Sarcasm Detection with Emojis on Tweets and Sentiment Analysis of Tweets

Ajeya B S PES1201701604 *CSE dept. PES University.*
Sirajahamed N D PES1201701496 *CSE dept. PES University.*
Amruth Karnam PES1201700266 *CSE dept. PES University.*
Prof. Ashwini M Joshi, *Assistant Professor, CSE dept. PES University.*

Abstract:-With social media booming, there is always the increase of data. There is an exponential increase in the amount of data created every day.
One form of data is textual data. Social media platforms like Twitter host a huge amount of textual data.
During the past few years, there has been an increase in opinionated textual data in social media platforms like Twitter.
Many of the tweets are directed towards a person, a body or a company. The companies might want to know the reactions from the users' side to make amends and thereby better their products. Sentiment analysis helps us in this cause.
In this paper, we go through the methods employed to analyze the sentiments of tweets of various playstore applications.
Written text many of the times lacks the emotional perspective and the same feeling that a face to face conversation would have. This effect is nullified to an extent by the use of emojis.But nowadays , the emojis used in text are opposite to that of the emotion conveyed by the words in the text, that is, the emojis are used in a sarcastic manner.
Many algorithms which are used to detect sarcasm in textual data usually ignore the emojis, which adds a lot of weight to the overall emotion conveyed in the text.
In this paper, we also go through the methods used by us to recognize sarcasm in tweets with emojis.

Keywords:-
Sentiment Analysis, Textblob, Kmeans, Sarcasm detection , LSTM , Bi-LSTM , Attention Model

I. INTRODUCTION

Natural Language Processing acts as a platform between the computer and humans. It helps in making the machine understand and analyze the data. One of the most important fields in NLP is Sentiment Analysis and its subfield Sarcasm Detection.[1]

Sentiment Analysis is the process of analyzing the opinions expressed by the writer and determining the attitude towards the topic.[1][2] It provides a quick understanding of the writer's attitudes. It is sometimes known as opinion mining where it speaks about a particular entity and discusses the feedbacks of it. [3]With the recent advances in deep learning, the ability of algorithms to analyse text has improved considerably. Creative use of advanced artificial intelligence techniques can be an effective tool for doing in-depth research.

Sarcasm is an indirect manner of conveying a message. It contradicts the meaning, in the context which is said. Sarcasm detection is a specific case of sentiment analysis where instead of detecting a sentiment in the whole spectrum, the focus is on sarcasm.[1]

The reviews of the various apps on the playstore play an important role in analysing the popularity of those apps and concerns regarding it. This helps the particular company to make changes to the same to improve its performance in the market.

Also the tweets and posts on various social media can contain sarcasm in them,may not be directly interpreted by simple means which can lead to faulty analysis.This requires a special technique to distinguish them from the general comments which mean what they say.

This is why these two fields are being rigorously been researched to improve their detection to help better perform than the normal means.

So,we have in this paper tried to explore various techniques and methods in accomplishing the above mentioned goal.
For Sarcasm Detection of the tweets a simple LSTM[4] model was tried which didn't yield good results with respect to various performance metrics.

Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems.[5]This method improved the results considerably giving accuracies of 85%.

But in wake of improving the model performance further we stumbled upon the Attention[6] technique which is the cognitive process of selectively concentrating on one or a few things while ignoring others.This gave us really good results of Accuracy,F1 score,Recall and Precision all near 96%.

II. Literature Review

A. Preprocessing techniques

- Vairaprakash Guruswamy and Subbu Kannan[7] mention the well-known preprocessing techniques for text mining. They mention the importance of preprocessing techniques like stop word removal, stemming, tokenization, which leads to the computer getting better data for processing from the previous unstructured data.

  Limitation: The text we are dealing with are tweets, and they are definitely some acronyms, which are more meaningful if they are converted to their actual meanings. Also some hashtags exist referring to some names and organizations, which does not have any effect on sentiment analysis. So we choose to include these features along with the ones mentioned by Vairaprakash Guruswamy and Subbu Kannan.

B. Sentiment Analysis:

- Hamid Bagheri and Md Johirul Islam [8] use Textblob as well for their sentiment analysis of twitter data.They also use tweepy library, but we choose to stick with Textblob and use KMeans along with textblob.

