1. Introduction

Sentiment Analysis using Neural Network architecture and comparison with standard machine learning algorithm. Here LSTM (Long Short-term memory) is chosen as sequence learning neural network model and Naïve Bayes is chosen as standard approach. Dataset used was IMDB large movie reviews (source - https://ai.stanford.edu/~amaas/data/sentiment/).

Libraries Used - Tensorflow.Keras (LSTM), Scikit-learn (Naïve Bayes), NLTK (NLP)

2. Data Pre-processing [Reference 3,5]

2.1 LSTM

- Encoding: All the reviews are encoded to UTF-8 format.
- Lowercase: All the reviews are converted to lowercase for matching the words better.
- Punctuations and special characters removal: All the special characters and punctuations in the review are removed as they are noise.
- Sequence Vectorising: In order to make the textual data accessible by RNN algorithms its converted into numeric vector where each value represents the rank given to the word based on its frequency in the corpus. Also, the position of each value in the vector corresponds to position of the words in the review, this preserves the sequence.
- Sequence Padding: Variable input lengths are adjusted using padding or truncation if the review is shorter on longer than the threshold. Here each review length is set to 500.

2.2 Naïve Bayes

In addition to Encoding, Lowercase, Punctuations and special characters removal below steps are done.

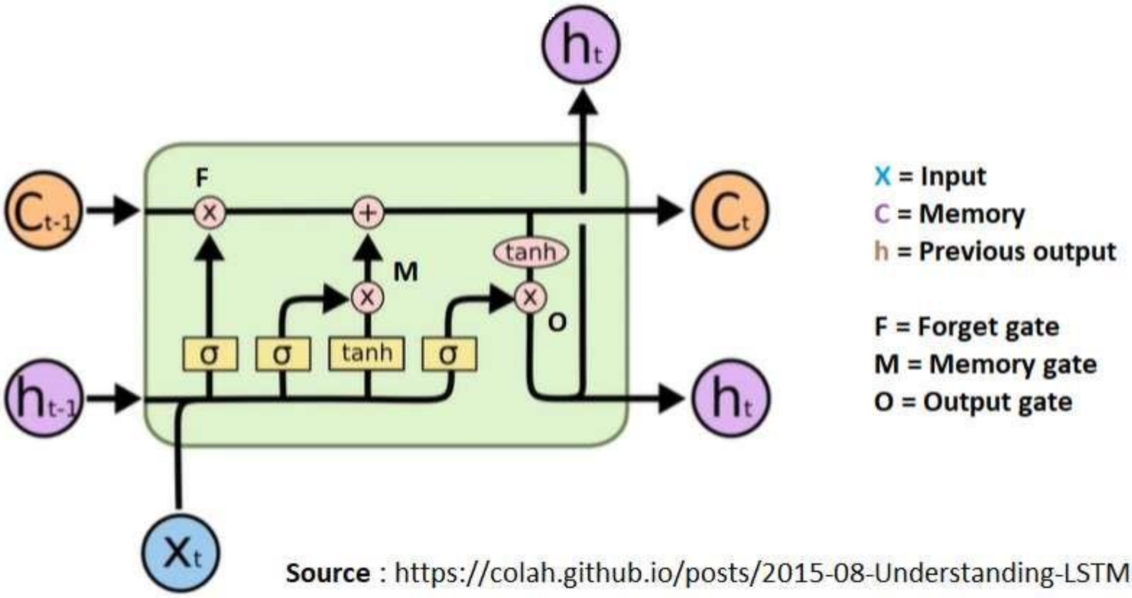
- Stop word removal: Frequently occurring words in the English grammar that doesn't help us better in classification are removed. Ex: The, and, there etc.
- Stemming: Words are reduced to their base form by removing prefixes/suffixes. Ex: Studied □ stud
- Count Vectorising: In order to make the textual data accessible by algorithms its converted into numeric vector where each value represents the frequency of the word in a review.

3. Long Short-Term Memory (LSTM)

It's a Recurrent neural network architecture that is used for sequential modelling. It is widely used in the industry for various NLP tasks (typically classification). Sentiment analysis can be considered as classification task and LSTM layer helps to learn the sequential pattern present in the review to label the polarity. It is more suitable to study long-term dependency and overcomes the vanishing / exploding gradient problem faced by standard RNN.

3.1 One to Many approach [Reference 1]

Here the approach followed for sentiment analysis is one-to-many. Standard LSTM consists of, Forget, Memory and Output gate. Forget gate decides the information that need to be passed over from the previous timestep, while Memory gate adds the new bit of information from the current timestep to the existing information. These two gates constitute a memory conveyor that runs across the LSTM units in sequence. LSTM cell is shown in the below figure,



Source : https://colah.github.io/posts/2015-08-Understanding-LSTMs/

3.2 Model & Parameter details [Reference 2,4]

1. Input layer: Input layer is the padded sequence vectorizer.

2. Embedding layer: Words embedding is useful in finding the relationship between the words based on features/context. Commonly used embedding technique is Word2Vec, it's a single layered NN where each word is converted into a vector based on features. Embedding vector dimension is 25 i.e., a single word is converted to 1D vector of size 25 and each value represents the corresponding feature relevance.

3. Dropout layer: Overall 30% of the input units are dropped to prevent model from overfitting. The same is used after LSTM layer.

