

---

# Intelligent Content Analyzer



## Department of Computer Science

Muhammad Musawar Baig	2018-UET-NML-CS-03	musawar2018@namal.edu.pk
Ali Raza Khan	2018-UET-NML-CS-23	raza2018@namal.edu.pk

**2022**

Final year project report submitted in partial fulfillment of requirement for degree of  
Bachelors of Science in Computer Science

**Namal Institute,  
30-KM, Talagang Road, Mianwali, Pakistan.  
[www.namal.edu.pk](http://www.namal.edu.pk)**

Date: May 20th, 2022

---

# DECLARATION

The project report titled “Intelligent Content Analyzer” is submitted in partial fulfillment of the degree of Bachelors of Science in Computer Science, to the Department of Computer Science at Namal Institute, Mianwali, Pakistan.

It is declared that this is an original work done by the team members listed below, under the guidance of our supervisor “Dr. Junaid Akhtar”. No part of this project and its report is plagiarized from anywhere, and any help taken from previous work is cited properly.

No part of the work reported here is submitted in fulfillment of requirement for any other degree/ qualification in any institute of learning.

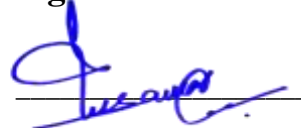
## Team Members

## University ID

## Signatures

Muhammad Musawar Baig

2018-UET-NML-CS-03



Ali Raza Khan

2018-UET-NML-CS-23



## Supervisor

Dr. Junaid Akhtar

## Signatures with date



May 24, 2022

# 1 Abstract

Online information has exploded with the rise of social media platforms (like Twitter) which has made it very difficult to process useful information for the majority of people. As people have a right to freely express their opinions on all sorts of things on social media. Similarly, They also have a right to choose the information that they want to read and block the negative and/or illogical content from their social media life. With this in mind, We tried to build an intelligent content analysis system (Intelligent Content Analyzer) that can automatically analyse comments/replies of tweets of its users and let the users have more control of their Twitter life by giving options to filter out negative or/and illogical replies/comments that they don't want to read and focus on only the positive and logical content. Sentiment Analysis is useful to filter out the negative and useless information if done properly. The current state-of-the-art sentiment analysis systems are not flawless. The majority of them (Vader, TextBlob, and SparkNLP) that we have experimented with fail to classify simple pieces of text (like "You think you deserve congratulations for this kind of work?"). This study was not only about finding the best system and exploring the limits of the current state-of-the-art sentiment analysis systems but also about the exploration and development of a human-centred, intelligent, reasonable, learnable, and interpretable system to perform logical analysis of the replies. A huge amount of work has been already done to perform "sentiment analysis" but none has been done to perform "logical analysis" of the text. We hope this study will open new research areas for future research, especially in the world of logical analysis.

## 2 Introduction

In recent years, especially with the emergence of social media giants (like Twitter), many new challenges and opportunities have arisen. The exponential growth of all kinds of online information has made it difficult for public figures and other professionals (like journalists etc) to keep track of their social media life. Imagine a person that receives thousands of replies in response to their statements posted on Twitter. It's impossible for this person to read all the replies and the replies also include hate speech and negative content that have the potential to negatively affect his/her mental health. A 2014 Pew Research Center study found that 22% (one in five) Internet users had been victims of online harassment in the comment section of a website<sup>1</sup>.

And due to that much negativity and hate speech, people (especially public figures) often try to quit or minimize the use of their social media accounts as much as possible. So, Considering the abundance of information and type of negative information, there is a need for a content analysis system that can automatically analyze comments/replies of tweets of its users and let the user have more control of their social media life by giving options to its users to filter out replies/comments (like hiding all the **negative** or/and **illogical** comments) that they don't want to read.

---

<sup>1</sup> <https://www.verywellmind.com/mental-health-effects-of-reading-negative-comments-online-5090287>

We are building the above system (Intelligent Content Analyzer) as part of our Final Year Project (FYP). Following are our two core goals of our FYP:

1. Use state-of-the-art language processing techniques to explore their limits and use them in building a deployable product.
2. Design and Development of a human-centred Deep-AI Algorithm that will try to resolve the problems with the state-of-the-art Rule-based and Deep Learning-based text processing techniques.

The Sentiment Analysis of textual data (replies/comments) needs to be performed properly to build the envisioned content filtration system (Intelligent Content Analyzer). Sentiment Analysis is a common natural processing task. Following are the two typical approaches used for Sentiment Analysis:

1. Lexicon-based approaches
2. Deep learning-based approaches

We have used and explored the limits of state-of-the-art sentiment analysis systems based on both the above approaches. We have also done comparative analysis in real of both approaches. Speaking of rule-based/lexicon-based sentiment analysis approaches, we have also tested a batch of tweets/comments to calculate accuracy. Vader had an accuracy of **65%** and TextBlob had an accuracy of **55%**. We have observed that rule-based approaches fail miserably to correctly classify simple texts (see Examples section). They fail to take into consideration the complex pattern of human language.

As far as the Deep learning-based approaches are concerned, We have fine-tuned and trained deep-learning-based language models (i.e. BERT, GPT3 and GPT-2) on a manually annotated dataset (RETWEET). We have observed that the granularity of the dataset considerably affects the accuracy of Deep learning models. For example, if we add a neutral class in a previously binary classification problem (of positive and negative classes) then the accuracy of BERT decreases from 99% to just 46%. Alongside, We have also used a pre-trained pipeline for sentiment analysis by John Snow Labs' SparkNLP. We used the model named sentimentdl\_user\_twitter from SparkNLP and it gives **69%** accuracy on the RETWEET dataset.

We have observed that both approaches to calculate sentiment analysis have their flaws. For binary classification, BERT seems like a reliable choice with **95%** accuracy. Considering the flaws of sentiment analysis systems and the present challenges in the field of AI, We have designed and developed a Deep AI Algorithm for the logical analysis of the text that takes the pros of both approaches and is also explainable. The Deep AI Algorithm provides the explainability of rule-based systems and learnability of Deep learning-based approaches. It is developed while keeping in mind the 21st-century challenges and opportunities of current state-of-the-art systems.

## 3 Literature Review

Typically, there are two major approaches used to find the sentiment of tweets. One is the lexicon-based approach, and the other is the deep learning-based approach. The following sections deal with both the famous approaches.

### 3.1 Lexicon-based Sentiment Analysis Systems and Techniques

Following are the state-of-the-art resources for lexicon-based Sentiment Analysis:

#### 3.1.1 TextBlob

It is a famous Python library for processing textual data. It provides a simple API for diving into typical natural language processing (NLP) tasks like part-of-speech tagging, noun phrase extraction, classification, sentiment analysis, translation, etc.

Features of TextBlob are listed below [1]:

- Noun phrase extraction
- Sentiment analysis
- Tokenization (splitting text into words and sentences)
- Parsing
- Word and phrase frequencies
- Word inflection (pluralization and singularization) and lemmatization
- Classification (Naive Bayes, Decision Tree)
- Add new models or languages through extensions
- N-grams
- Part-of-speech tagging
- Spelling correction
- WordNet integration

##### 3.1.1.1 TextBlob Sentiment Analysis

For sentiment analysis, TextBlob gives us two values/measures to analyse the sentiment:

1. Polarity: This value expresses how positive or negative the text is. It ranges between -1 to 1. Closer to 1 is more positive, 0 is neutral, and closer -1 is negative.
2. Subjectivity: This value expresses the subjectivity and objectivity of the text. It ranges from 0 to 1. 0 means very objective, and 1 means very subjective.

#### 3.1.2 Vader

The VADER a rule-based sentiment analysis tool. It's support is available with Python. It stands for Valence Aware Dictionary and Sentiment Reasoner. It is open-source under the MIT licence. VADER measures the sentiment of the given text using a list of lexical features or

words which are labelled as positive or negative according to their semantic orientation. VADER lexicon performs very well in the social media domain [2].

Vader sentiment returns the probability of a given input sentence to be positive, negative, and neutral [3]. It also returns a compound score which is used to classify the overall sentiment of the text.

### 3.1.2.1 Vader Sentiment Analysis

Vader sentiment tells us how positive, negative, and neutral a text is. It also tells us about the intensity value of each emotion.

For Example:

```
analyzerObj.polarity_scores("Ali love flying aeroplanes")
```

The above statement gives the following output:

```
{'neg': 0.0, 'neu': 0.417, 'pos': 0.583, 'compound': 0.6369}
```

The sum of neg, neu, and pos intensity values equals 1. The compound value ranges between -1 to 1. The compound score is a metric that calculates the sum of all intensity values which have been normalized between -1 and +1. It is useful to set a threshold for classifying the text as positive, negative and neutral. Typical threshold values are as follows [4]:

Neutral Sentiment: compound score  $> -0.05$  and  $< 0.05$

Positive Sentiment: compound score  $\geq 0.05$

Negative Sentiment: compound score  $\leq -0.05$

### 3.1.3 Affin

Affin is another simple and popular lexicon used for sentiment analysis. It was developed by Finn Arup Nielsen in 2009 to specifically analyse Tweets [5].