C. Sarcasm Detection

- Piyoros Tungthamthiti et al.[9] use the SVM approach for sarcasm detection. But even with tri gram models which is the highest value of n taken, they arrive at 79% accuracy, and in their limitations, they say they could not handle words with #.
  So that provides an idea of removing hashtag from our data.

- Jeffrey Pennington et al.[10] explain how GloVe Model works , how it is a term-term matrix rather than a term-document matrix, and shows how it performs better than other baseline models like Word2Vec.The GloVe model we use has 6 billion tokens and 300 dimensions.

- Jenq-Haur Wang et al. [11] mention how RNN might lead a vanishing weight or exploding weights problem, and hence suggest an approach where Word Embedding should be followed by LSTM for a better result. But with LSTM which have fixed length vectors, there 'could' be loss of some information. So, we use the Attention Model on top of LSTM, so that minimal information loss occurs.

- C. Du, L. Huang [12] explains how Attention captures the best of both - keyword classification and neural network classification, as Attention assigns higher weight to some words compared to others.

We are talking in the context of Soft Attention here, as Hard Attention would be focussed on one feature and it would be non deterministic. Attention model is tried on top of the LSTM model which leads to a good overall result.

III. Datasets and Preprocessing

A . Datasets

Data is the most important factor for any analysis. Proper and trustworthy data always lead to fruitful and better conclusions.

Some of the problems we faced while finding the proper data:

- Very few datasets in sarcasm detection having labels of sarcasm, and having Twitter Data, and having emojis and in a single domain.
- So, we chose to skip the single domain constraint
- No reliable dataset with emoji tweets having sentiment labels
- Same tweet copied in one such dataset, The same smiley appended to all tweets in another.

Dataset for Sentiment Analysis

- Source - Kaggle[13]

Dataset for Sarcasm Detection

- Source - Github[14]

B. Preprocessing techniques

The following data preprocessing methods are applied:

- Various tweets are filled with NaNs. Such tweets are not of any use, so they are removed
- Hashtags and URLs are removed from the tweets.
- Non ASCII characters are removed.
- Some common acronyms are converted to their full forms.
- Spellchecker library is used so that some misspelled words are converted to the nearest meaningful word.
- All tweets are converted to lowercase characters for homogeneity.
- Word lemmatization is done, which leads to easier classification of tweets.
- Stop Words are removed.

But, on applying the stop word removal method, the performance metrics obtained go down. The reason is, words like ‘not’ are also removed which totally change the sentiment expressed in a tweet.
So, the stop word removal method is not used before we start to work with the solution.

IV. Classifiers/Models

- Sentiment Analysis:
  - K-Means
  - TextBlob
- Sarcasm Detection:
  - LSTM - BiDirectional
  - Attention (Soft Attention)

V. Methodology and Implementation

For both sentiment analysis and sarcasm detection, we tried approaches till we got results which had good accuracy and F1 score.

1. Sentiment Analysis

This section deals with sentiment analysis of tweets on playstore applications.

The data has more than 64000 tweets, and each tweet is associated with a label denoting if the tweet is positive, neutral or negative, the application the tweet is directed to, the polarity and subjectivity of the tweet.

Before applying any model, an important step is preprocess and clean the data.The above mentioned Pre-Processing techniques were applied.

Initially , the K-Means algorithm is applied.
This is a machine learning based approach chosen. As there are 3 types of sentiments, namely positive, negative and neutral, k is taken as 3.
K-Means algorithm is used from the sklearn library. It is an Unsupervised classifier which classifies the data points into the desired number of clusters. The tweets are not directly fed in as input to the algorithm , but instead modified into the TF-IDF matrix and then this is fed as input.

TF-IDF is used to make a feature vector of every tweet. The whole bag of words is seen , and the words occurring in the tweet are made 1.
IDF is calculated as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.[15]

This matrix is fed into the K-Means algorithm.

