Report

# Team Members: Declan, Sid, Saad

# Dataset:

We decided that formal language would be best for helping to initially predict what the astronauts would say. Although our current predictions utilize only English data, we plan to improve in the future by incorporating data from multiple languages. We had two sources of data in our project. We used one to train and may or may not use the other to train as well in the future.

* **Wikipedia Dataset**: Although we didn’t end up using this dataset for our initial training set, it appears that it could be useful in the future for training due to the formal nature of the sentences. It contains 7.8 million sentences from Wikipedia articles. (The whole dataset is too large of a file to upload to GitHub). We did create a script to randomly select 100k of the 7.8 million sentences and have included “smallerData.txt” as an example run of our script, to get a dataset that would fit within the 10MB max file size.
  + <https://www.kaggle.com/mikeortman/wikipedia-sentences>
* **Metamorphosis Dataset**: This is simply a cleaned version of the famous book Metamorphosis by Franz Kafka. We chose it due to the fact that the book is in the public domain, contains a lot of data and is fairly formally focused. One possible issue is that the book was written in 1912, so the language may be a bit outdated.
  + <https://github.com/Bharath-K3/Next-Word-Prediction-with-NLP-and-Deep-Learning/blob/master/metamorphosis_clean.txt>

# Method:

We have designed a three-layer neural network.

* **Input layer:** The input layer will accept the data (word) where each character is represented by one hot vector.
* **Hidden layer:** we will compute one hidden state using the tanh activation function represent by the following formula:



* **Output layer:** For the output layer we will use a SoftMax layer to predict the probability of the next character. We can represent it using the following equation:



We will be using Stochastic Gradient Descent (SGD) where each batch consists of only one example. In other words, the RNN model will learn from each example (name) separately, i.e., run both forward and backward passes on each example and update parameters accordingly. Created with assistance from [this project](https://github.com/geekquad/TEXTGEN/blob/master/char%20rnn.ipynb?fbclid=IwAR0i5O7OZr5HEY9hCUYnSRGaLsjVVUZQjVQUJrgdW5faNtF4uxREYXL5aA8).