#### 3.1.3.1 Affin Sentiment Analysis

Affin is a wordlist-based approach. The English language dictionary of Affin uses 3300+ words. A polarity score from -5 to 5 is associated with each word. -5 means extremely negative and +5 means extremely positive. A total sentiment score of a text is calculated by summing up all the individual scores of the words of any text (e.g. tweet). Typical threshold values are as follows [6]:

Neutral Sentiment: total score  $= 0$

Positive Sentiment: total score  $> 0$

Negative Sentiment: total score  $< 0$

## 3.2 Deep-Learning-based Sentiment Analysis Systems and Techniques

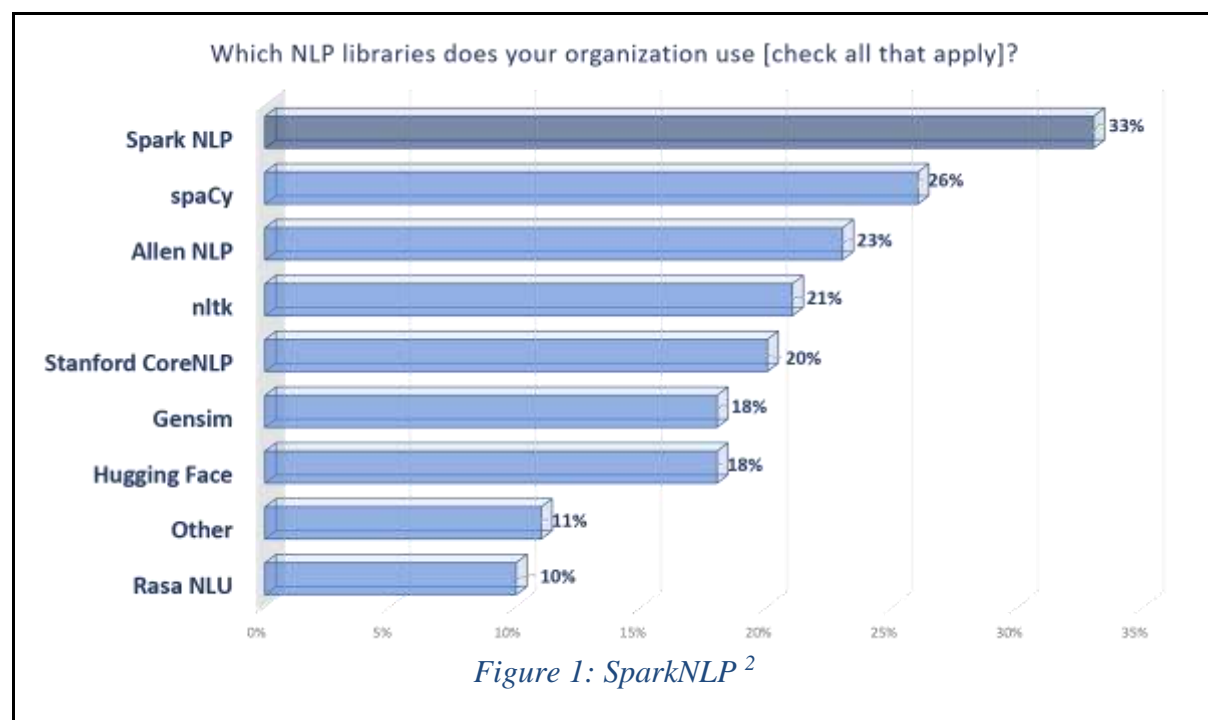
### 3.2.1 BERT

BERT stands for Bidirectional Encoder Representations from Transformers. BERT is a cutting-edge model for Natural Language Processing (NLP) presented by the Google AI team. It is simple but powerful. It is made for the pre-training of deep bidirectional representations from the unlabeled textual data. It does the pre-training of bidirectional representations by jointly conditioning on both the left and the right context in all layers. So, a pre-trained BERT can be fine-tuned using only one additional output layer for developing the state of the art models for a variety of tasks including sentiment analysis, question answer, and more without considerable task-specific architecture changes [7]. BERT exists in a variety of flavors, but we found BERT base (uncased) to be the most applicable to our problem. There are two general BERT versions that have been pre-trained:

The base model is a 24-layer, 1024-hidden, 16-heads, 340M parameter neural network, while the large model is a 12-layer, 768-hidden, 12-heads, 110M parameter neural network [8].

### 3.2.2 Spark-NLP

Spark-NLP is one the most famous libraries for Natural Language Processing (NLP) used by the organizations. It is substantially the most widely used NLP library by Enterprise according to gradientflow.com Figure 1.



<sup>2</sup> <https://nlp.johnsnowlabs.com/>

To calculate the sentiment analysis, Spark-NLP provides a pre-trained model named `sentimentdl_use_twitter`. This model is trained on the Sentiment140 dataset. More information about the model is shown in Figure 2:

<b>Model Name:</b>	<code>sentimentdl_use_twitter</code>
<b>Compatibility:</b>	Spark NLP 2.7.1+
<b>License:</b>	Open Source
<b>Edition:</b>	Official
<b>Input Labels:</b>	<code>[sentence_embeddings]</code>
<b>Output Labels:</b>	<code>[sentiment]</code>
<b>Language:</b>	<code>en</code>
<b>Dependencies:</b>	<code>tfhub_use</code>

*Figure 2: Twitter sentiment analysis pipeline from SparkNLP <sup>3</sup>*

Benchmarking
loss: 7930.071 - acc: 0.80694044 - val_acc: 80.00508 - batches: 16000 <sup>4</sup>

### 3.2.3 GPT-3

Recent research has shown that pre-training on a large corpus of text followed by fine-tuning on a single task can result in significant increase in performance on various NLP tasks and benchmarks. Despite the fact that architecture is often task-agnostic, this strategy still necessitates task-specific fine-tuning datasets of thousands or tens of hundreds of instances. Humans, on the other hand, can usually learn a new language task with just a few examples or simple instructions, something that current NLP systems still struggle with. We show that scaling up language models improves task-agnostic, few-shot performance significantly, and that it can even compete with past state-of-the-art fine-tuning techniques [9].

<sup>3</sup> [https://nlp.johnsnowlabs.com/2021/01/18/sentimentdl\\_use\\_twitter\\_en.html#model-information](https://nlp.johnsnowlabs.com/2021/01/18/sentimentdl_use_twitter_en.html#model-information)

<sup>4</sup> [https://nlp.johnsnowlabs.com/2021/01/18/sentimentdl\\_use\\_twitter\\_en.html#model-information](https://nlp.johnsnowlabs.com/2021/01/18/sentimentdl_use_twitter_en.html#model-information)



GPT-3's developers described how generative pre-training of a language model on a diverse corpus of unlabeled text, followed by discriminative fine-tuning on each unique task ,increased language understanding results in natural language processing (NLP) <sup>5</sup>.

GPT-3 model is made up of multiple transformer model's decoder parts. Figure 3: Architecture of Transformers model shows the complete architecture of transformers model.

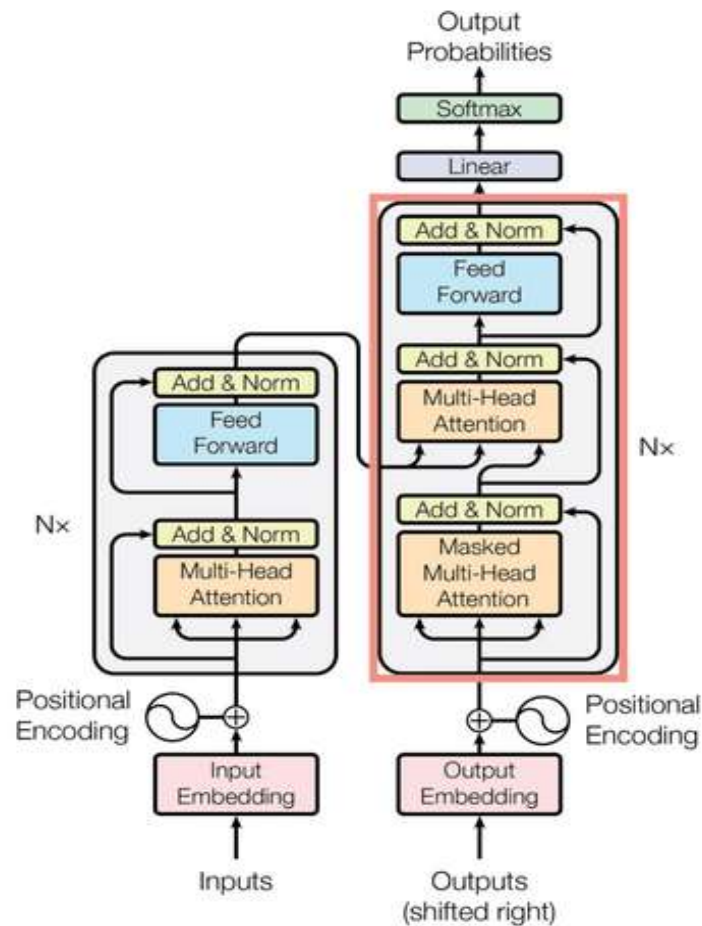


Figure 3: Architecture of Transformers model <sup>6</sup>

Multiple decoders are added in a certain way which makes up the GPT-3 model. The model is trained on a huge dataset from different sources.