So the K means classifies the tweets into three different clusters namely the positive, neutral and negative, which is then used to calculate the performance metrics

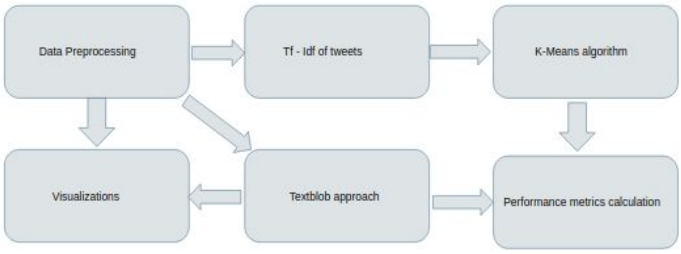
The results obtained which are shown in a later section, are very bad and a better model is required for prediction of sentiments of the tweets.

Next, the textblob library is used to assist in sentiment analysis.
This is a lexicon based approach taken.
Textblob library has functions for extracting properties of sentiments like polarity, and subjectivity.
We make use of the property polarity. We classify the tweets such that , if the polarity we get is greater than equal 0.1, the tweet is classified as positive; if the polarity is less than or equal to -0.1, it is classified as negative; if the polarity is equal to 0, it is classified as neutral.
These classifications we have done on the tweets are later compared with the actual labels provided in the dataset. A confusion matrix is constructed, and with the help of that, the performance metrics, namely accuracy and F1 Score are calculated.
Next, we divide the tweets based on application, and for each application, a similar approach as above is followed, and a confusion matrix and some visualizations are done, and performance metrics are calculated for each application.

The results obtained from the textblob approach is found to be much better than the former model used. Hence, TextBlob is a better approach compared to K-Means algorithm for this data.

In addition to these models, for better understanding of the data, various data visualizations are done which are shown in a later section.

The pathway followed for the Sentiment analysis section is as follows:



Block Diagram of pathway taken for Sentiment Analysis

2. Sarcasm Detection

This section deals with sarcasm detection with tweets having emojis.

The data has more than 5400 tweets, and each tweet is associated with a label, denoting if the tweet is sarcastic or not.

The PreProcessing of the data is performed before applying the model.

We did not try the RNN model, as LSTM is a direct upgrade of RNN, which removes the vanishing and exploding weights problem caused in RNN.

Glove[16]:- The words in the tweets are converted to vectors before they are fed to the model. This can be done by Word2Vec library but it discards all the words which are not in the vocabulary of the training data which is not a good practice. So we make use of a Global word to vector model which has mappings of nearly 6 billion words into 300 dimensions.

Emoji2vec[17]:- The emojis in the tweets also have to be vectorized.This is done by mapping the emojis to corresponding vectors using a emoji2vec model available which has hashing to all of the emojis.

The *Glove* and *Emoji2vec* is used to build a embedding matrix which acts as an initial weights to the embedding layer which is the first layer of the model.

The Embedding layer is followed by the following corresponding approaches in the model

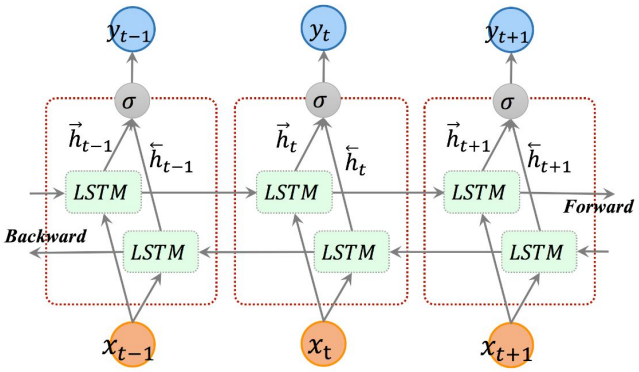
i) Bidirectional-LSTM

Initially, the Bidirectional LSTM Model was tried which is a machine learning based approach.
The tweets are divided such that 80% of the tweets are provided for training, and 20% for testing.

GloVe is used to convert words to vector forms, and the emojis are embedded with the help of emoji2vec. The converted corpus is now fed to the model.

Bidirectional Long Short Term Memory (Bi-LSTM) is a model that is used to connect two hidden layers of opposite directions to the same output.With this form of generative deep learning, the output layer can get information from past (backwards) and future (forward) states simultaneously.

The output of the embedding layer is fed to Bi-LSTM layer in the model which is followed by the Dense layer with the sigmoid activation function which classifies the tweet into 2 classes.



