can only translate a very small set of simple instructions to Unix commands, such as the ones supported by Betty[1].

Oracle: The sequence of Unix commands I, a computer science student, would write out for English instructions of any complexity.

The Gap & The Toolkit

Machine translation between human languages has been greatly improved with the advancement of machine learning, e.g. RNN, RNN Encoder-Decoder[2], LSTM[3]. It would be great to be able to translate human languages to machine commands to provide a more layman approach to interfacing with computers. The main challenges in this project are learning/extracting the same tasks behind different phrasing of an instruction, and identify the right order of tasks encapsulated in the instruction's logic. In this project, I would like to utilize the tool of LSTM as it has seen some very promising results[3], and it is nicely integrated to TensorFlow.

References:
[1] SK, *"Betty: Translate English Phrases Into Linux Commands"*, Unixmen, https://www.unixmen.com/betty-translate-english-phrases-linux-commands/
[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, Yoshua Bengio "*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*", Cornell University
https://arxiv.org/abs/1406.1078
[3] Colah, "*Understanding LSTM Networks"*
http://colah.github.io/posts/2015-08-Understanding-LSTMs/