The Effect of Lexical Class on Sentiment Analysis

Filip Aleksic, Felicity Hade, Justin Long, and Evan Moore
CSCI 1470, Fall 2022

Introduction

Our group set out to construct a sentiment classification algorithm in order to predict whether a certain comment/review is positive or negative. In order for our model to be generalizable, we opted to use data from Twitter, since we saw it as one of the best sources to collect text data that has a wide range of context. We were also interested in seeing how different parts of speech (such as nouns, verbs, adjectives, and adverbs) affect the accuracy of our sentiment classifier by omitting them from the data that we trained our model on.

Data Collection & Preprocessing

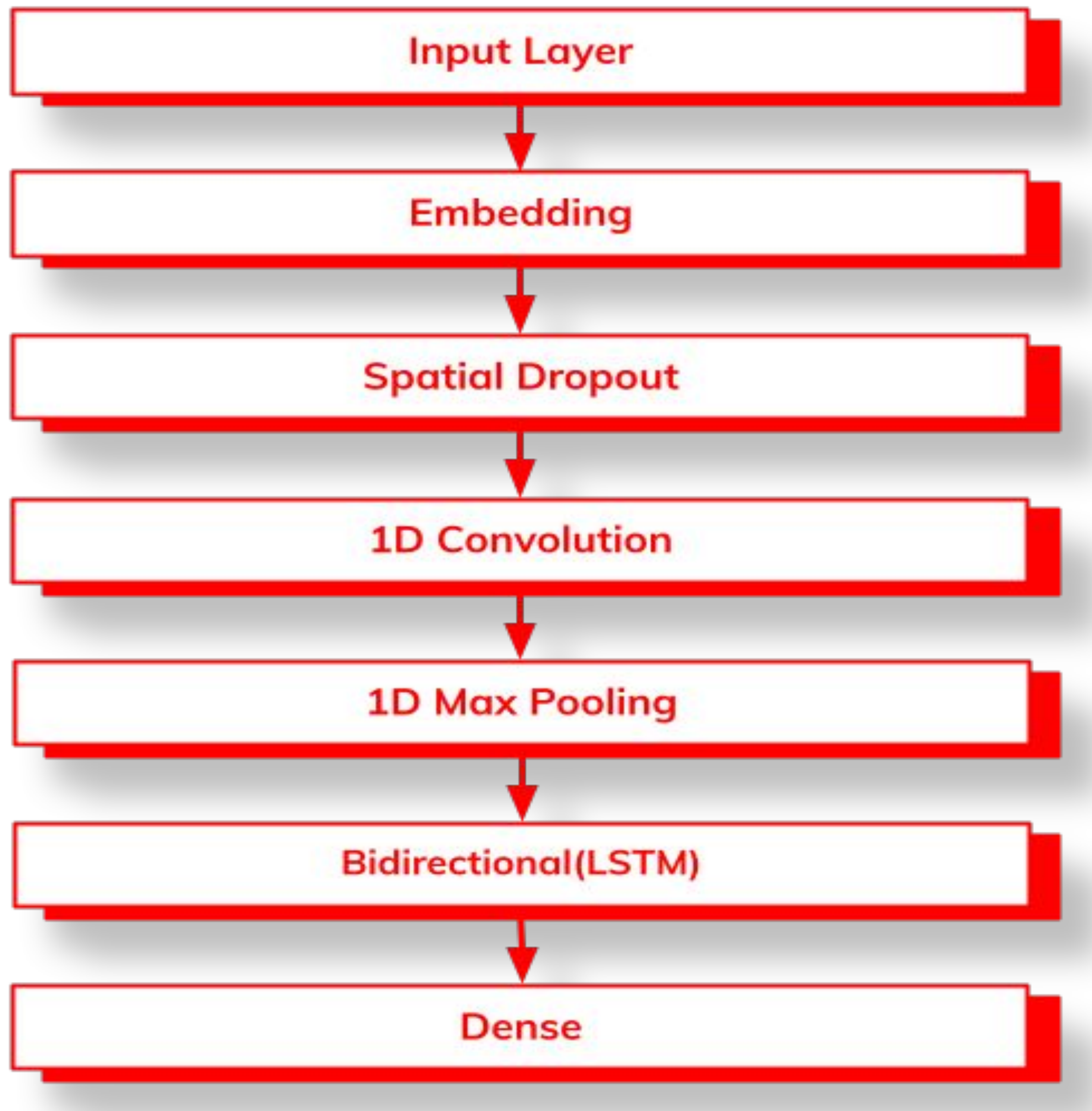
The dataset contained the tweet, the author, the date, the tweet ID, and the sentiment of the tweet. We decided to drop the author, date, and tweet ID columns because we did not want these to affect our analysis and wanted to focus solely on the text and its sentiment. There were 1.6 million tweets in the dataset, but we opted for using a random sample of 500,000 tweets. We wanted to improve the efficiency of preprocessing, training, and testing while retaining a large enough sample to maintain accuracy. We then split the sentences into words, removed all the stop words, lemmatized the text, filtered out characters that were not alphanumeric, and made all words lowercase. We then performed a random 70/30 train/test split on the data. We then attached labels to each word with their corresponding parts of speech in the training data. Once the labels were attached, we were able to go through the data and remove all words belonging to a specified lexical class. After removing the words, the data was restored to the original format (an array of tweets). We then fit the tokenizer on the text, transformed the text to sequences, and finally padded the sequences.

