Sentiment Analysis of Twitter Data

A Project Report in partial fulfillment of the degree

Bachelor of Technology

in

Computer Science & Engineering/ Electrical & Electronics Engineering

By

19K41A05F1	B. Vishnu Priya
19K41A05F3	D. Sujitha Rao
20K45A0222	R. Rakesh

Under the Guidance of
D. Ramesh
Assistant Professor of CSE

Submitted to

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
S R ENGINEERING COLLEGE(A), ANANTHASAGAR, WARANGAL
(Affiliated to JNTUH, Accredited by NBA)

Nov-2022

SR Engineering College

Innovation . Creativity . Entrepreneurship

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled "Sentiment Analysis of Twitter Data" is a record of bona fide work carried out by the student(s) B. Vishnu Priya, D. Sujitha Rao, R. Rakesh bearing Roll No(s) 19K41A05F1, 19K41A05F3, 20K45A0222 during the academic year 2022-2023 in partial fulfillment of the award of the degree of **Bachelor of Technology** in **Computer Science Engineering/Electrical & Electronics Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

Supervisor **Head of the Department**

External Examiner

ABSTRACT

Over the past ten years, sentiment analysis and opinion mining have seen significant growth. Customer reviews are one of the most important elements that determine the satisfaction of customers about the products. Also, it gives a comprehensive picture of products to business owners. This field of study makes an effort to ascertain, among other things, what people think, feel, and feel about something or someone. Natural language processing methods and machine learning algorithms are employed in this. The initial goal is to automatically identify a review's orientation and compare it to the judgement the article reviewer made. This would make it possible for researchers to categorize and contrast reviews across the board and more impartially support the overall evaluation of a scientific work. To assess reviews, a hybrid methodology is developed that combines a machine learning algorithm with methods from natural language processing.

Table of Contents

1. INTRODUCTION

The development of technology and the rise of the E-commerce, internet buying, and the information age among internet users has greatly increased. Online purchasing has been a great success, thus there are many clients relocated or changed to alternative buying methods from purchases made online. Despite the benefits that online shopping and ecommerce provide, there are several drawbacks like the requirement for the customer to find measures to assure the quality of the goods since occasionally there are differences a website's merchandise was shown in either an image or a photograph and the product's quality.

Reading or examining customer evaluations before making a purchase is one of the most crucial techniques for customers to verify the quality of products. Users can publish their thoughts and comments on the things they sell on numerous websites, forums, and blogs like Amazon. Reviews from consumers are very useful information for business and marketing owners as well as for the customers themselves. Business and marketing owners should examine customer reviews to determine the best-selling item, the item with the highest rating, the item with the most positive customer feedback, and the item with the most negative feedback.

User-generated content concerning users' opinions about goods or services is known as customer reviews. Additionally, a product or service receives comments or evaluations at a very fast rate. It is now exceedingly challenging for customers and business owners to get a full picture of opinions due to the expansion and rise in customer reviews. To address these issues, natural language processing (NLP) and data mining approaches were developed. There are several approaches that may be used to analyze customer evaluations, but a sentimental analysis is another useful one. Determine whether a piece of literature is favorable, negative, or neutral by performing a sentimental analysis.

A technique to sentiment analysis that combines technology learning NLP methods for polarity or weighted assignment themes, a portion of topics, and entities' sentiment ratings within a word or phrase. Aside from that, Sentimental analysis aids in the ability of data analysts in major businesses and enterprises to customer understanding and public opinion measurement experiences, carry out precise market research, and keep an eye the product's reputation and brand.

Sentiment analysis is frequently used to assess customer satisfaction by assessing opinions of people about various objects, including goods, services, themes, etc. It also goes by the moniker "opinion mining," and it has a broad problem domain. Natural Language Processing (NLP), text analysis, and computational linguistics are all used in sentiment analysis to find and extract subjective information from source materials, such as determining if a review is good or negative. Subjectivity analysis, emotion analysis, review mining, and other terms for similar but slightly distinct activities include analysis of sentiments, opinion extraction, mining of sentiments and opinions, and subjectivity analysis.

Additionally, sentiment analysis can be performed at four levels: word-level, sentence-level, document-level, aspect-level, and concept-level (Appel et al.). The following text is examined to determine the polarity of the affective word, subjective sentence, and entire document, respectively, at the word-level, sentence-level, and document levels of sentiment analysis. Features or elements of a product or any other item that is being analyzed are chosen as targets for aspect-level sentiment analysis, and attitude toward these targets is discovered. Words in opinionated texts with similar meanings are recognized and taken into account as the same notion in concept-level sentiment analysis. Additionally, concept-based sentiment analysis provides new approaches to perform sentiment analysis and opinion mining that go beyond word-level sentiment analysis of text (Cambria et al.). Additionally, concept level methods use semantic networks like Concept Net to categorize text based on its semantics.

Text pre-processing was the original application for natural language processing. These days, NLP applications are widely used. In the domain of NLP, transfer learning is able to train a model on one task and then modify it to execute other NLP functions on other tasks. It is put into practice using pre-trained models. Instead of creating a new model from scratch for our objective, a pre-trained model can be used. This model can be improved a little to conserve computing time and resources. There are numerous pre-trained models that can complete various jobs.

