Predicting hotel score by review using LSTM and Transformer

Setup

The data might be found here. The data consists of 100k reviews and scores collected from 1500 hotels. Each review includes two texts: positive and negative aspects. The column score may have a real value from 0 to 10. In the presented data, the lowest score is 2.5.

Goal: get the lowest possible MAE for score using only review texts.

LSTM

First, I use a bi-directional LSTM model with 2 layers, and an additional Dropout layer setting 70% of the output to zero to prevent overfitting with L1 Loss from PyTorch. I did not find any gain in performance from separately learning embedding layers for "negative" and "positive" columns. Thus, I combine them into one column.

Secondly, I use a bi-directional 2-layer LSTM model with an Attention module and Dropout layer, setting 85% of the neuron output to zero. Again, feeding columns "negative" and "positive" separately did not improve the performance but increased the time.

I trained both models for 30 epochs. The result for a regular LSTM is below:

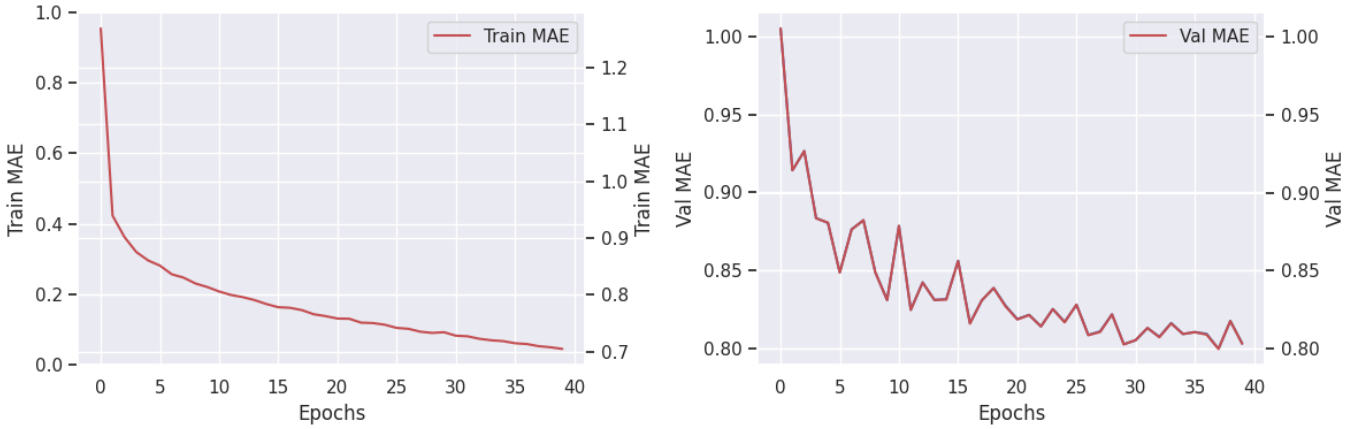


Figure 1: LSTM

Clearly, the model overfits slightly; the MAE for the validation set reaches 0.800, which is 8% error. The model reaches its peak performance after 30 epochs. Next, I present the results for LSTM with an Attention layer:

