FINAL PROJECT REPORT

SENTIMENT ANALYSIS OF TWEETS

MARREDDY SAI NIKHIL REDDY

VELLORE INSTITUTE OF TECHNOLOGY

sainikhilreddym2000@gmail.com

ABSTRACT

In current situation e-commerce markets where shopping and tourism are growing day by day and it is very important to pre-process and analyze a huge amount of data present online. Generally, sentiment analysis is a method to classify web data such as product reviews, views into various polarities such as positive, negative, or neutral. Sentiment analysis using social platforms such as twitter has achieved tremendous results. However, because the data is imbalanced and semantic content it is still a challenge to give proper and effective semantic labeling to the data. Analyzing the tweets can tell a person the intention of tweets, whether it is positive, negative, or neutral. This sentiment analysis plays a major role in many companies. Sentiment analysis is not only used just on tweets but also for different purposes. Companies like Amazon, flip-kart rely on the sentiment analysis of the review posted by the customers on their platform, which helps them to understand the product and the customer needs in a better way. In this project, we perform the investigation using a Kaggle dataset to classify each of the tweets as positive, neutral, or negative sentiment.

INTRODUCTION

In recent times people have started using social networking platforms like twitter..etc. This has generated huge amounts of structured and unstructured data. Processing these huge amounts of data can make us understand the intentions of the user which as a result can help in optimizing user experience which plays a key role. This data requires analysis due to the need to easily and accurately label sentiment classes on a large scale (huge data). Sentiment analysis is the automated process of analyzing text data and sorting it into sentiments positive, negative, or neutral. Using sentiment analysis tools to analyze opinions in Twitter data can help companies understand how people are talking about their brand. Appropriate natural language processing techniques are required to accomplish the goals. Methods that can analyze noisy data are required as the data is on a large scale we may not have it in a structured manner. In the marketing field, companies use it to develop their strategies, to understand customers' feelings, how people respond to their campaigns or product launches. In the political field, it is used to keep track of political views, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well. Sentiment

Analysis also is used to monitor and analyze social phenomena, for the spotting of potentially dangerous situations and determining the general mood of the blogosphere. Sentiment analysis is also called opinion mining. Opinion mining involves analyzing opinions, sentiments, or mentality of the author from the transcription. Online opinions have a great and direct influence on the business of many e-commerce sites. Before buying any product on any online platform it is a general tendency for a user to look at the review of the product and then buy it according to the review given. This way sentiment analysis is very important in the industry.

NECESSITY OF SENTIMENT ANALYSIS:

In present situation where we are suffering from huge data overload, companies might have lot of customer feedback collected and it is impossible for humans to manually and analyse the feedback from the customers without any error. The most important issues from the feedbacks can be identified by using automated process which includes algorithms in machine learning, deep learning which can provide a greater insight into the problem. Analysing the customer feedback will help the companies to improve their products. Traditional sentiment analysis involves using reference dictionaries of how positive certain words are and then calculating the average of these scores as the sentiment of that text. The next step from here is using a simple ML model to make the classification. This is done by generating features from the text then using these features to predict a label. An example of generating features is splitting the text up into words and then using these words and their frequencies in text as features.

USING MACHINE LEARNING:

The label will be a measure of how positive or negative the sentiment is. Once the problem has been set up, mathematical optimization techniques are used to create a model. The key difference here is that you are using ML to assign how positive or negative the features or words are, rather than looking this information up in dictionaries. The traditional ML techniques are able to obtain reasonable results, although suffer from a few problems such as requiring manual work in creating the features, also they do not have a good solution for considering word order. These problems have been addressed by a family of ML techniques known as Deep Learning.

DIFFICULTIES INVOLVED:

In recent years sentiment analysis has been used by many companies and organizations to understand the customer better. To perform the sentiment analysis we get a huge amount of data that contains noisy and unstructured data. To derive the sentiments out of these data, we need to use appropriate natural language processing techniques. There are several elements in a piece of text that factor into sentiment analysis. To obtain complete, accurate, and actionable information from a piece of text, its important to not only identify each of these five elements individually but to also understand how they work together to provide the full context and sentiment. Natural language processing uses machine learning and data mining to provide a more complete picture, but the inherent complexity of language makes it difficult to ensure algorithms accurately analyse tone and context. Factors that limit these algorithms include grammatical nuances, implied meaning from facial expressions and body language, misspellings, ambiguity, and regional or cultural variations in language.

APPROACH:

Sentiment analysis can be performed with different algorithms. They can be performed using machine learning and deep learning algorithms. Machine learning algorithms are good to a certain extent whereas deep learning techniques can dig deep inside the data and understand the patterns from the data. We read in the data set and analyse the data by preprocessing techniques. We only consider the “sentiment” and “tweet_text” column and drop the tweet_id column as it is not necessary. The first thing we do after this is removing the stopwords from the tweet_text column. Generally these words do not contribute and have no value in computing the sentiment of a tweet. After removing the stopwords we also remove the mentions in the text. In order for our deep learning model to understand the text data we convert into numbers. To use the text as input for our model, we first need to convert the words into integers that refer to an index in a dictionary. Only the most frequent words are kept. We clean the text by applying filters and putting the words to lowercase. Words are separated by spaces. Once the dictionary has been created we can convert the text to a list of integer indexes. This is done with the text_to_sequences method of the Tokenizer. These integers are now converted into one hot encoding features which are provided to the model. We also need to convert our target classes to numbers in order for the model to understand. We perform OneHotEncoding to the target labels with the to_categorical() method in keras. We use labelEncoder() function in sklearn to convert the ‘positive’, ‘negative’, ‘neutral’ to their numerical representation. After converting to numbers we apply the to_categorical() method in keras and convert it into one-hot-encoded form. After our data is ready we use the validation set to evaluate the model performance when we tune the parameters of the model. We start with a model with 2 densely connected layers of 64 hidden elements. The input_shape for the first layer is equal to the number of words we allowed in the dictionary and for which we created one-hot-encoded features. As we need to predict 3 different sentiment classes, the last layer has 3 hidden elements. The softmax activation function makes sure the three probabilities sum up to 1. In the first layer we need to estimate 640064 weights. This is determined by (nb inputs * nb hidden elements) + nb bias terms, or (10000 x 64) + 64 = 640064 In the second layer we estimate (64 x 64) + 64 = 4160 weights In the last layer we estimate (64 x 3) + 3 = 195 weights. Because this project is a multi-class, single-label prediction, we use categorical_crossentropy as the loss function and softmax as the final activation function. We run for a predetermined number of epochs and will see when the model starts to overfit. Now, we can try to do something about the overfitting. There are different ways to do that.