2. LITERATURE REVIEW

[1] Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes 2019 12th International Conference on Information & Communication Technology and System (ICTS). This study tries to classify Surabaya restaurant customer satisfaction using Naive Bayes. Data sampling is crawling by using WebHarvy Tools. The result from this research shows that these two methods get the customer response accurately and Naive Bayes method is more accurate than Text-Blob sentiment analysis with a different accuracy of 2.9%.

[2] Hassan et.al conducted research for improvement of sentiment analysis models for Twitter. They proposed a bootstrapping ensemble framework. They depend on comprising between flowing classifier: SVM, Logistic Regression, Naïve Bayes, Bayes Net, REP Tree, Random Tree, and RBF Neural Network. The result showed a bootstrapping ensemble framework produced a more balanced and accurate sentiment polarity representations in social media.

[3] Wang et.al conducted a study for sentiment classification. They depend on using five classifiers: Support Vector Machine, K Nearest Neighbor, Decision Tree, Maximum Entropy and Naive Bayes. Moreover, they compare the performance of three ensemble methods: Random Subspace, Boosting, and Bagging. The result showed the Random Subspace give the best performance.

[4] Fersini et.al Development a new ensemble method based on Bayesian Model Averaging. They compare five classifiers: Conditional Random Fields, Maximum Entropy, Support Vector Machines, Naïve Bayes and Dictionary. Moreover, they based on using Bagging and Simple Voting as combination rules. The result showed the Bayesian Model Averaging give the best performance.

[5] Chalothom et.al conducted research on Sentiment analysis on Twitter. They depend on using minority voting as a combination rule for the ensemble method. Moreover, they based on the following classifiers: Stacking, SentiStrength Naive Bayes and Support Vector Machine.

[6] Sentiment Analysis with KNN Algorithm proposed by Hyunwoo Max Ch. A machine learning model that can predict if the person thinks positively or negatively about the movie based on the movie review data. To this end, the given data were first preprocessed to turn it into a dataset suitable for training. Then, the machine learning model using the KNN algorithm was

trained and the model was able to predict peoples' sentiment with 82.5% accuracy. Since the above program uses the KNN algorithm, it does not need to know how negative or positive the word.

[7] Twitter sentiment analysis training data contains corpus of already classified tweets in terms of sentiment analysis training and testing where it contains more than 1,500,000 classified tweets, each row is marked as "1" for positive sentiment and "0" for negative sentiment.

[8] Sentiment analysis on travel reviews using three machine learning models namely, Naïve Bayes, SVM and character-based N-gram model has been performed in which SVM and N-gram approaches have better performance than Naïve Bayes. It has been observed that in maximum number of cases SVM showcases best performance in comparison to other classification models.

[9] Fan et al develops a decision support for the vehicle defect discovery by making use of consumer feedback reviews (textual online discussion forums). The research text corpus comes from the 2010 snapshot of the United Stated Department of Transportation, National Highway Traffic Safety Administration (NHTSA), Office of Defect Investigations, vehicle safety complaint database. The min review at least 50 words, and the max review is 8586 words, the review contains an average of 502 words.

[10] Sentiment Analysis with KNN Algorithm proposed by Hyunwoo Max Ch. A machine learning model that can predict if the person thinks positively or negatively about the movie based on the movie review data. To this end, the given data were first preprocessed to turn it into a dataset suitable for training. Then, the machine learning model using the KNN algorithm was trained and the model was able to predict peoples' sentiment with 82.5% accuracy. Since the above program uses the KNN algorithm, it does not need to know how negative or positive the word.

3. DATASET:

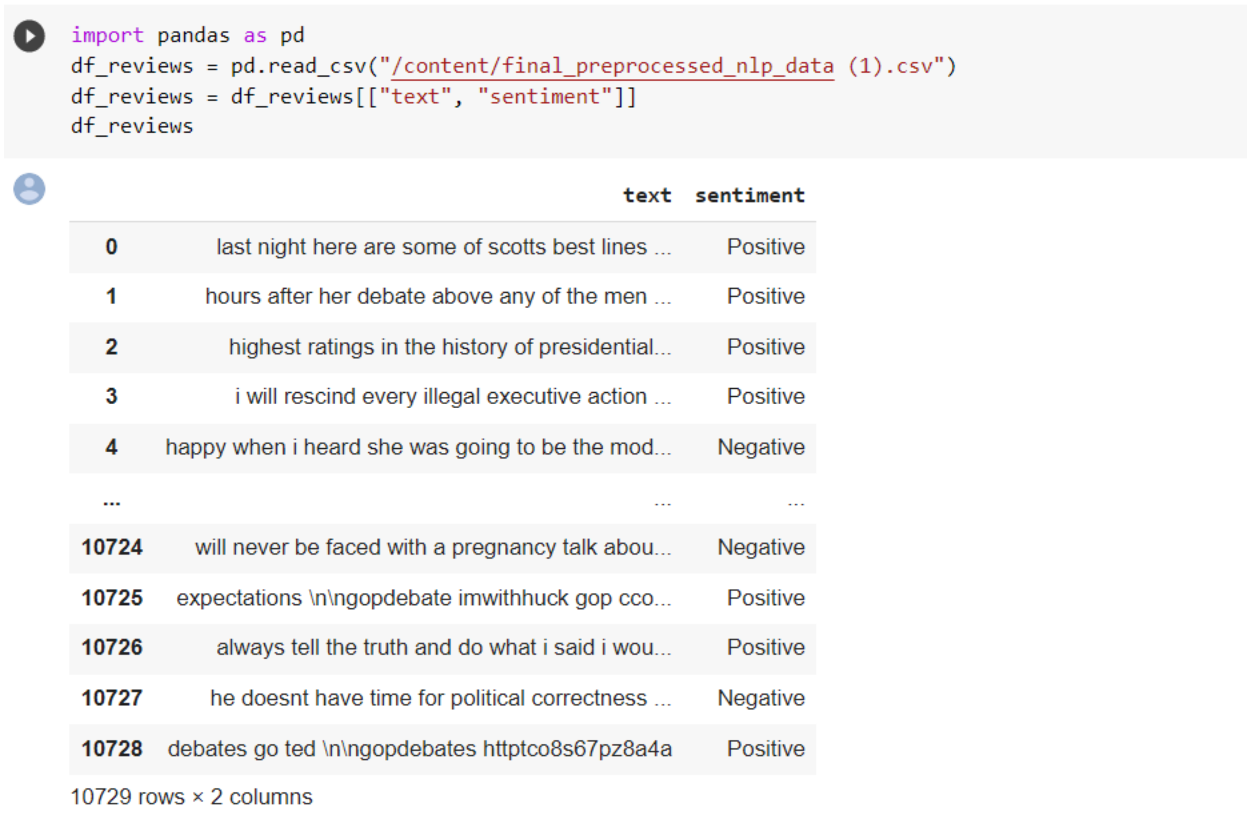
The dataset is based on Twitter tweets from Kaggle which has three columns S.No, Text and Sentiment. Dataset consists of 10729 rows and 2 columns.