Bidirectional LSTM Model[18]

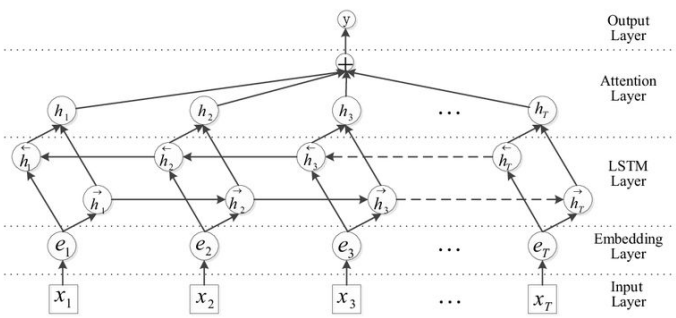
This model was preferred because it gave the model insight about the context which was seen in the past as well as future which leads to better mapping of the sentences to their corresponding classes.

The results obtained from this model are pretty good, but we still try a hybrid model hoping for a better result.

ii) Bi-LSTM with Attention Model

The features given to the dense layers of the model for classification include only the last output state of the intermediate layers which becomes a bottleneck and it becomes very difficult to summarize all of the sentence in a single vector, this can reduce the performance of the system drastically.

But, if the hidden state outputs can be given to the inputs of the dense layers it will give much better relation to as to which words were making impact in deciding the classification of the sentences.This is the idea of the Attention model



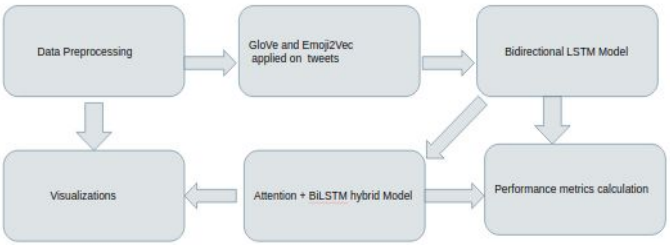
Bi-LSTM with Attention[19]

Hence, the Attention model is used along with the Bidirectional LSTM Model.to nullify any ill effects caused due to the fixed length vectors in LSTM.

The output of the embedding layer is fed to Bi-LSTM layer and Attention model parallelly.These are then joined using a multiplication layer to allow only those features which play an important role. This is followed by a Dense layer with the sigmoid activation function which classifies the tweet into 2 classes.

The results obtained from this model is slightly better than the previous model, and hence we can say that the hybrid model is better to use than the Bidirectional LSTM Model.

The pathway followed for the Sentiment analysis section is as follows:

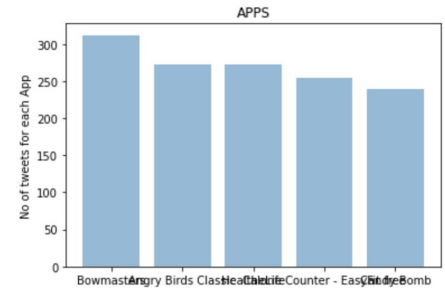


Block Diagram of pathway taken for Sarcasm Detection

VI. Results And Discussions

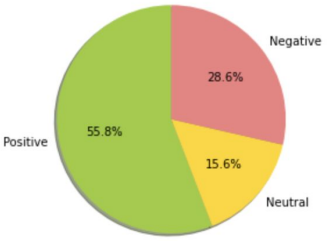
➢ Sentiment Analysis

Initially, a bar chart of apps vs number of tweets is plotted. But as there are more than 800 applications, a bar chart is shown which has the tweets vs number of tweets for the top 5 apps.



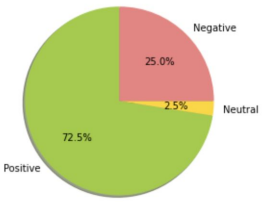
Bar Chart of Top 5 apps

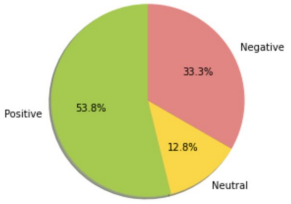
A pie chart is plotted which has the division of the predicted sentiments of the tweets.



Pie Charts of predictions of the tweets

Also, after predictions are done for each application, a pie chart is plotted having the division of the tweets.
As an example, the pie chart is shown for two applications.