1] Reduce the network's size by removing layers or reducing the number of hidden elements in the layers.

2] Add regularization, which comes down to adding a cost to the loss function for large weights.

3] Adding dropout layers, which will randomly remove certain features by setting them to zero.

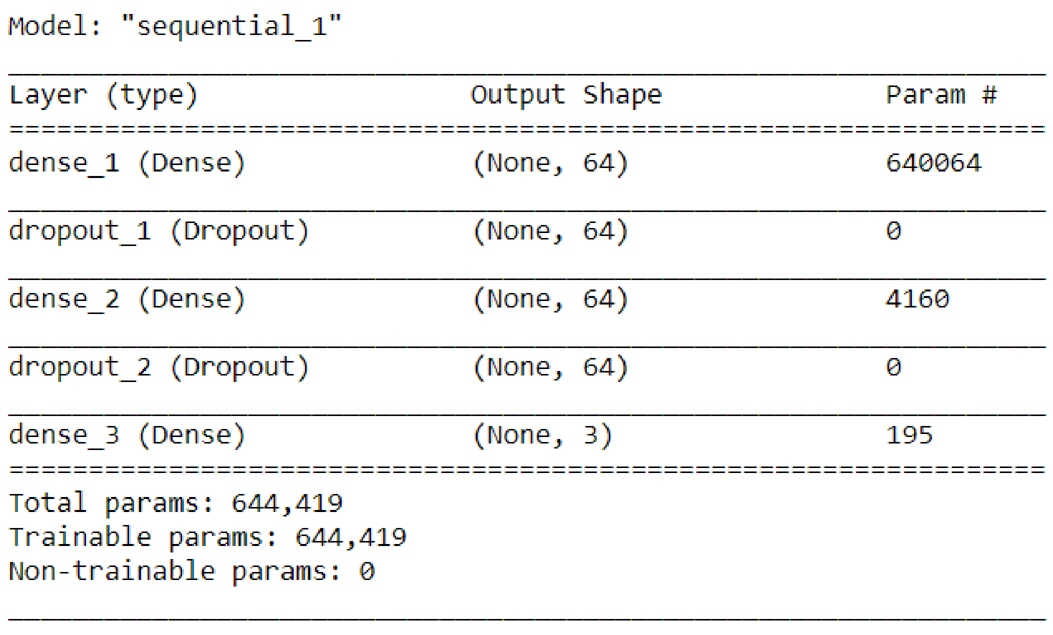
1. Reducing the network's size: We reduce the network's size by removing one layer and lowering the number of hidden elements in the remaining layer to 32. We can see that it takes more epochs before the reduced model starts overfitting (around epoch 10). Moreover, the loss increases much slower after that epoch compared to the baseline model.

2. Adding Regularization: To address overfitting, we can also add regularization to our model. For the regularized model we notice that it starts overfitting earlier than the baseline model. However, the loss increases much slower afterwards.

3. Adding drop out layers: The model with dropout layers starts overfitting a bit later than the baseline model. The loss also increases slower than the baseline model.

From the results obtained by training and testing our model, the model with the dropout layers performs the best on the test data.

MODEL ARCHITECTURE:



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 64)                640064    
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_2 (Dense)              (None, 64)                4160      
_________________________________________________________________
dropout_2 (Dropout)          (None, 64)                0         
_________________________________________________________________
dense_3 (Dense)              (None, 3)                 195       
=================================================================
Total params: 644,419
Trainable params: 644,419
Non-trainable params: 0
_________________________________________________________________
```

RmsProp is used as optimizer, loss function is categorical_crossentropy. As this is multiclass classification we used categorical_crossentropy as loss function and softmax activation layer.

CONCLUSION:

Deep learning algorithms perform very well when compared to conventional machine learning algorithms like SVM classifier, naïve bayes and logistic regression. The architecture used in this project can further be modified in order to achieve better accuracy(better predictions).

REFERENCES:

1] A. Tariyal, S. Goyal and N. Tantububay, "Sentiment Analysis of Tweets Using Various Machine Learning Techniques," 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), Bhopal, India, 2018, pp. 1-5, doi: 10.1109/ICACAT.2018.8933612.

2] Guang Qiu, Xiaofei He, Feng Zhang, Yuan Shi, Jiajun Bu, Chun ChenDASA: dissatisfaction-oriented advertising based on sentiment analysis Expert Syst Appl, 37 (2010), pp. 6182-6191

3] N. F. F. da Silva, E. R. Hruschka, and E. R. Hruschka. Jr., “Tweet Sentiment Analysis with Classifier Ensembles.,” In Decision Support Systems, vol. 66, 2014, pp. 170–179.

4] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification using Machine Learning Techniques,” In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2002, pp. 79-86.