Second column: **Text**

This column includes the tweets of the users.

Third column: **Sentiment**

This column describes the sentiment of the tweet like- positive, negative or neutral.

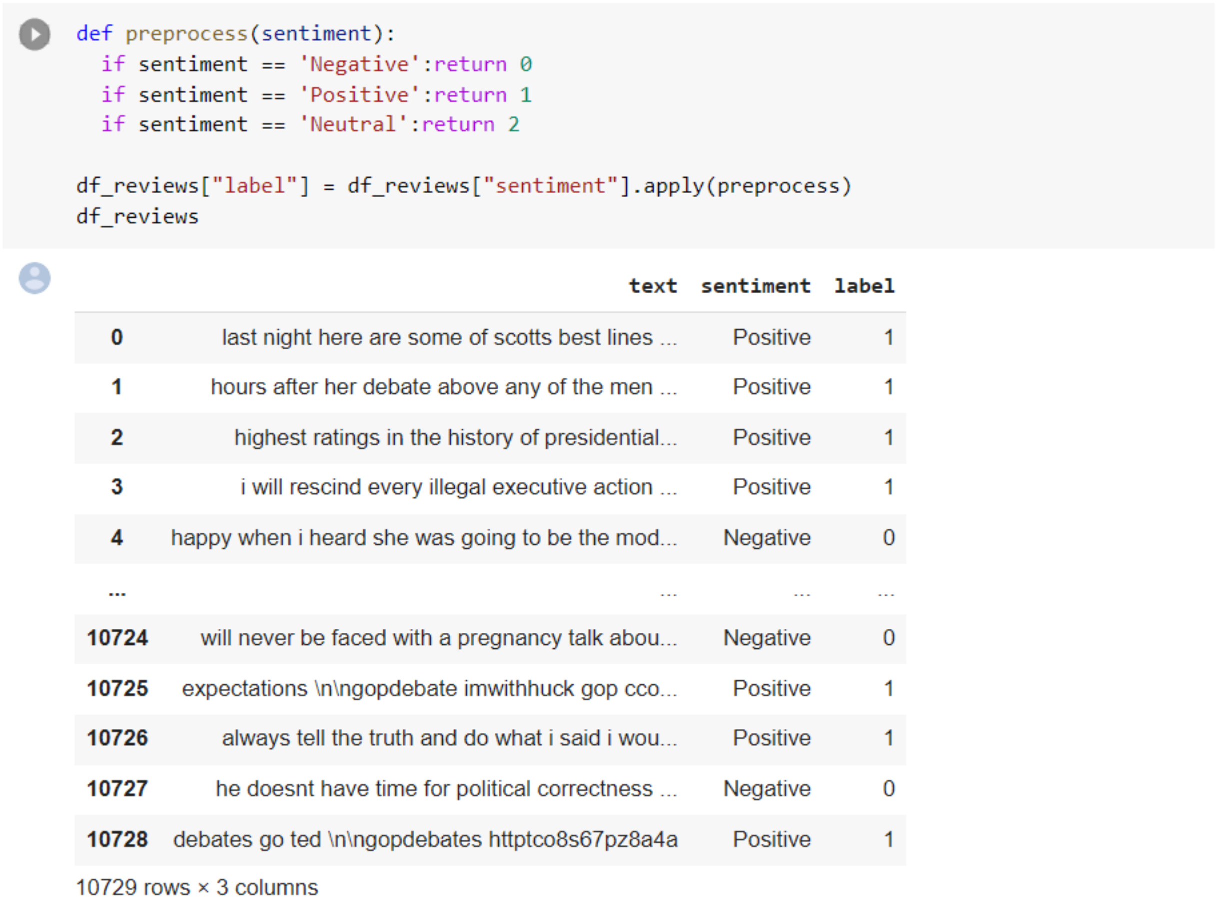


```
import pandas as pd
df_reviews = pd.read_csv("/content/final_preprocessed_nlp_data (1).csv")
df_reviews = df_reviews[["text", "sentiment"]]
df_reviews
```

	text	sentiment
0	last night here are some of scotts best lines ...	Positive
1	hours after her debate above any of the men ...	Positive
2	highest ratings in the history of presidential...	Positive
3	i will rescind every illegal executive action ...	Positive
4	happy when i heard she was going to be the mod...	Negative
...	...	...
10724	will never be faced with a pregnancy talk abou...	Negative
10725	expectations \n\ngopdebate imwithhuck gop cco...	Positive
10726	always tell the truth and do what i said i wou...	Positive
10727	he doesnt have time for political correctness ...	Negative
10728	debates go ted \n\ngopdebates httptco8s67pz8a4a	Positive