Challenges

We initially wanted to extract data from the Twitter API, but the data would have been unlabeled; thus, we would not have been able to train our sentiment classifier on it. Preprocessing the data was challenging because there were a lot of parts to it, and we have not had exposure to this step previously.

Methodology

In accordance with the model described in the paper, we used a recurrent neural network (RNN) to classify the Twitter data. Because of their ability to represent temporal sequences, RNNs are well-suited to sequential problems like NLPs. We adjusted the parameters of each layer based on the respective input and output sizes provided in the paper, including an embedding size of 128. We found that adding a batch normalization step had little impact on model performance, so we ultimately omitted it.



Input Layer

Embedding

Spatial Dropout

1D Convolution

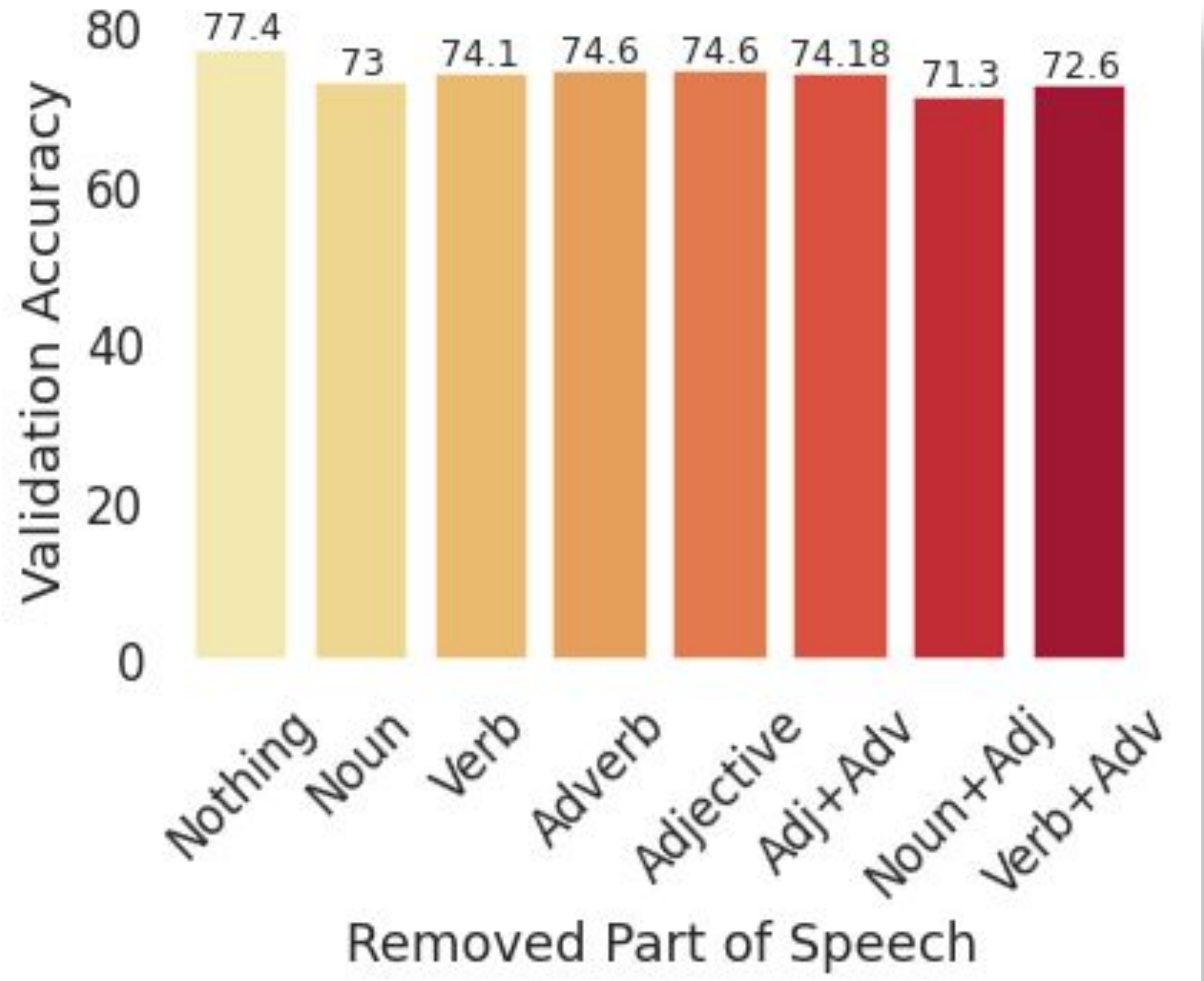
1D Max Pooling

Bidirectional(LSTM)

Dense

Results/Discussion

We found that removing nouns from the training data resulted in the greatest decrease in accuracy for a single lexical class. Since there wasn't a significant drop in validation accuracy after removing any part of the speech, it goes to show that language is very complex, and each component is rarely enough on its own to sway the sentiment of the language.



Ideas for Future Work

In the future, we could attempt multi-class classification to classify text by emotion; for example, rather than classifying a tweet simply as "negative", we could determine whether its author expresses anger, sadness, or frustration. Additionally, it would be interesting to see whether the effects of removing a lexical class would be magnified for the new model.

Neha et al. (2021). Twitter Sentiment Analysis Using Deep Learning.
Μαριος Μιχαηλιδης KazAnova. (2017). Sentiment140 dataset with 1.6 million tweets, Version 2.