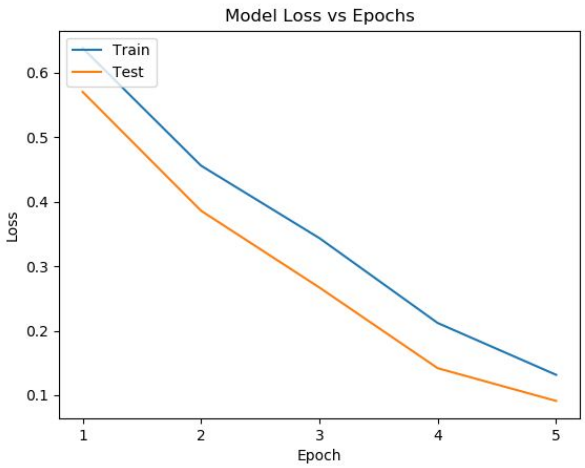


Pie Charts of predictions of two random apps

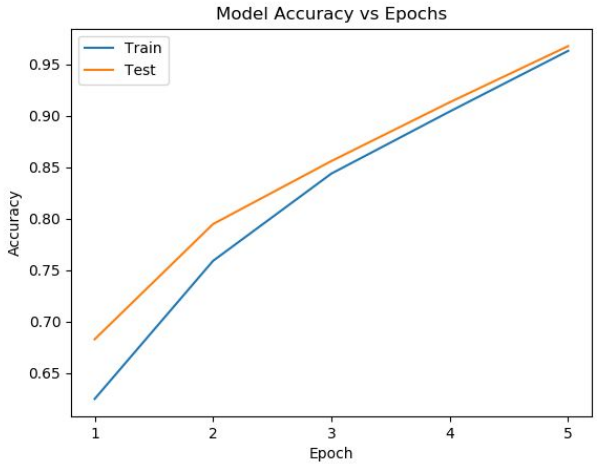
➢ Sarcasm Detection

The hybrid model's improving accuracy per epoch is plotted.

The model's validation loss per epoch is also plotted.



Model loss per epoch



Model accuracy per epoch

Performance of the models are measured with the following parameters:

- Accuracy
- Precision
- Recall
- F1 Score

Accuracy is the measure of percentage of all correctly identified tweets.

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positive.[20]

Recall is the number of true positives divided by the number of true positives plus the number of false negatives [20]

F1 Score is the harmonic mean of Precision and Recall and gives a better measure of the incorrectly classified cases than the Accuracy Metric.[20]

For Sentiment Analysis, a train-test split is not done. Rather the labels are kept separately for validation purposes, and the results obtained from applying the models on the tweets are compared to the labels provided.

K-Means Algorithm:

- Accuracy -> 60%
- Precision -> 31%
- Recall -> 22%
- F1 Score -> 26%

```
The f1Score is =  0.25712699457404603

The Confusion Matix is =
 [[   0  236  928]
 [   0    2  663]
 [   3  201 2967]]

The accuracy is =  0.5938

The recall is =  0.3128915002738635

The Precision is =  0.21849973496081718
```

Performance Metrics of K-Means Algorithm

TextBlob Approach:

- Accuracy -> 85%
- Precision -> 83%
- Recall -> 85%
- F1 Score -> 80%

```
The f1Score is =  0.8027033015125141

The Confusion Matix is =
 [[ 5835  2436     0]
 [    0  5155     3]
 [    1  3108 20889]]

The accuracy is =  0.851764768749833

The recall is =  0.8584476280766866

The Precision is =  0.8271685950116159
```

Performance Metrics from Textblob approach

Hence, from the above results, we can conclude that the TextBlob approach is much better compared to K-Means approach for sentiment analysis of the play store application tweets data.

For Sarcasm detection, a train-test split is done. 80% of the tweets are used for training, and 20% of the tweets for testing.
The labels here are used here for training the model.