10729 rows × 2 columns

Fig 1: Dataset



```
def preprocess(sentiment):
  if sentiment == 'Negative':return 0
  if sentiment == 'Positive':return 1
  if sentiment == 'Neutral':return 2

df_reviews["label"] = df_reviews["sentiment"].apply(preprocess)
df_reviews
```

	text	sentiment	label
0	last night here are some of scotts best lines ...	Positive	1
1	hours after her debate above any of the men ...	Positive	1
2	highest ratings in the history of presidential...	Positive	1
3	i will rescind every illegal executive action ...	Positive	1
4	happy when i heard she was going to be the mod...	Negative	0
...	...	...	...
10724	will never be faced with a pregnancy talk abou...	Negative	0
10725	expectations \n\ngopdebate imwithhuck gop cco...	Positive	1
10726	always tell the truth and do what i said i wou...	Positive	1
10727	he doesnt have time for political correctness ...	Negative	0
10728	debates go ted \n\ngopdebates httptco8s67pz8a4a	Positive	1

10729 rows × 3 columns

Fig 2: Preprocessed Dataset

Dataset after pre-processing we have added label column. Here all the positive and neutral are considered as 1 and negative is considered as 0.

4. DATA PREPROCESSING:

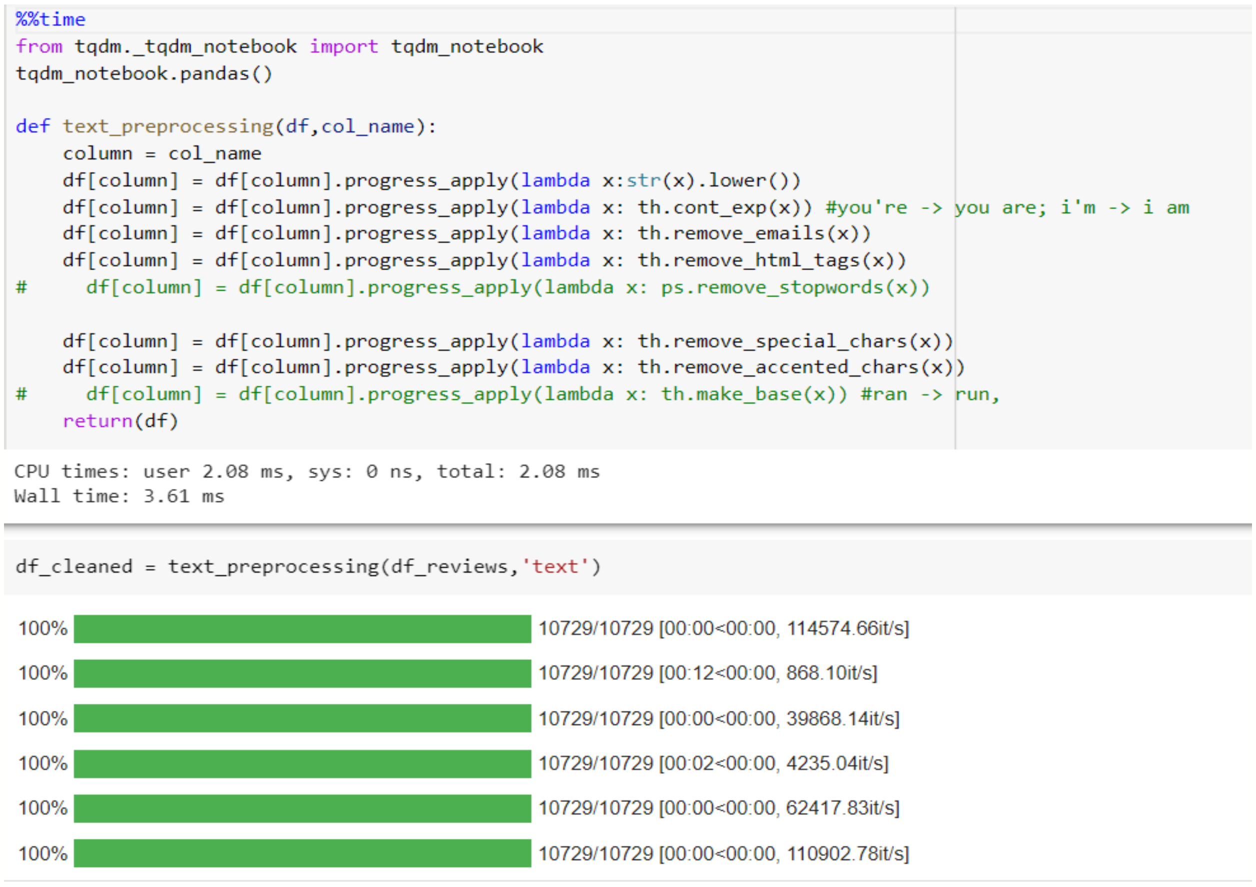
We have used universal sentence encoder and LSTM models to complete this process, where the main goal is to convert the data into vector, perform embedding on it and tokenize it. These are the steps taken for data Pre-processing.

