Final Project Writeup

For this project, we decided to use a GRU model due to their straightforward implementation and their quick training speed. One of the main challenges of creating a model of this size is the time required to train it. Even using the cloud computing resources it took us several hours to train for a few hundred epochs. Because of this we wanted to make sure we got as much training done in the time provided as possible and so we chose to use a GRU implementation for or model. The accuracy of GRU models is generally similar to the performance of LSTM models which was the type that seemed most obvious initially. However, GRU models are computationally more efficient that LSTM models and thus we could get more training done in the same period of time.

We constructed our training dataset by downloading Wikipedia articles in 15 different languages. We chose languages based on number of speakers in the present day, and also on which countries currently have space programs or astronauts on the International Space Station. Our set of languages was English, Mandarin, Cantonese, Italian, German, Japanese, Russian, Spanish, French, Norwegian, Dutch, Danish, Swedish, Hindi, and Arabic. This represents a very diverse coalition of countries and speakers of these languages account for the majority of the human population.

Wikipedia is the source of our data. We retrieved articles by translating topics related to space, astronomy, and science using the google_trans_new python package and downloaded the article in that language with the wikipedia python package. Our dataset consisted of a maximum of 1,000,000 characters per language, though some (Chinese, for example) were under that. While some languages were under-represented, overall most of the languages had a similar presence in the dataset.

Our model itself was constructed using the GRU object from PyTorch's nn library. Ours uses two layers, 300-dimensional embedding vectors, and its hidden state is of size 2048. We initialized our embeddings from the GLoVe 300 embeddings vectors, which gave our model a boost for working with English characters and punctuation. We separated our dataset into lines, then separated those lines into 400 character chunks (or less, if the line was too short), then randomized the order of these chunks. There were 107569 chunks in our dataset. Each epoch consisted of choosing 128 chunks from the dataset and training the model on each character in each chunk. None of the lines in the dataset was trained on more than once. We trained the model for 60 epochs. We would have liked to train for more, but we ran into a problem late on Friday and had to re-train our model from scratch in a couple hours.