GPT-3 Training Data <sup>7</sup>		
Dataset	No of Tokens (billion)	Weight in Training Mix
Common Crwal	410	60%

<sup>5</sup> <https://en.wikipedia.org/wiki/GPT-3>

<sup>6</sup> <https://machinelearningmastery.com/the-transformer-model/>

<sup>7</sup> <https://en.wikipedia.org/wiki/GPT-3>

WebText2	19	22%
Books1	12	8%
Books2	55	8%
Wikipedia	3	3%

### 3.2.4 GPT-2

OpenAI released Generative Pre-trained Transformer 2 (GPT-2) in February 2019 as an open-source language model. It is also comprised of transformer model's decoder parts.

When creating long passages, GPT-2 translates text, answers questions, summarises passages, and provides text output on a level. It can be used for many other language tasks, such as sentiment analysis. According to an article on GPT-2 on wikipedia, "GPT's architecture itself was a twelve-layer decoder-only transformer, using twelve masked self-attention heads, with 64 dimensional states each (for a total of 768). Rather than simple stochastic gradient descent, the Adam optimization algorithm was used; the learning rate was increased linearly from zero over the first 2,000 updates, to a maximum of  $2.5 \times 10^{-4}$ , and annealed to 0 using a cosine schedule"<sup>8</sup>. Due to its size, **CommonCrawl** –which is part of training dataset of GPT-3 model, a big corpus created through web crawling and previously used in training NLP systems, was considered as a dataset for GPT-2, but it was rejected. Instead of scraping content from the World Wide Web at random, OpenAI created a new corpus called **WebText**, which was created by scraping only pages referred to by Reddit posts that had gotten at least three upvotes prior to December 2017. Figure 4: Architecture of GPT-2 shows an abstract architecture of GPT-2.

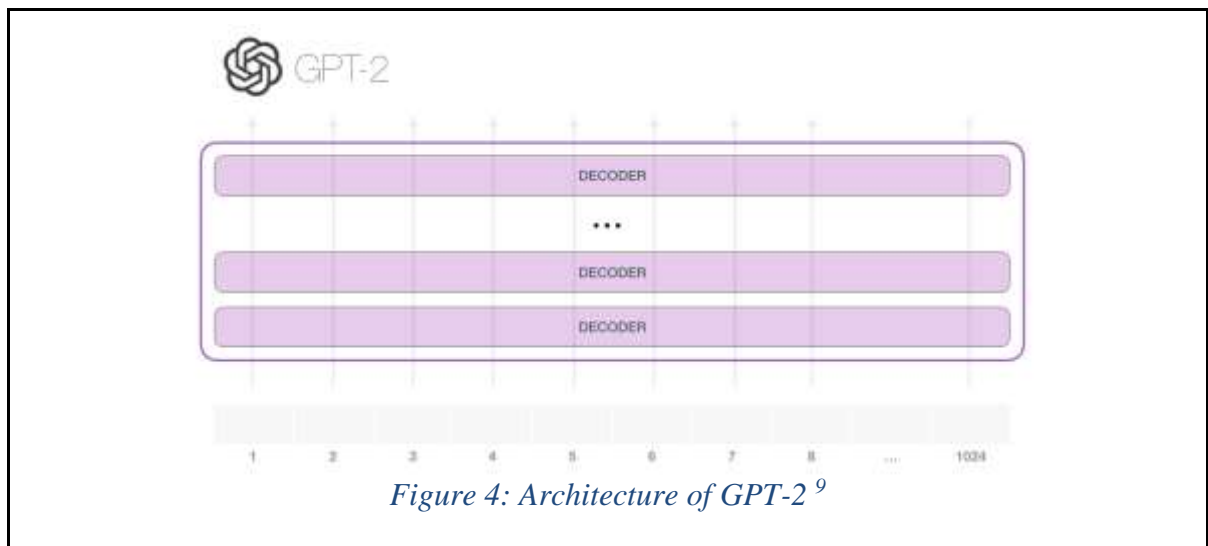


Figure 4: Architecture of GPT-2<sup>9</sup>

<sup>8</sup> <https://en.wikipedia.org/wiki/GPT-2>

<sup>9</sup> <https://jalammar.github.io/illustrated-gpt2/>

## 3.3 Datasets

Following are the four major data sets that we have considered for the sentiment analysis:

### 3.3.1 Sentiment140

The Sentiment140 was not manually created using human annotators, rather it was created automatically using emoticons. The tweets with positive emoticons, like :), were categorized as Positive, and the tweets with negative emotions, like :(, were categorized as Negative [10].

#### 3.3.1.1 Pros:

1. Large data set with 1.6 million tweets.

#### 3.3.1.2 Cons:

1. Automatically created.
2. Outdated: Now, Tweeter uses 240 characters<sup>10</sup> and Sentiment140 consists of tweets of a maximum length of 140.

### 3.3.2 Twitter US Airline Sentiment

The "Twitter US Airline Sentiment" dataset was scraped in 2015 from Twitter and manually curated. The curators were asked to first classify the tweets related to six major US airlines in neutral, positive, and negative tweets, followed by categorising negative reasons (like "bad flight", or "Customer Service Issue", etc).

#### 3.3.2.1 Pros

1. Manually curated.
2. Reasons given with negative categorization.

#### 3.3.2.2 Cons

1. Outdated: Same problem as with Sentiment140.
2. Domain-specific: It contains only tweets related to six airlines of the USA.
3. Unbalanced: It contains 63% negative, 21% positive, and 16% neutral tweets.

### 3.3.3 SemEval

This is from SemEval-2016 Task 4 for the Sentiment Analysis on Twitter.

SemEval is not a balanced dataset; positive tweets are almost twice the neutral tweets and nearly 4 times the number of negative tweets<sup>11</sup>.

---

<sup>10</sup> <https://arxiv.org/abs/2009.07661>

<sup>11</sup> <https://alt.qcri.org/semeval2016/task4>

#### 3.3.3.1 Pros

1. Manually curated.

#### 3.3.3.2 Cons

1. Outdated: Same problem as with Sentiment140.
2. Limited: This dataset contains limited tweets.
3. Unbalanced.

### 3.3.4 Retweet Dataset

This dataset was manually curated using three trained annotators (students). All annotators independently curated the replies of 5,015 tweets without observing the original tweets. Considering the challenging nature of the task, to prevent human errors, the results of the three annotators were correlated and those tweets were finalised in which all the three annotators have the same opinion. Therefore, they ended up with a standard labelled dataset consisting of 1519 labelled tweets [11].

#### 3.3.4.1 Pros

1. Balanced.
2. Manually curated.
3. Up-to-date.

#### 3.3.4.2 Cons

1. Limited: This dataset contains limited data.

## 4 Experimentation

### 4.1 Rule-based Systems

#### 4.1.1 TextBlob

TextBlob is one of the famous and frequently used lexicon-based systems/libraries for sentiment analysis. We have made/developed an interface to interact with TextBlob for sentiment analysis of one single piece of text (tweet or comment) and a batch of tweets (i.e. A CSV file). Textblob uses sentiwordnet dictionary to assign sentiment values to lexicons in a sentence and returns the average sentiment value. Figure 5 shows the confusion matrix of the output of TextBlob on the RETWEET dataset.

