Natural Language Processing

Final Project Report

Sentiment Analysis on Social Media Data

By
Team WhiteWash
-Ravsimar Singh
-Vaibhav Bajaj
-Sourav Kumar

1 Overview

Our project consists of sentiment analysis on twitter data. The data was obtained from SemEval. For the baseline, we used a SVM to classify tweets into positive or negative sentiments. Extending this, we classified tweets on a 5-point scale, ranging from very negative to very positive and switched our classifier from a simple SVM to a LSTM neural network. Various parameters were varied and tried out to find the optimal classifier for the given data.

2 Working and Results

2.1 Preprocessing and Tokenization

- Usernames indicate to whom the tweet is addressed to. This does not help in sentiment analysis, hence all usernames (@username) are removed.
- Hashtags are often used in tweets and can indicate a topic or sentiment. All hashtags in the tweets are preserved.
- All other special characters are removed. URLs are also replaced with a keyword 'URL'
- All remaining clean tokens are converted to lowercase. An in built tokenizer is used to tokenize the cleaned tweets.

2.2 SVM

Here we used the SVM provided by sklearn to classify the tweets. Initially using CountVectorizer from sklearn we extracted the features out of text and converted all the words in to vectors. Then these vectors were used as input to the SVM. A lot of parameters were tweaked using GridSearchCV and the best parameters were selected. The parameters were as follows:

- Kernel: rbf, polynomial and linear kernel were tested.
- C: The penalty term for deciding the soft margin in SVM. The values that were selected are: 0.001, 0.01, 0.03, 0.05, 0.1, 1
- Gamma: The kernel coefficient was chosen as either 'auto' which sets its value to 1 / n_features or 'scale' which sets its value to 1 / n_features * X.std().

Using the GridSearchCV the best params that we got are as follows:

Kernel	C	gamma
linear	0.03	scale

The accuracy on 3 point scale was as follows:

F Score	Avg Recall	Accuracy
0.652	0.620	0.655

The above results were generated for the baseline of the project. For the final project, SVM was applied for 5 class classification of tweets. Again using GridSearchCV following parameters were tweaked.

- Kernel: rbf and linear
- C: 0.001, 0.01, 0.03, 0.05, 0.1, 1
- : Gamma: $10^{-9}, 10^{-7}, 10^{-5}$ The best params that we got are as follows:

Kernel	C	gamma
linear	0.03	10^{-9}

and the accuracy for this on 3 point scale was as follows:

F Score	Avg Recall	Accuracy
0.265	0.208	0.401

Then LSTM was applied to improve the results we got using SVM in both the cases (2 and 5 class).

2.3 LSTM

As for the final project we used the LSTM from keras to predict the class of a tweet. On top of LSTM softmax was used. First embedding layer was used, then LSTM layers and then dense layer to generate the output. Again in LSTM a lot of parameters were tweaked to get the best possible results. The parameters that were tweaked as follows(for 2 class classification):

- No. of Epochs: The no of epochs were varied and the values that we checked upon are: 5, 7, 10, 15. The model was over-fitting on 10 and 15 so we got huge training accuracy when no. of epochs were 10 and 15 but due to over-fitting the model under-performed on testing.
- Batch Size: While training the model i.e. while fitting the data the data was divided into batches of different sizes. The values that were checked are: 16, 32, 64
- Dropout: For linear transformation of inputs no of units to drop. Initially we tried with 0.0 and the model was under-performing then we changed it to 0.2 which was the most optimal value we got.
- Embedding layer output dimension: This is the dimension of output of the embedding layer of the neural network. The values that were checked upon are: 128, 256.
- LSTM output dimension: This is the dimension of output layer of the LSTM and the values that were checked upon are: 196, 300.

After tweaking the above parameters the optimal parameters we got were:

No. of Epochs	Batch Size	Dropout	Embedding Dim.	LSTM Dim.
7	32	0.2	256	300

- No. of Epochs: 7
- Batch Size: 32
- Dropout: 0.2
- Embedding layer output dimension: 256
- LSTM layer output dimension: 300

The results on 3 point scale for 2 class classification are as follows:

F Score	Avg Recall	Accuracy
0.770	0.639	0.761

The results on 3 point scale for 5 class classification are as follows:

F Score	Avg Recall	Accuracy
0.450	0.277	0.446