- **Text Hammer**

In any machine learning task, cleaning or pre-processing the data is as important as model building if not more and when it comes to unstructured data like text, this process is even more important. Objective of this kernel is to understand the various text pre-processing steps with code examples. Some of the common text pre-processing / cleaning steps are:

- Lower casing
- Removal of Punctuations
- Removal of Stop words
- Removal of Frequent words
- Removal of Rare words
- Stemming
- Lemmatization
- Removal of emojis
- Removal of emoticons
- Conversion of emoticons to words
- Conversion of emojis to words
- Removal of URLs
- Removal of HTML tags
- Chat words conversion
- Spelling correction

Code:



```
%%time
from tqdm._tqdm_notebook import tqdm_notebook
tqdm_notebook.pandas()

def text_preprocessing(df,col_name):
    column = col_name
    df[column] = df[column].progress_apply(lambda x:str(x).lower())
    df[column] = df[column].progress_apply(lambda x: th.cont_exp(x)) #you're -> you are; i'm -> i am
    df[column] = df[column].progress_apply(lambda x: th.remove_emails(x))
    df[column] = df[column].progress_apply(lambda x: th.remove_html_tags(x))
#     df[column] = df[column].progress_apply(lambda x: ps.remove_stopwords(x))

    df[column] = df[column].progress_apply(lambda x: th.remove_special_chars(x))
    df[column] = df[column].progress_apply(lambda x: th.remove_accented_chars(x))
#     df[column] = df[column].progress_apply(lambda x: th.make_base(x)) #ran -> run,
    return(df)
```

```
CPU times: user 2.08 ms, sys: 0 ns, total: 2.08 ms
Wall time: 3.61 ms
```

```
df_cleaned = text_preprocessing(df_reviews,'text')
```

```
100% 10729/10729 [00:00<00:00, 114574.66it/s]
100% 10729/10729 [00:12<00:00, 868.10it/s]
100% 10729/10729 [00:00<00:00, 39868.14it/s]
100% 10729/10729 [00:02<00:00, 4235.04it/s]
100% 10729/10729 [00:00<00:00, 62417.83it/s]
100% 10729/10729 [00:00<00:00, 110902.78it/s]
```

Fig 3: Data Pre-processing

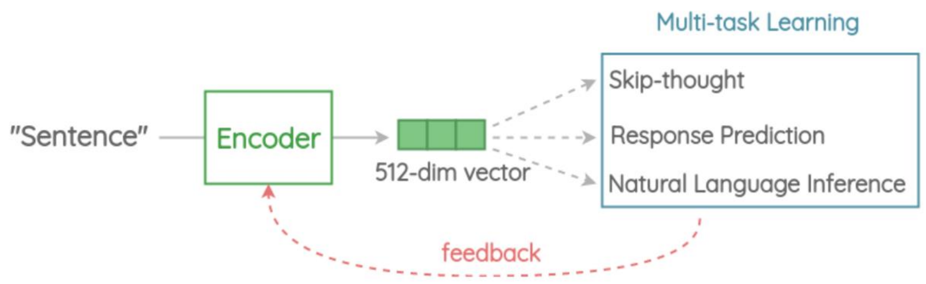
5. METHODOLOGY:

This section talks about the Universal sentence encoder and LSTM models used for the project.

❖ UNIVERSAL SENTENCE ENCODER

The universal sentence encoder makes looking up embeddings at the sentence level as simple as it has previously been to look up embeddings at the word level. Then, using less supervised training data, the sentence embeddings can be easily employed to compute sentence-level meaning similarity and improve performance on subsequent classification tasks. The universal sentence encoder model converts textual information into numerically represented, high-dimensional vectors called embeddings. It aims to transfer learning especially to other NLP tasks like text categorization, semantic similarity, and clustering. The freely accessible universal sentence encoder is listed in Tensor flow-hub. To learn for a wide range of jobs, it is trained on a number of data sources.

On a high level, the idea is to design an encoder that summarizes any given sentence to a 512-dimensional sentence embedding. We use this same embedding to solve multiple tasks and based on the mistakes it makes on those, we update the sentence embedding. Since the same embedding has to work on multiple generic tasks, it will capture only the most informative features and discard noise. The intuition is that this will result in an generic embedding that transfers universally to wide variety of NLP tasks such as relatedness, clustering, paraphrase detection and text classification.



1. Tokenization

First, the sentences are converted to lowercase and tokenized into tokens using the Penn Tree bank (PTB) tokenizer.

2. Encoder

This is the component that encodes a sentence into fixed-length 512-dimension embedding. In the paper, there are two architectures proposed based on trade-offs in accuracy vs inference speed.

Variant 1: Transformer Encoder

In this variant, we use the encoder part of the original transformer architecture. The architecture consists of 6 stacked transformer layers. Each layer has a self-attention module followed by a feed-forward network. The self-attention process takes word order and surrounding context into account when generating each word representation. The output context-aware word embeddings are added element-wise and divided by the square root of the length of the sentence to account for the sentence-length difference. We get a 512-dimensional vector as output sentence embedding. This encoder has better accuracy on downstream tasks but higher memory and compute resource usage due to complex architecture. Also, the compute time scales dramatically with the length of sentence as self-attention has O(n2) time complexity with the length of the sentence. But for short sentences, it is only moderately slower.