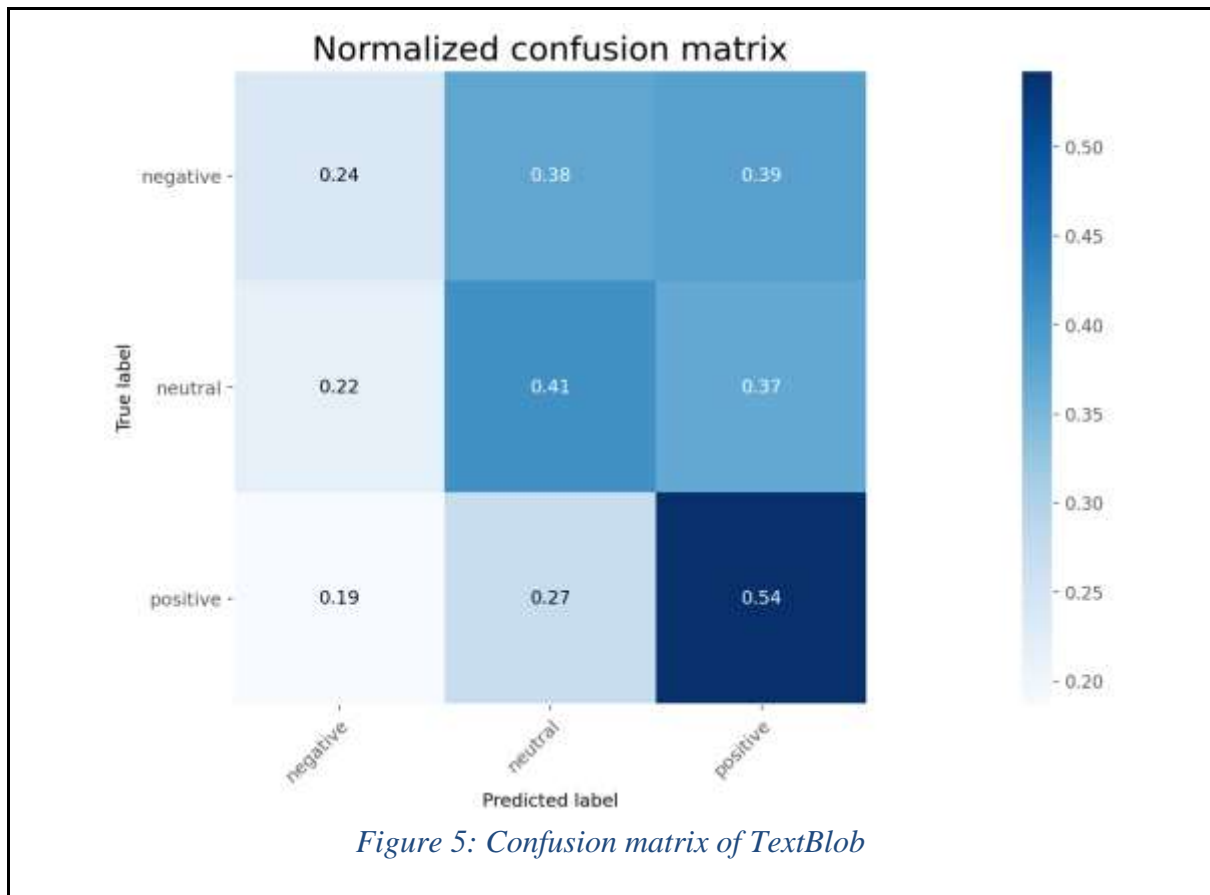


Figure 6 shows the evaluation matrix of the output of TextBlob. The evaluation matrix includes accuracy, precision, and recall.

	precision	recall	f1-score	support
neg	0.42	0.24	0.30	462
neu	0.35	0.41	0.38	359
pos	0.40	0.54	0.46	385
accuracy			0.39	1206
macro avg	0.39	0.40	0.38	1206
weighted avg	0.39	0.39	0.38	1206

Figure 6: Classification Report of Textblob

### 4.1.2 Vader

Vader is one of the lexicon-based sentiment analysis systems which performs best in the case of social media comments' sentiment analysis. We, also, have developed/written a piece of code to perform sentiment analysis on a batch of tweets/comments. Figure 7 shows the confusion matrix of the results from Vader Sentiment Analysis.

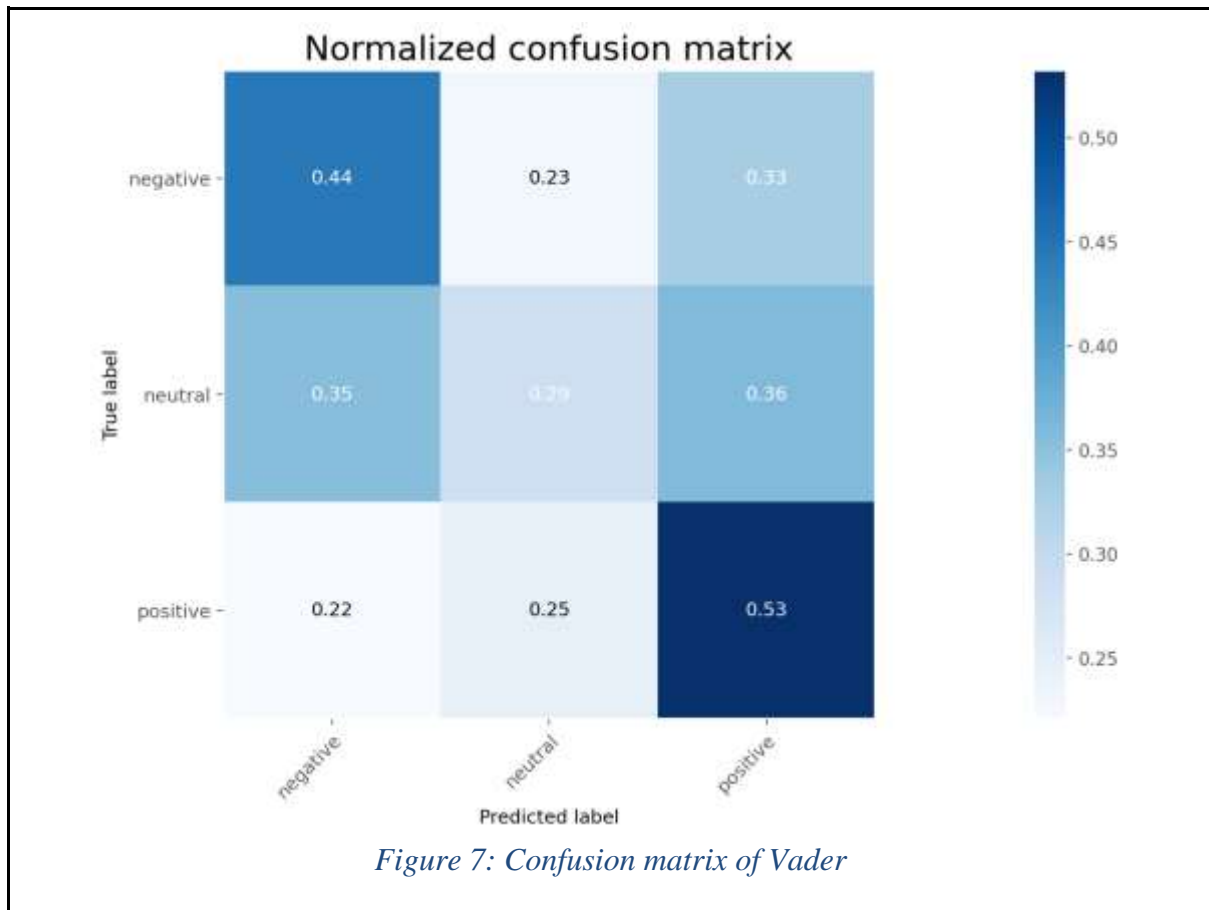


Figure 8 shows the accuracy, precision, and recall of the output of Vader.

	precision	recall	f1-score	support
neg	0.49	0.44	0.47	462
neu	0.34	0.29	0.31	359
pos	0.42	0.53	0.47	385
accuracy			0.43	1206
macro avg	0.42	0.42	0.42	1206
weighted avg	0.42	0.43	0.42	1206

*Figure 8: Classification report of Vader*

## 4.2 Machine-learning Systems

### 4.2.1 SparkNLP's Tweets Sentiment Pipeline

SparkNLP's tweets sentiment analysis pipeline, trained on a large dataset of 1.6M tweets using encodings of Universal Sentence Encoder. We have used the **sentimentdl\_use\_twitter** pipeline from SparkNLP's pool of pre-trained pipelines. It takes a spark data frame and transforms the input data frame by appending sentiment values in the data frame. We have used this pipeline for sentiment prediction/analysis of both single tweets and batch of tweets. The confusion matrix of the output of the pipeline can be seen in the Figure 9.

