CSE 447 Final Project Report

For our course project we opted to train separate character level autoregressive transformer models, each consisting of 4 residual multihead attention layers, for each of our supported languages. For each supported language, a model was trained auto-regressively on a corpus of text in said language. Character embeddings for each language's corresponding character set were learned during training by way of propagating gradients into the embedding layer and initializing character embeddings with random normal noise. All characters encountered during training which were not in the predefined character sets for each language were mapped to a special "<UNK>" character. To enable autoregressive training, attention weights were masked with a lower triangular context_window by context_window square matrix, ensuring that a prediction for character x_i depends only on x_j given j <= i for input sequence x (those characters preceding the ith character in the input sequence). Input sequences were appropriately shifted to obtain target sequences when computing the loss function. The mean of cross entropy loss between the model's predicted distribution over the relevant character set and the next character over all character positions in the input sequence served as the loss function. Though experimentation was limited, we found using an embedding dimension of 192, 4 attention heads per attention layer and a context window of length 64 yielded reasonable results and acceptable training and inference times given our resource constraints. Models were evaluated by way of computing top-3 accuracy on relevant held out test sets.

At inference time, each input sequence was left padded with relevant filler text if the input sequence was below the context window size (64) and, otherwise, the last context window size characters of the input sequence were used as input. At inference time, the top-k (k=3) predicted logits were returned from the model's prediction (excluding the logit corresponding to "<UNK>") for the last character in the input sequence. To select which model to use for character prediction, the lang_id python module was utilized to identify the most probable supported language given the provided input sequence and the model corresponding to said most probable language was subsequently selected.

The data we collected was from DataBus DBpedia. This database consisted of Wikipedia articles translated in over 140 languages. This data is updated every month by DataBus. We downloaded large collections of "short abstracts"' of Wikipedia articles, (each abstract was about a paragraph of text). In total we downloaded seven of these large collections, one for each of the languages we decided to train on (English, Spanish, French, Chinese, Japanese, Hindi, Russian, and Norwegian). While these abstracts were from random Wikipedia articles, and so had some pretty complicated/technical wording in them, the trained model ended up performing quite nicely.