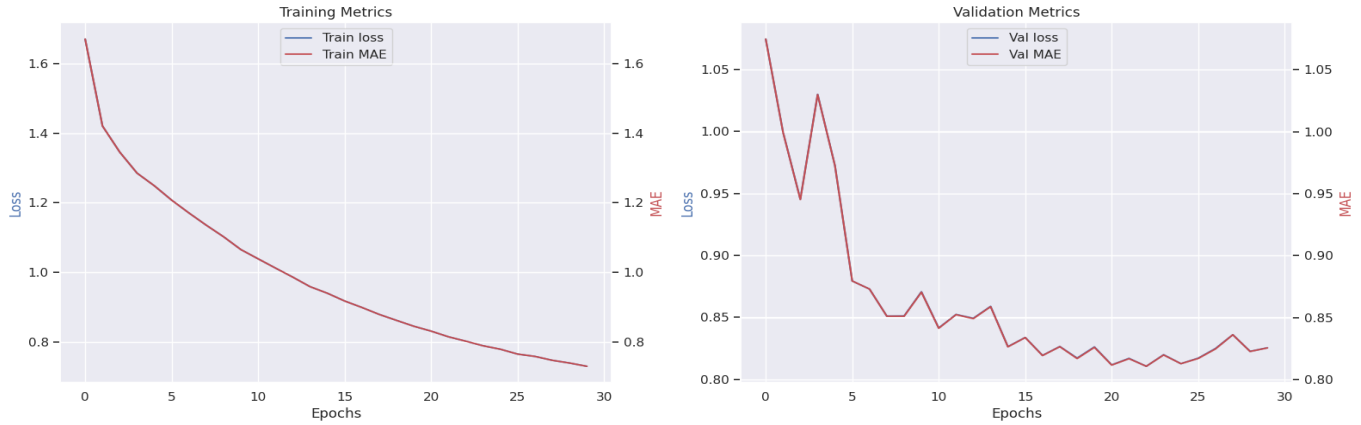
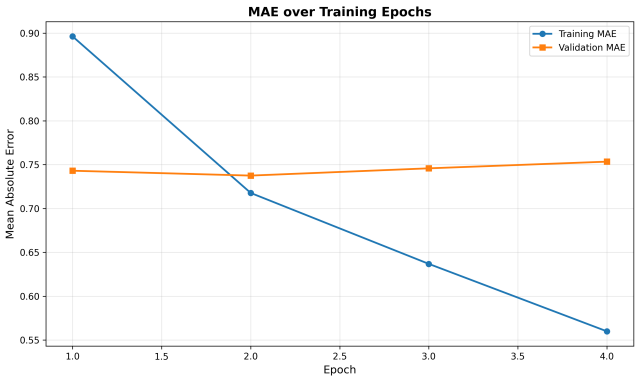


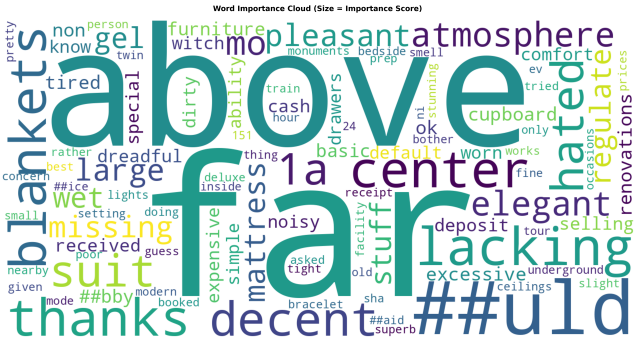
Figure 2: LSTM with Attention

I increased the Dropout layer rate compared to the regular LSTM due to overfitting. LSTM with an Attention layer overfits faster than a regular LSTM, but reaches its peak performance after approximately 20 epochs. The lowest obtained MAE is 0.813, which is 8.1% error for the 1 to 10 scale. Thus, LSTM with an Attention layer takes less time to train, but the overall performance might be worse compared to a regular LSTM.

Transformer

I used a fine-tuned DistilBERT transformer from HuggingFace with PyTorch Lightning module for a scalable training infrastructure. For stability and efficiency, I used cosine learning rate scheduling with 0.15 warmup proportion, gradient clipping, and early stopping. I trained the transformer for 5 epochs. The obtained results are below, as well as the words' importance analysis:





The results demonstrate that the Transformer overfits after two epochs, while MAE for the validation set stays roughly on the same level. The lowest obtained MAE was 0.7375, which is 7.37% for the 1 to 10 scale score. As expected, the transformer demonstrates the best performance over the 3 considered models. As for words' importance, some words are quite meaningless (1a, ##uld), while others are expected for the hotel review.