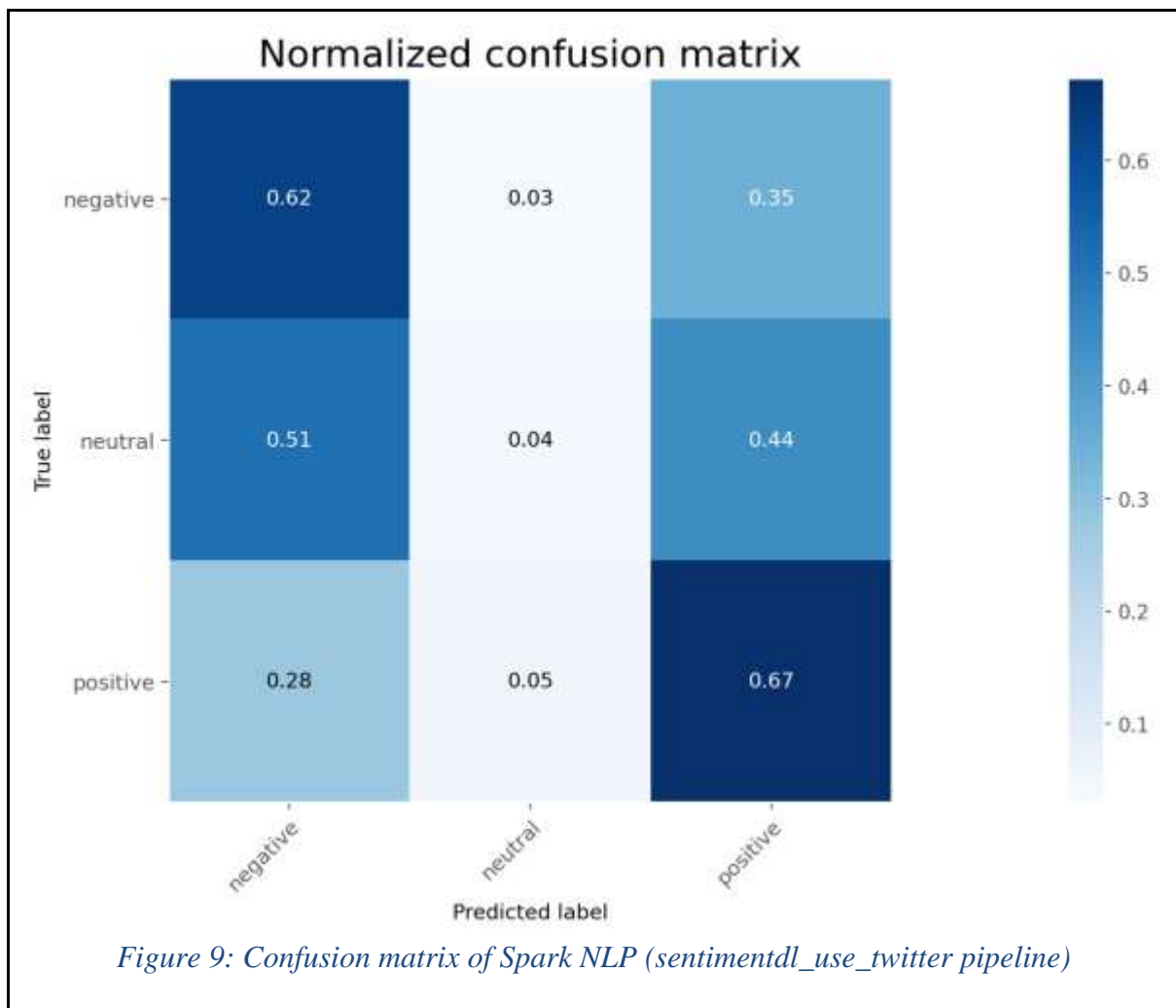


Figure 10 shows an evaluation matrix which includes accuracy, precision and recall.

	precision	recall	f1-score	support
neg	0.50	0.62	0.55	462
neu	0.32	0.04	0.08	359
pos	0.45	0.67	0.54	385
accuracy			0.47	1206
macro avg	0.42	0.45	0.39	1206
weighted avg	0.43	0.47	0.41	1206

Figure 10: Classification report of Spark NLP (sentimentdl\_use\_twitter pipeline)

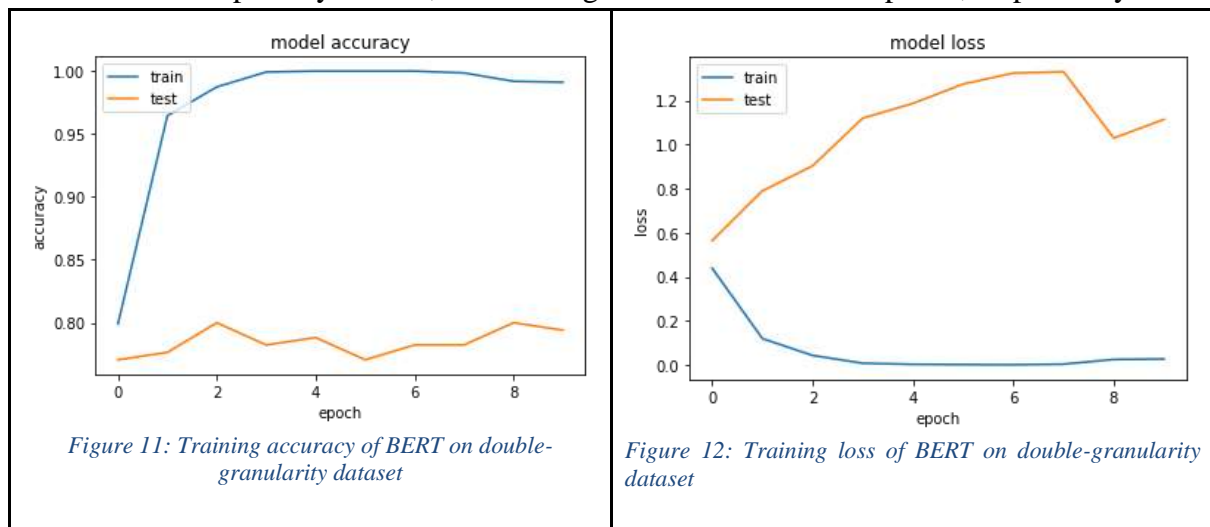


### 4.2.2 BERT For Sequence Classification

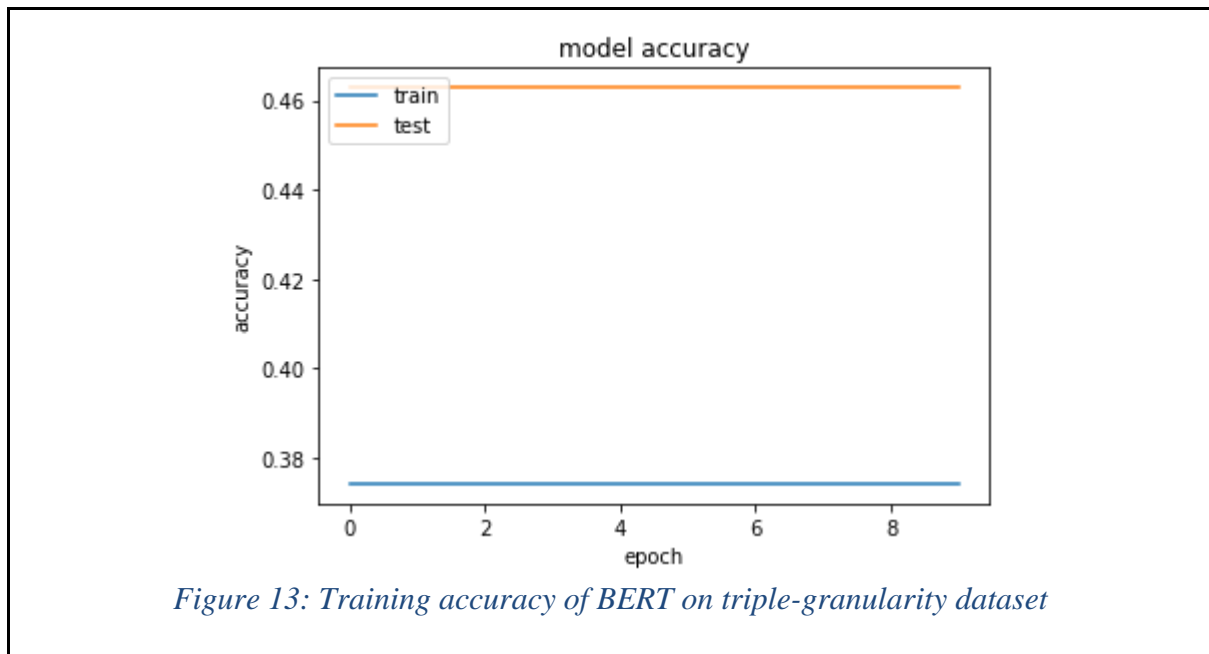
We have fine-tuned Bidirectional Encoder Representations from Transformers aka BERT for sentiment classification. We have used BERT from transformers/hugging-face. We have fine-tuned **BERT\_base\_uncased** on the RETWEET dataset. We have trained BERT on two different datasets, one of double polarity/granularity means binary sentiment values and the other with three polarity/granularity means three sentiment values, positive, negative, and neutral. This model is trained using Tensorflow and is from the TF familt of models from hugging-face. The BERT model is trained with the following settings, in both double and triple polarity datasets:

Parameter	Value
No. of Epochs	10
Optimizer	Adam
Loss Function	Cross Categorical Cross Entropy
Metrics	Sparse Categorical Accuracy

Figure 11 and Figure 12 represents training accuracy from different epochs while training BERT on double polarity dataset, and training loss across different epochs, respectively.