Bidirectional LSTM Model:

- Accuracy -> 87%
- Precision -> 83%
- Recall -> 89%
- F1 Score -> 86%

```
Accuracy: 86.666667
F1 Score: 86.165775
Recall: 89.408976
Precision: 83.918042
```

Performance Metrics of Bidirectional LSTM

Bidirectional LSTM + Attention Hybrid Model :

- Accuracy -> 96.5%
- Precision -> 96%
- Recall -> 96%
- F1 Score -> 96%

```
1095/1095 [==============================] - 42s 38ms/step
Accuracy: 96.529680
F1 Score: 96.099147
Recall: 96.227916
Precision: 96.131144
Model Saved
```

Performance Metrics of Bidirectional LSTM + Attention Hybrid Model

Hence, from the above results, we can conclude that the hybrid model is slightly better compared to the Bi-LSTM for sarcasm detection of tweets for the dataset used.

VII. Conclusions

For Sentiment Analysis of play store application tweets,2 models are tried - k-means algorithm and the TextBlob.
The TextBlob approach is shown in the paper to be a much better tool compared to K-Means algorithm for sentiment analysis of the tweets.
Separation of the tweets based on the applications is enabled, so that the result generated from this hybrid system can help the application developers in sensing the general emotion of the users towards the application.

For Sarcasm detection of tweets with emojis , 2 models are tried - Bidirectional LSTM, and a model which is the combination of Bidirectional LSTM and Attention Model.
The hybrid model of Bidirectional LSTM and Attention Model is shown in the paper to be slightly better than the Bi-LSTM model for sarcasm detection of the tweets.
While emojis are considered important sometimes in sarcasm detection, they are not given more importance or than the textual content.
The result generated from this hybrid system can thus help in recognizing if any tweets, one with emojis or not are sarcastic or not.

Acknowledgement

We would like to thank our Chairperson Dr. S Shylaja for her support and valuable inputs to our project.We would also like to thank PES University and CSE Department for giving us the opportunity and encouraging such projects as a part of our course curriculum.

Future Works

- We have seen that clustering algorithms do not work well for Sentiment Analysis.
  As part of the Machine Learning based approach, classification algorithms like Random Forests can be applied on the data to see if any improvements are seen compared to algorithms like K-Means.

- We have seen that the Textblob approach performs pretty well on the data. If the classification algorithms perform well, the Lexicon based model and the Machine Learning based model can be combined into a hybrid model to test if the overall performance of the model increases.

References

[1]https://acadpubl.eu/hub/2018-118-22/articles/22a/63.pdf
[2]https://algorithmia.com/blog/introduction-sentiment-analysis
[3]https://towardsdatascience.com/sentiment-analysis-concept-analysis-and-applications-6c94d6f58c17
[4]https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21
[5]https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python-keras
[6]https://www.sciencedaily.com/terms/attention.htm
[7]https://www.researchgate.net/publication/273127322_Preprocessing_Techniques_for_Text_Mining
[8]https://arxiv.org/pdf/1711.10377.pdf
[9]https://www.aclweb.org/anthology/Y14-1047.pdf
[10]https://pdfs.semanticscholar.org/1baa/3f4fda7c92600a5c192adaed80a834d13ff9.pdf
[11]https://www.aclweb.org/anthology/O18-1021.pdf
[12]https://pdfs.semanticscholar.org/7ac1/e870f767b7d51978e5096c98699f764932ca.pdf
[13]https://www.kaggle.com/saiamogh/google-play-store-analysis/data
[14]https://github.com/jsubram/Sarcasm-Detection-Using-Emoji/tree/master/Data
[15]http://www.tfidf.com/
[16]https://nlp.stanford.edu/projects/glove/
[17]https://github.com/uclnlp/emoji2vec
[18]https://arxiv.org/pdf/1801.02143.pdf
[19]https://github.com/SeoSangwoo/Attention-Based-BiLSTM-relation-extraction
[20]https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c
[21]https://medium.com/@raghavaggarwal0089/bi-lstm-bc3d68da8bd0
[22]https://datascience.stackexchange.com/questions/25650/what-is-lstm-bilstm-and-when-to-use-them
[23]https://stackoverflow.com/questions/56071689/whats-the-major-difference-between-glove-and-word2vec
[24]https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-5-50b4e87d9bdd
[25]https://www.lexalytics.com/technology/sentiment-analysis
[26]https://textblob.readthedocs.io/en/dev/
[27]https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/
[28https://deepai.org/machine-learning-glossary-and-terms/attention-models
[29]https://towardsdatascience.com/having-fun-with-textblob-7e9eed783d3f