Variant 2: Deep Averaging Network (DAN)

In this simpler variant, the encoder is based on the architecture. First, the embeddings for word and bi-grams present in a sentence are averaged together. Then, they are passed through 4-layer feed-forward deep DNN to get 512-dimensional sentence embedding as output. The embeddings for word and bi-grams are learned during training. It has slightly reduced accuracy compared to the transformer variant, but the inference time is very efficient. Since we are only doing feedforward operations, the compute time is of linear complexity in terms of length of the input sequence.

3. Multi-task Learning

To learn the sentence embeddings, the encoder is shared and trained across a range of unsupervised tasks along with supervised training on the SNLI corpus.

❖ LSTM

Long short-term memory (LSTM) is an artificial neural network used in the fields of artificial intelligence and deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. Such a recurrent neural network (RNN) can process not only single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition, machine translation, robot control, video games, and healthcare. LSTM has become the most cited neural network of the 20th century.

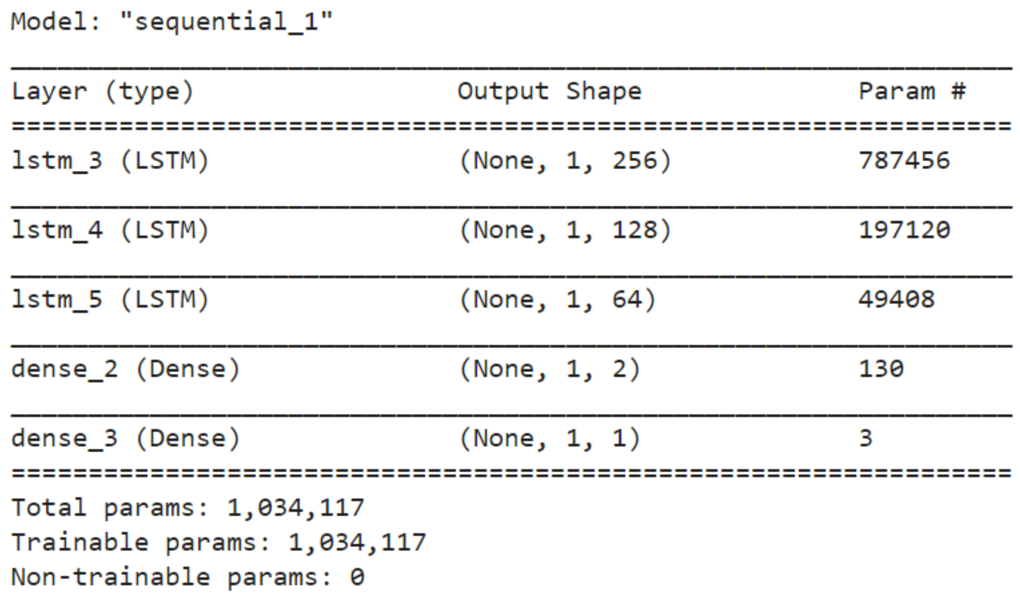
The name of LSTM refers to the analogy that a standard RNN has both "long-term memory" and "short-term memory". The connection weights and biases in the network change once per episode of training, analogous to how physiological changes in synaptic strengths store long-term memories; the activation patterns in the network change once per time-step, analogous to how the moment-to-moment change in electric firing patterns in the brain store short-term memories. The LSTM architecture aims to provide a short-term memory for RNN that can last thousands of timesteps, thus "long short-term memory".

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNNs, hidden Markov models and other sequence learning methods in numerous applications.At a high-level LSTM works very much like an RNN cell. Here is the internal functioning of the LSTM network. The LSTM consists of three parts, as shown in the image below and each part performs an individual function. The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp.

These three parts of an LSTM cell are known as gates. The first part is called Forget gate, the second part is known as the Input gate and the last one is the Output gate**.** Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that LSTM also have a cell state represented by C(t-1) and C(t) for previous and current timestamp respectively.

5.2 Model Architecture



```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
lstm_3 (LSTM)                (None, 1, 256)            787456    
_________________________________________________________________
lstm_4 (LSTM)                (None, 1, 128)            197120    
_________________________________________________________________
lstm_5 (LSTM)                (None, 1, 64)             49408     
_________________________________________________________________
dense_2 (Dense)              (None, 1, 2)              130       
_________________________________________________________________
dense_3 (Dense)              (None, 1, 1)              3         
=================================================================
Total params: 1,034,117
Trainable params: 1,034,117
Non-trainable params: 0
```