The granularity of the dataset affects the training/learning of the model. While increasing granularity levels of training, accuracy faces a decline. Below is the training accuracy metric of BERT model on a triple granularity dataset (RETWEET dataset) consisting of three sentiment values, positive, negative, and neutral.



Accuracy of the BERT model is very much high as you can see in Figure 13 in case of binary classification of tweets' sentiments. But when it comes to sentiment classification of more than two classes, the accuracy drops by a margin.

#### 4.2.3 GPT-2

GPT-2 aka Generative Pre-trained Transformers-2, is another example of transformers architecture based language models. GPT-2 is trained on a large corpus of data from web pages books and Wikipedia. GPT-2 is an open-source model by OpenAI, available via transformers library/API. We have fine-tuned GPT-2 model to predict the sentiment of a text, which in our case our tweets or their replies. We have fine-tuned GPT-2 model on the RETWEET dataset splitting the dataset into training and validation/testing data. We have trained GPT-2 on triple polarity RETWEET dataset on 10 epochs. Figure 14 shows training accuracy w.r.t. Each epoch. And, Figure 15 shows the confusion matrix accuracy of the GPT-2 model on the three-granularity dataset, respectively.

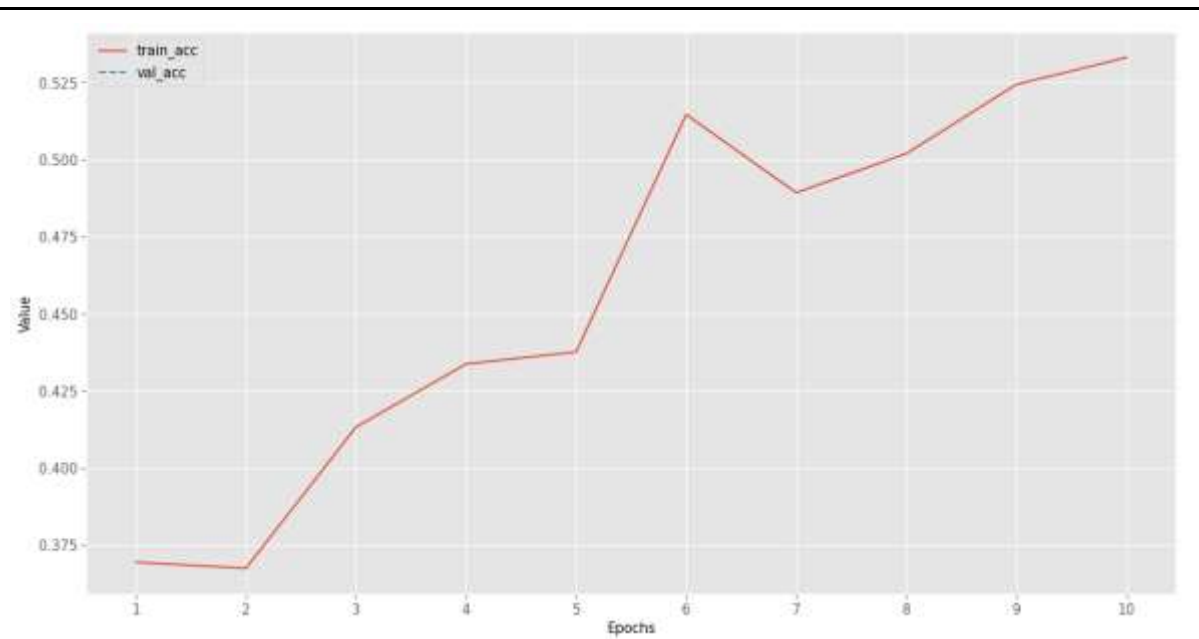


Figure 14: Training accuracy of GPT-2 on three-granularity dataset

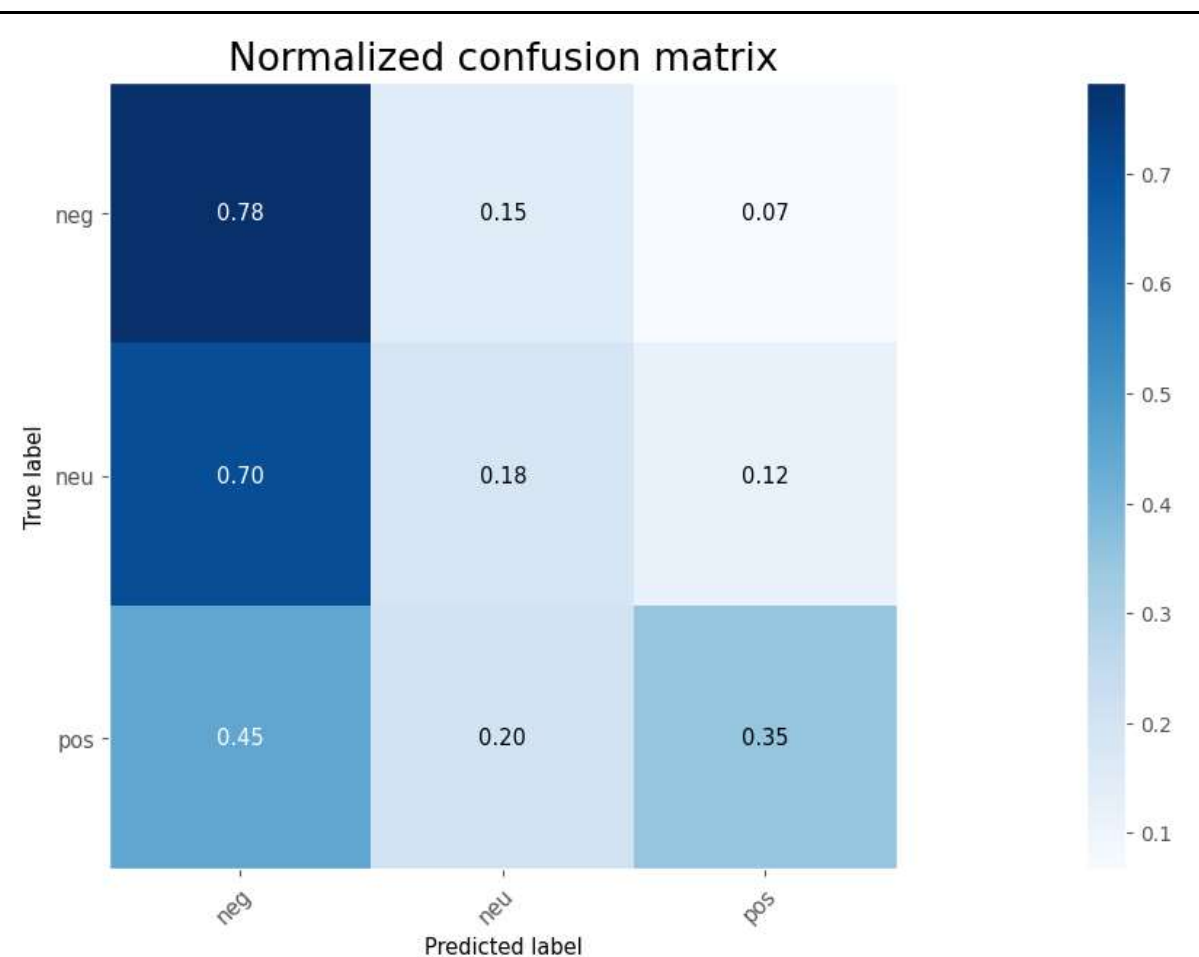
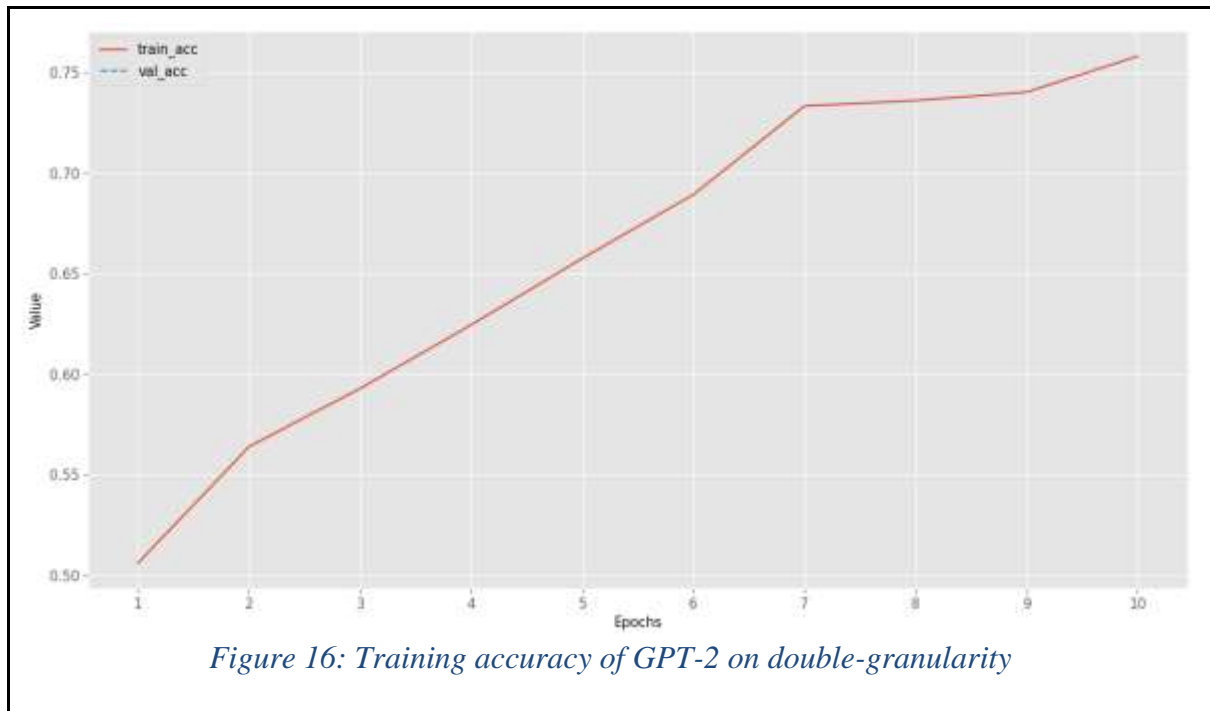
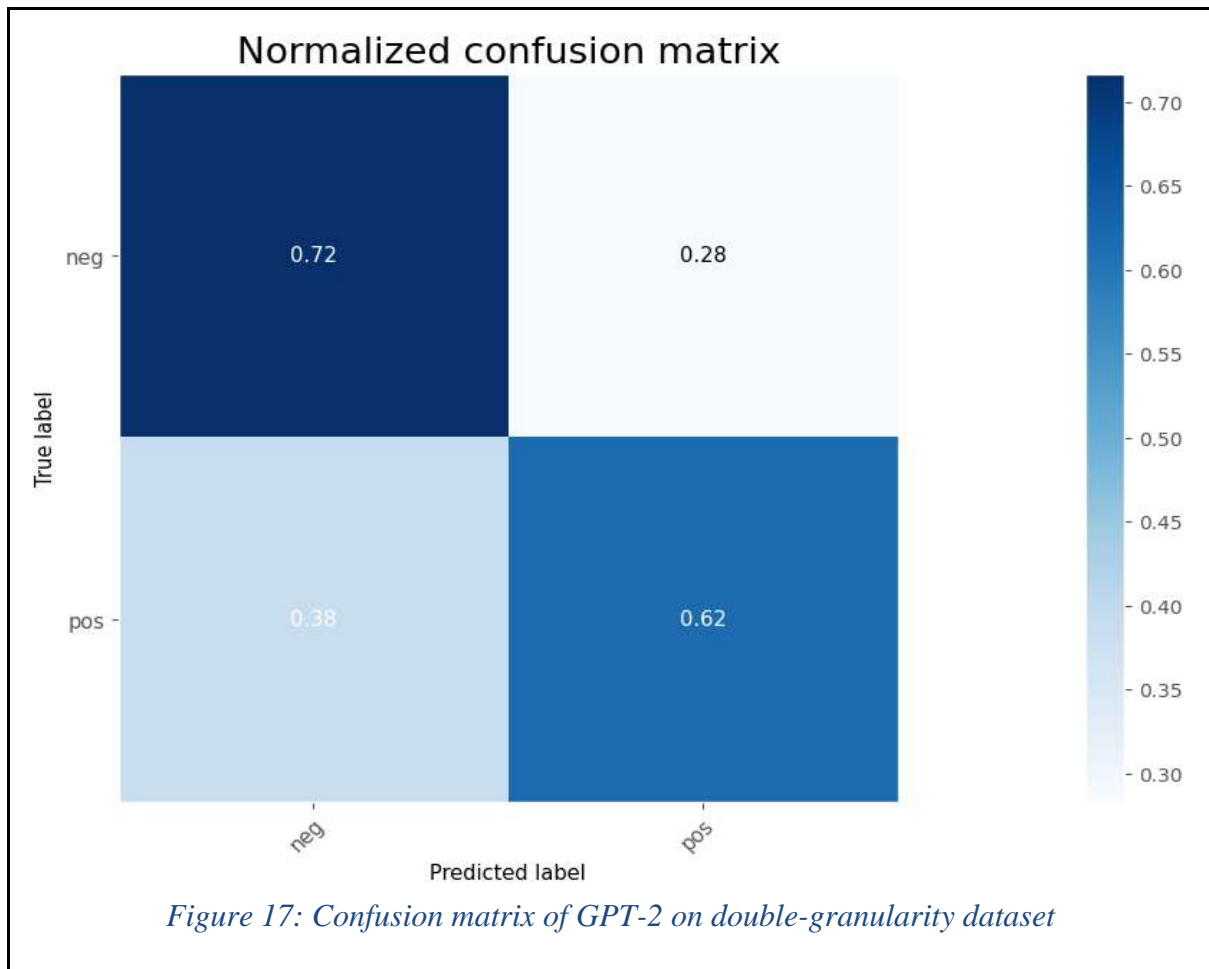


Figure 15: Confusion Matrix upon testing the GPT-2 model (three-granularity).

Figure 16 and Figure 17 show training accuracy and confusion matrix on a double-granularity dataset, respectively.



