Assignment: EmoInt

Introduction

The assignment is to predict the intensity of expressions in a selected tweets. The intensity scores are values between 0 and 1, representing low and high intensities of the emotion being expressed, respectively.The emotions analyzed in this dataset are anger, fear, joy and sadness.

This task describes the techniques used to analysing cleaning tweets, sentiment on tweets, to produce a vector representation of the tweets and train data using Machine Learning regression and Neural-network regression models to fit the vector representations to predict intensity scores.

Data Preprocessing/Cleaning

Tweets, in general, are not syntactically well-structured and the language used doesn't follow to grammatical rules.There is a necessity to clean the raw text in order to filter noisy data including special characters, alphanumeric strings, tags etc. All tags with '#', '@', 'https' are removed with the help of regular expressiom, Stopwords are removed using NLTK and punctuations using String. All the text is being lower cased and wordnet tokenizer is used.

Feature Extraction

For feature extraction from the tweets I used three techniques to get the sentiments and emotions intensity

1) TextBlob
2) VADER
3) SentiWordNet

Model Training

For model Learning two techniques are used Machine learning Model and Deep Learning Model.
In Machine Learning two models were worked upon Random Forest and Extreme Gradient Boosting after cleaning raw text and converting into vectors using CountVectorizer and parameters like n_estimators, learning rate etc are tuned accordingly.

In Deep Learning (Keras) I have chosen Recurrent Neural Network and LSTM model for training. For this model to train we need to convert the text to floating point tensors with the help of word embedding and in tensorflow and keras it is done by using embedding layer, before that we have converted our array to 2D numeric vectors using tokenizer and padding sequence. Model is compiled with RMSProp optimizer (we can use 'adam' as well) and for regression values between 0 to 1, 'sigmoid' activation function at the output layer with loss as 'MSE'.

Results and Conclusion

XGBoost comes out to be better model when comes to performance and time complexity and is chosen for final prediction. The evaluation is done on the basis of Pearson(P) and Spearman(Sp) correlation, According to the prediction on development set XGB gives P=0.366 and for LSTM P=0.358. XGB predictions were submitted to Codalab where the evaluation comes out to be P=0.322 and Sp=0.308. There is a scope of improvement with the metrices using word embeddings like Word2Vec or GloVe models which are assumed to be best performing feature for most of the emotions to convert text from pre trained large word corpora and can work with hidden layers neurons in LSTM or using CNN.

Codalab ID: **Shub24**