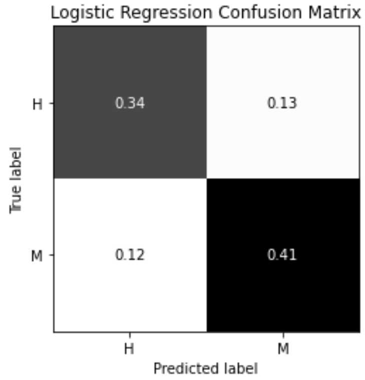
REU Application Task: Sam Fuchs

Baseline Modeling



To set a baseline, I train a single-variable logistic regression to predict the class of the translation based only on the Bleu translation score. This predictor's weights indicate that we expect human translations to have a significantly higher Bleu score than a machine translation. The average F1 score for this model on the test set is 0.75, a significant improvement over the baseline model, and a performance that is corroborated by its confusion matrix, which shows strong consistency in classification. Moving forward, I compare my model to this baseline logistic regression.

Preprocessing

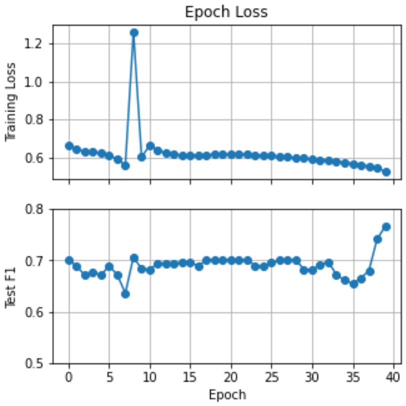
To prepare the text for classification, we use GloVe word embedding vectors with length 100, pretrained on the Wiki-Gigaword dataset. In my architecture, I aim to classify translations by comparing the candidate translation to the reference translation, disregarding the source text.

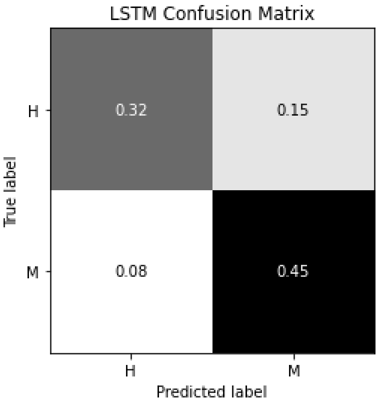
To this end, I construct input tensors by appending the corresponding (by position) word embeddings from the candidate and the reference translations together to form a dual embedding vector. To these embedding vectors, I add the quality score of the translation, which is constant across the sequence length. In cases where the candidate and reference translations vary in length, the shorter sequence is padded with a zero-vector embedding. Additionally, when we preprocess words in the test set that are unknown in the training set, they are assigned an embedding vector of random values.

Model Architecture

The core of the deep neural model that I design for this task is an LSTM with an input size of 201 and a hidden size of 500. This LSTM takes in each feature vector (of reference word embedding, candidate word embedding, and translation quality) and produces an output that is passed back to itself along with the next feature vector. We take the final output of this LSTM and pass it through a linear layer to generate two outputs: a score for classifying the sample as "human", and a score for classifying the sample as "machine". We take the argmax of these scores to find the model's predicted class.

Results





The neural model underperforms my expectations, scoring only slightly better than the logistic regression, with an F1 score of .766. It tends to classify many human translations as machine translations, while nearly all machine translations are classified correctly. I suspect that this is a result of limited training data. With less than 600 samples, the frequency of each word in the dataset is very low, and the model struggles to learn meaningful insights about particular embeddings as a result. The model might also be helped by more feature engineering, such as computing the cosine similarity of the input sequences in the preprocessing step. The LSTM architecture may struggle to capture similarities between words that are in different positions in the input sequences, and a minor offset at the beginning of the sequence could make it much more difficult for the model to classify the input. With more iteration, I expect that an architecture similar to this one would score higher on this problem.