Fig 4: Model parameters

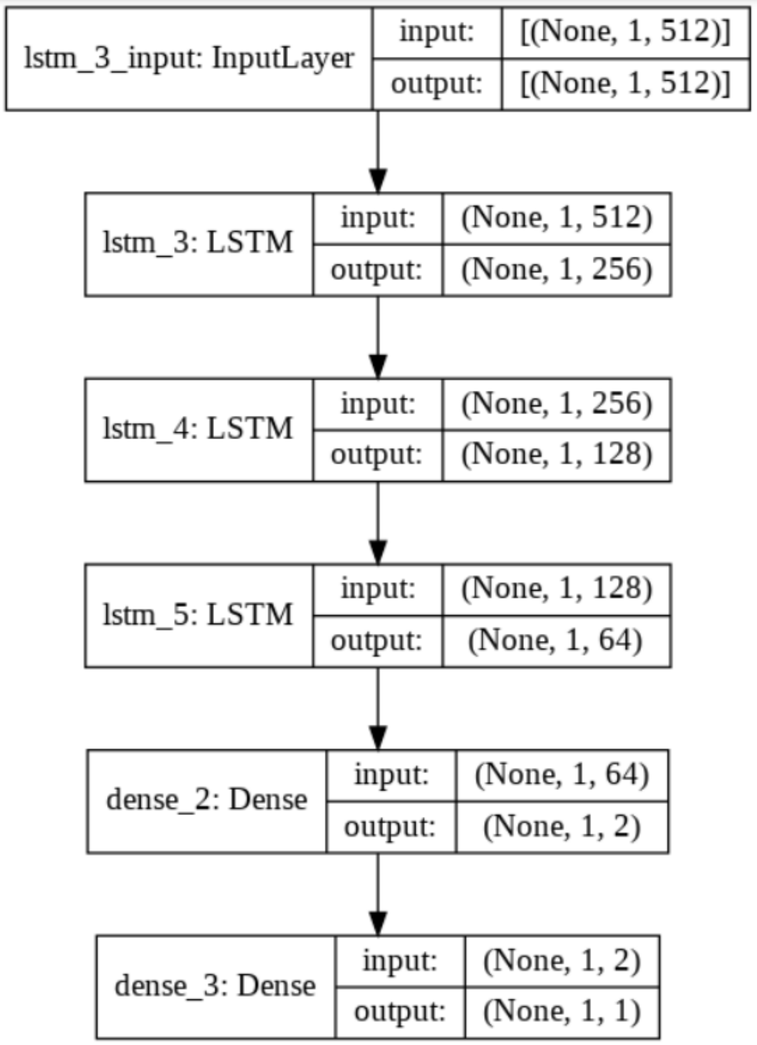


Fig 5: Model layers

6. RESULTS:

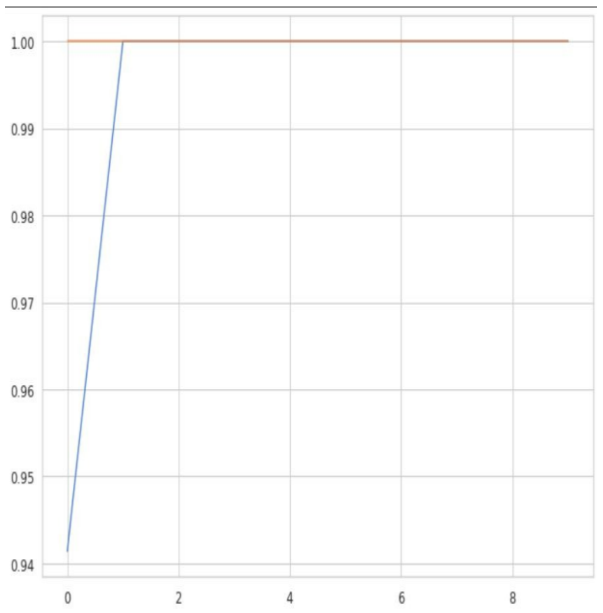


Fig 7: Model accuracy

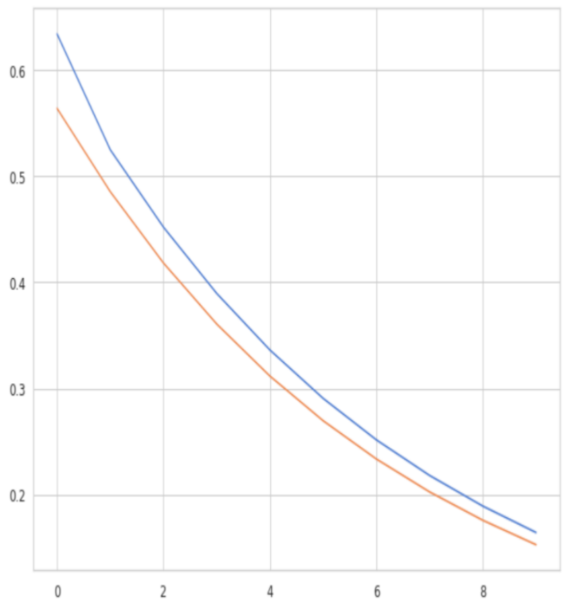
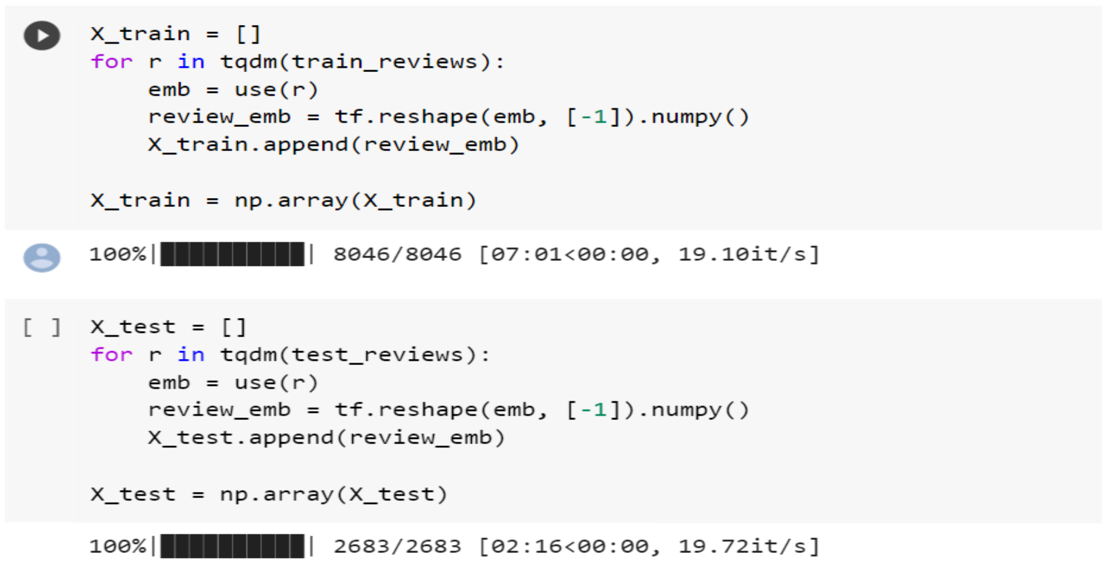