4. LSTM layer: Total units is 50, which represents the output dimension for each review from LSTM layers.

5. Dense layer: Final Layer used to find the polarities uses a single unit. Since we are dealing with classification, sigmoid activation function is more suitable to scale the values in the range 0 – 1.

4. Naïve Bayes

Naïve Bayes assumes conditional independency between the predictor variables given the target class. Most of the time this assumption doesn't hold and still Naïve Bayes gives better result, hence it regarded as the Naïve approach. Its derived based on the Bayes rule/ theorem.

4.1 Multinomial Naïve Bayes [Reference 3]

It's the variant used when the features are discrete. In case of sentiment classification, we supply word frequencies as input features. It is the ideal algorithm for document classification.

$$P(x_1,\ldots,x_n \mid c) = P(x_1 \mid c) \bullet P(x_2 \mid c) \bullet \ldots \bullet P(x_n \mid c)$$

$$= \prod_{x \in X} P(x \mid c)$$

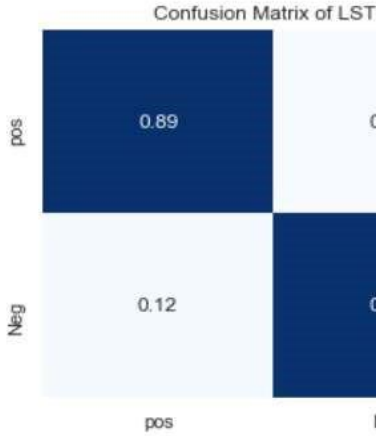
$$c = \underset{c_j \in C}{\operatorname{argmax}} P(c_j) \prod_{x \in X} P(x \mid c_j)$$

In the formula shown, c refers to target class and x1, x2, … xn represents the features. Finally, the class is assigned for the test data based on maximum likelihood P(x|c). When we encounter zero probabilities Laplace Smoothing is applied to overcome.

5. Model Evaluation

In both the models 30 % of data is reserved for testing i.e., Train – 35,000 & Test – 15,000. Also, vectorization keeps top 7500 words in the corpus based on their frequency.

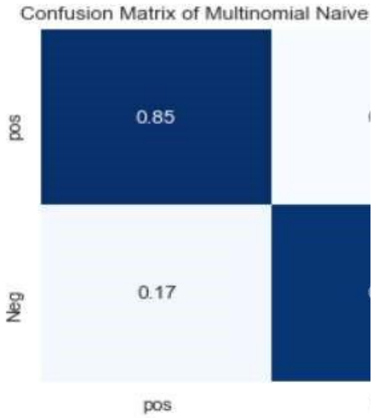
5.1 LSTM



Classification report				
	Precision	Recall	F1-score	Support
Positive	0.88	0.89	0.89	7442
Negative	0.89	0.88	0.89	7558
Accuracy	0.89			

LSTM can predict both instances precisely and classifies the actual review to its corresponding polarity with high recall. Overall the accuracy and F1 score are 89%.

5.2 Naïve Bayes



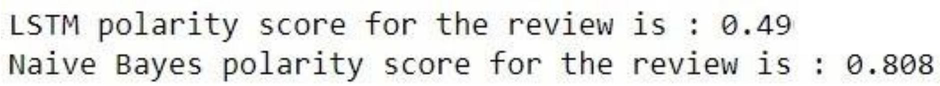
Classification report				
	Precision	Recall	F1-score	Support
Positive	0.83	0.85	0.84	7442
Negative	0.85	0.83	0.84	7558
Accuracy	0.84			

Naïve Bayes is also to predict negative instances more precisely and labels actual positive reviews correctly with high recall. Accuracy and F1 score are 84%.

5.3 Model Performance for unseen events

Negative review – "'A movie of a cynicism so vast and pervasive as to render the viewing experience even emptier than its slapdash aesthetic does." [Source: Rotten tomatoes]

Sentiment lexicons present in the review are ‘emptier’ & ‘slapdash’. This review was selected because these sentiment lexicons were not part of the vocabulary in the corpus.



```
LSTM polarity score for the review is : 0.49
Naive Bayes polarity score for the review is : 0.808
```

LSTM was able to model this unseen event better than Naïve Bayes as it assigns low probability for this review. Probability difference is also huge, and it signifies the complex pattern learned by LSTM.

6. Conclusion

Comparing the 2 models its evident that LSTM has more predictive power than Naïve Bayes. We also see that TP & TN rate are high and FP & FN rate are low for LSTM compared to Naïve Bayes. Sequential learning Neural network model can model the underlying pattern better than standard approaches when we have huge amount of data.

7. References

1. https://colah.github.io/posts/2015-08-Understanding-LSTMs/ [LSTM explanation]
2. https://machinelearningmastery.com/sequence-classification-lstm-recurrent-neural-networks-pythonkeras/ [LSTM implementation for IMDB dataset]
3. https://medium.com/@cristhianboujon/how-to-list-the-most-common-words-from-text-corpus-using-scikitlearn-dad4d0cab41d [Bag of Words implementation, count vectorizer]
4. https://towardsdatascience.com/understanding-lstm-and-its-quick-implementation-in-keras-for-sentimentanalysis-af410fd85b47 [Keras - Sequence Vectorizer implementation]
5. https://www.guru99.com/stemming-lemmatization-python-nltk.html [NLP, text pre-processing]