NLP Assignment 3

Akhilesh Ravi 16110007

The given task is to classify tweets into three classes: positive, neutral and negative. The tweets are code-mixed - contain English words and Hindi words in the Latin script. The tweets also contain emojis. Usually, the emojis almost directly represent which of the classes the tweet would fall into.

I have used these methods to convert the tweets into vectors:

Method 1:

a. Use all emojis directly as unicode characters.
b. Use the Hindi and English words directly.
c. Remove all stop words and all words containing non-alphanumeric and non-emoji characters.
d. Run the keras tokenizer directly to convert the sentences into vectors
e. Use a decision tree, multi-layer perceptron and a 1-D CNN model separately to obtain the results.

Method 2:

a. Make a list of the most frequently used 50 emojis[3]. Create a vector that stores the count of each of these emojis in a tweet. Concatenate this vector to the vector created in Method 1.
b. Same as <Method 1 Step e>.

Method 3:

a. Use all emojis directly as unicode characters.
b. Use the Hindi and English words directly.
c. Remove all stop words and all words containing non-alphanumeric and non-emoji characters.
d. Create 4 vectors:
    i. One is the same as in Method 1.
    ii. The next one uses only the English words used in the tweet. Fit and apply the keras tokenize to generate the vector.
    iii. The third one uses only the Hindi words in the tweet. Fit and apply the keras tokenize to generate the vector.
    iv. The last one uses only the words termed ‘O’ in the tweet. Fit and apply the keras tokenize to generate the vector.
e. Concatenate all these vectors.
f. Use a decision tree, multi-layer perceptron and a 1-D CNN model separately to obtain the results.

Method 4: Similar to method 1. But, instead of adding the emojis directly, they have been converted to text using the “emoji” library of Python.

The best model used was the method 1 with CNN-based model - accuracy: 56.5%
The CNN model for method 3 and method 4 give accuracy 55%-56%

The evaluation of the best model is as follows:

	Precision	Recall	F1-Score
Micro	56.5 %	56.5 %	56.5 %
Macro	56.59 %	58.4 %	56.63 %
Weighted	56.76 %	56.5 %	55.7 %

Architecture of the Best Model:

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_12 (Embedding)     (None, None, 300)         6000000
_________________________________________________________________
spatial_dropout1d_11 (Spatia (None, None, 300)         0
_________________________________________________________________
conv1d_9 (Conv1D)            (None, None, 300)         270300
_________________________________________________________________
lambda_9 (Lambda)            (None, 300)               0
_________________________________________________________________
dense_46 (Dense)             (None, 300)               90300
_________________________________________________________________
dropout_32 (Dropout)         (None, 300)               0
_________________________________________________________________
activation_46 (Activation)   (None, 300)               0
_________________________________________________________________
dense_47 (Dense)             (None, 3)                 903
_________________________________________________________________
activation_47 (Activation)   (None, 3)                 0
=================================================================
Total params: 6,361,503
Trainable params: 6,361,503
Non-trainable params: 0
_________________________________________________________________
```

References:

1. Sklearn documentation
2. Kim, Yoon: Convolutional Neural Networks for Sentence Classification
3. https://www.kaggle.com/thomasseleck/emoji-sentiment-data