Fig 8: Model loss



```
X_train = []
for r in tqdm(train_reviews):
    emb = use(r)
    review_emb = tf.reshape(emb, [-1]).numpy()
    X_train.append(review_emb)

X_train = np.array(X_train)
```

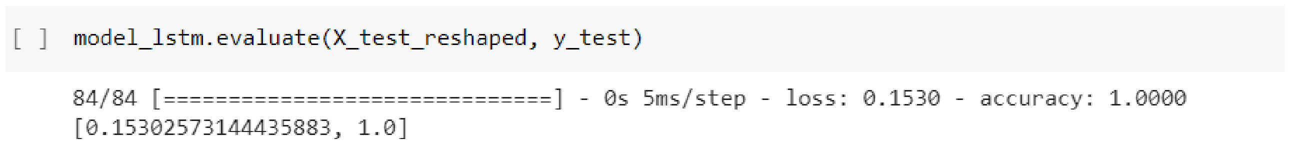
```
100%|██████████| 8046/8046 [07:01<00:00, 19.10it/s]
```

```
[ ] X_test = []
for r in tqdm(test_reviews):
    emb = use(r)
    review_emb = tf.reshape(emb, [-1]).numpy()
    X_test.append(review_emb)

X_test = np.array(X_test)
```

```
100%|██████████| 2683/2683 [02:16<00:00, 19.72it/s]
```

Fig 9: Embeddings



```
[ ] model_lstm.evaluate(X_test_reshaped, y_test)
```

```
84/84 [==============================] - 0s 5ms/step - loss: 0.1530 - accuracy: 1.0000
[0.15302573144435883, 1.0]
```

Fig 10: Accuracy

7. CONCLUSION:

Sentiment analysis is a field of study that analyzes people’s sentiments, attitudes, or emotions towards certain entities. It tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. A sentiment polarity categorization process has been proposed along with detailed descriptions of each step. Experiments for both sentence-level categorization and review-level categorization have been performed. We have considered twitter tweets to analyzed the sentiments and proposed the model with Universal sentence encoder and LSTM. Universal sentence encoder is to embed text into high dimensional vectors of size 512 that is used for natural language tasks. LSTM is capable of learning long term dependencies and predicts future output from sequence of inputs. Finally, we have observed that our technique is accurate and efficient.

8. REFERENCES:

[1] Laksono, R. A., Sungkono, K. R., Sarno, R., & Wahyuni, C. S. (2019). Sentiment Analyzis of Restaurant Customer Reviews on TripAdvisor using Naïve Bayes. 2019 12th International Conference on Information & Communication Technology and System (ICTS). doi:10.1109/icts.2019.8850982

[2] A. Hassan, A. Abbasi, and D. Zeng, "Twitter Sentiment Analysis: A Bootstrap Ensemble Framework," in 2013 International Conference on Social Computing, 2013, pp. 357–364.

[3] G. Wang, J. Sun, J. Ma, K. Xu, and J. Gu, "Sentiment classification: The contribution of ensemble learning," Decis.Support Syst., vol. 57, pp. 77–93, Jan. 2014.

[4] E. Fersini, E. Messina, and F. A. Pozzi, "Sentiment analysis: Bayesian Ensemble Learning," Decis. Support Syst., vol. 68, pp. 26–38, Dec. 2014.

[5] T. Chalothom and J. Ellman, "Simple Approaches of Sentiment Analysis via Ensemble Learning," Springer, Berlin, Heidelberg, 2015, pp. 631–639.

[6] L. Naji, "Twitter Sentiment Analysis Training Corpus (Dataset)," 22 september 2012.

[7] S. Hridoy, M. Ekram, M. Islam and R. M. Rahman, "Localized twitter opinion mining using sentiment analysis," Decision Analytics, vol. 2, no. 1, 22 october 2015.

[8] Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," Expert Systems with Applications, vol. 36, no. 3, pp. 6527–6535, 2009.

[9] A. S. Abrahams, J. Jiao, G. A. Wang, and W. Fan, "Vehicle defect discovery from social media", Decision Support Systems, vol. 54, pp. 87-97, 2012.

[10] B. H. Kasthuriarachchy, K. De Zoysa and H. L. Premaratne, "A review of domain adaptation for opinion detection and sentiment classification", in Advances in ICT for Emerging Regions (ICTer), 2012 International Conference on, 2012, pp. 209-213.