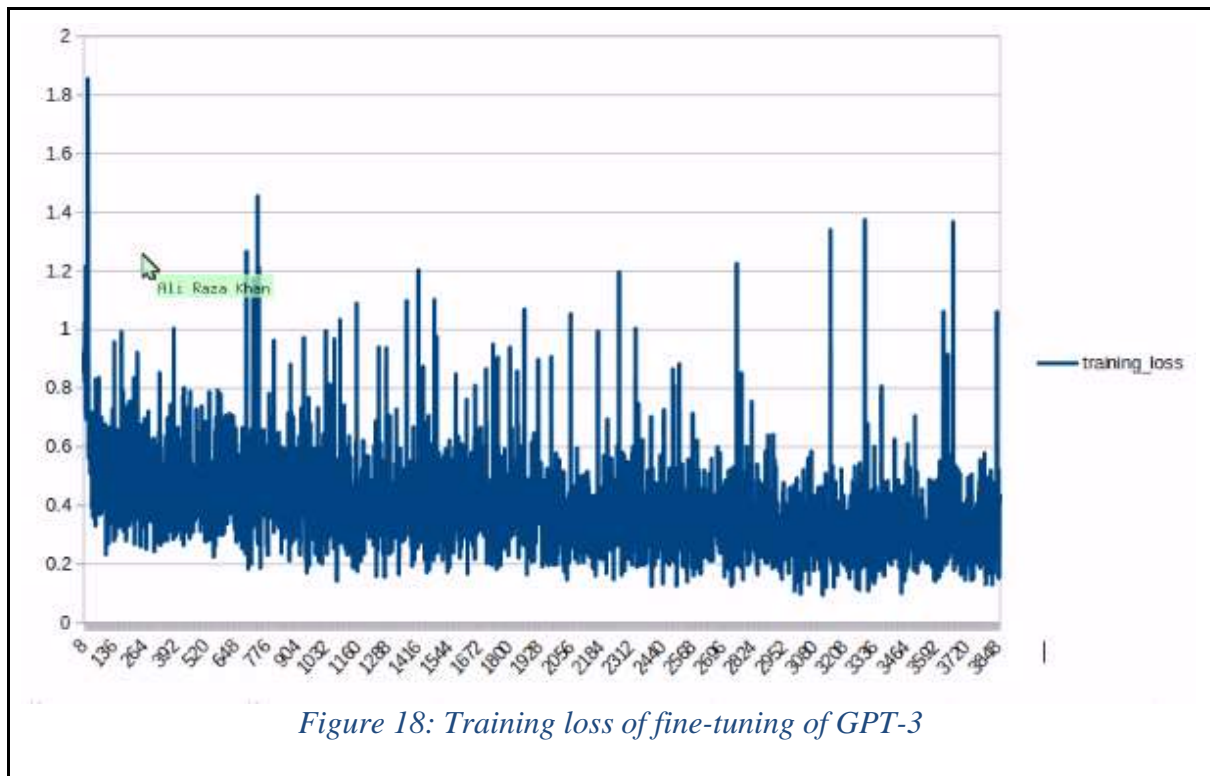


#### 4.2.4 GPT-3

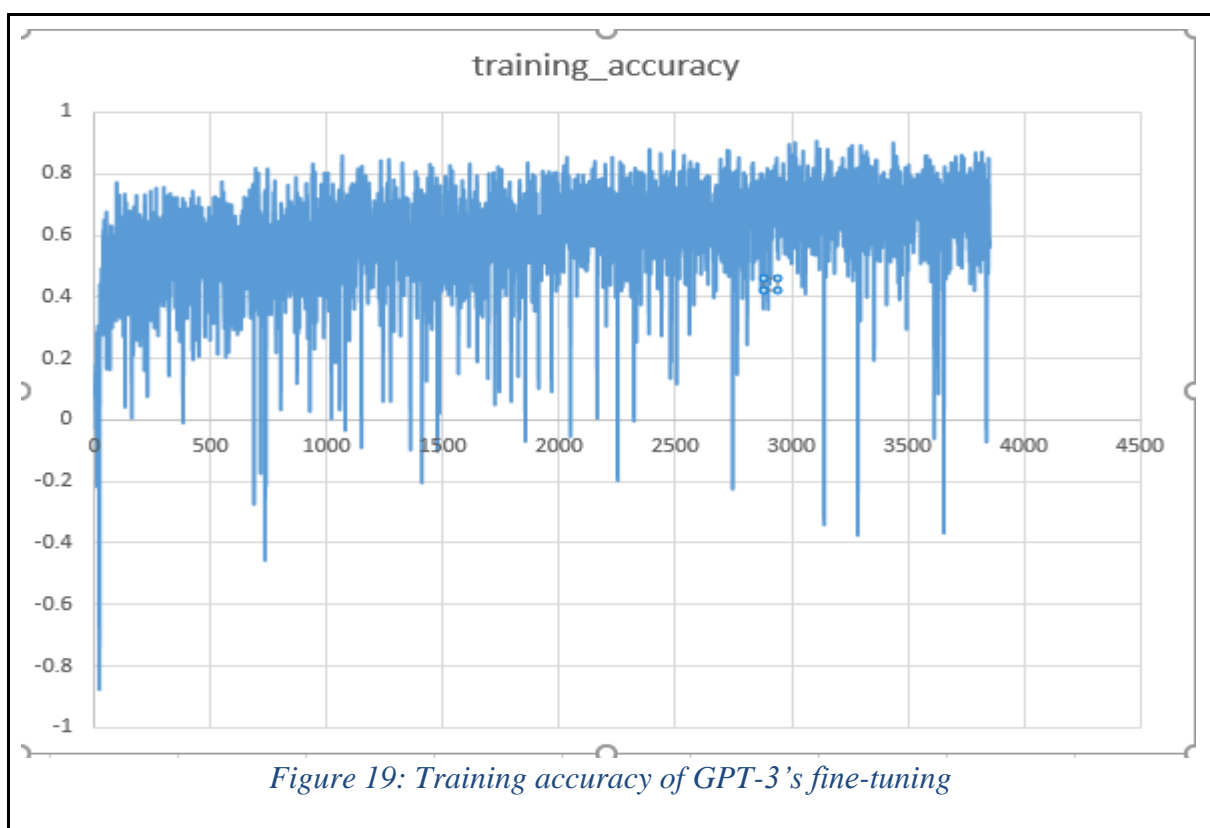
GPT-3 is not yet open-sourced by OpenAI but we can fine-tune the model by providing dataset and settings to OpenAI API command-line interface. GPT-3 operates a particular format in which they, OpenAI, call prompt and completion, a pair of two where prompt is the text and completion is the actual answer of the prompt. We have fine-tuned GPT-3 on a cleaned version of the RETWEET dataset. We have used the **Davinci** engine which is one of the best engines which can perform every task GPT-3 supports. Also, it is the most accurate engine as well as costs more than any other engine, for instance, curie costs 10 times lesser than Davinci which is currently second on the list of GPT-3 engines. We have fine-tuned the model on 4 epochs. After fine-tuning it simply works as the API works, you just give a prompt and it returns a completion which is the sentiment of text/tweet. We have only tested GPT-3 on single sentences because of cost issues with GPT-3, as we have to pay \$0.0600 / 1K tokens and it costs a lot while testing on a batch of tweets or sentences.

Figure 18 shows training loss while fine-tuning GPT-3 model on our, RETWEET, dataset

.



There is not such trend in training loss of the model and shows random behaviour. Similarly let us have a look at the training accuracy of the model. Figure 19 shows the accuracy of GPT-3 model.



Just like training loss, training accuracy also doesn't show any proper behaviour or increasing behaviour

**Note:** The classification reports and confusion matrices given in Fig 5 to 10 are a little outdated because when they were generated, we were not preprocessing the RETWEET dataset correctly. At the time we generated them, VADER was giving 42%, TextBlob 37%, and SparkNLP's model 42% accuracy which was later improved when we started preprocessing the data better and we end up getting 65% accuracy from VADER, 55% from TextBlob and 69% from SparkNLP's model.

#### 4.2.5 Comparative analysis of different sentiment analysis systems

We have performed multiple experiments to see limits and power of all the systems (GPT3, Vader, TextBlob and SparkNLP model) we have studied, in literature review. And their behaviour to different kinds of unseen textual data (comments or sentences). We have used our self developed tool for the real time analysis as well:

**Example-01:**

Text: I love your speech. your choice of words was perfect. While listening to your words, I felt inspired.

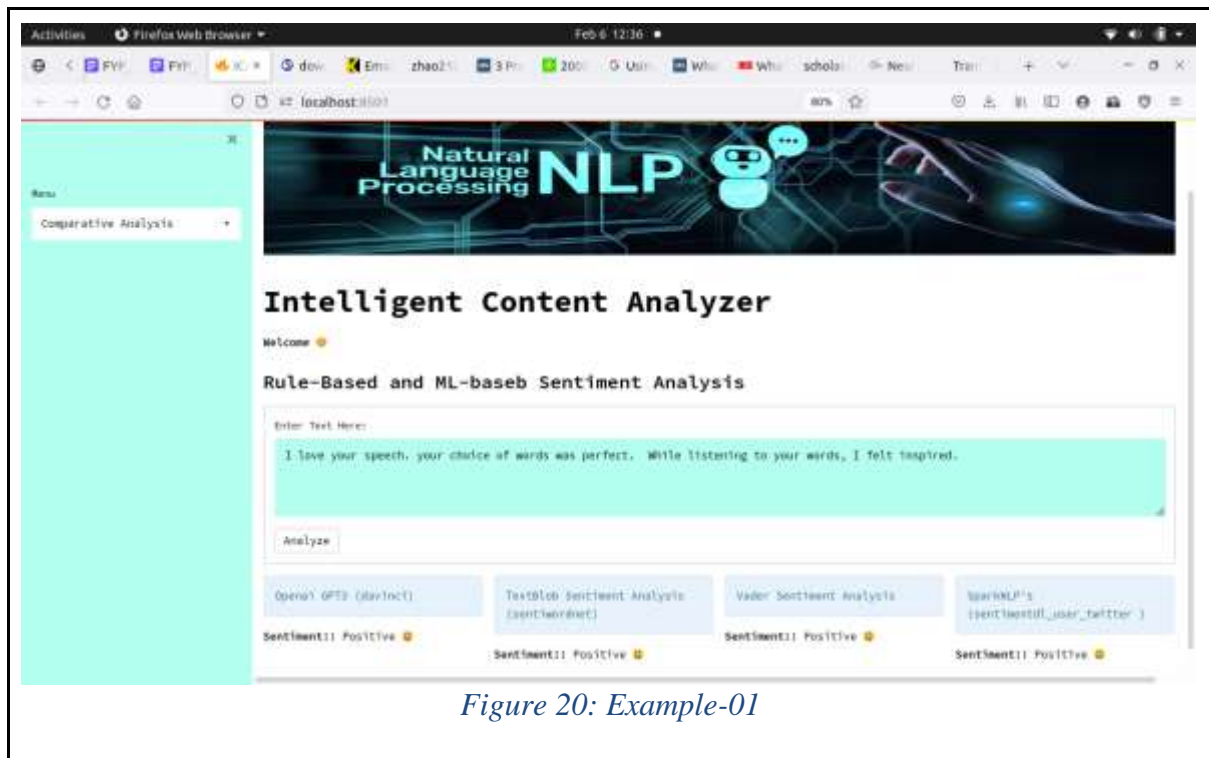
Results:

GPT3: Positive

SparkNLP: Positive

Text Blob: Positive

Vader: Positive



*Figure 20: Example-01*

### **Comments/Observation:**

All systems are giving correct results and categorising the given text as positive.

### **Example-02:**

Text: Farooq is not a good man. He doesn't deserve to be respected because He is a liar.

Results:

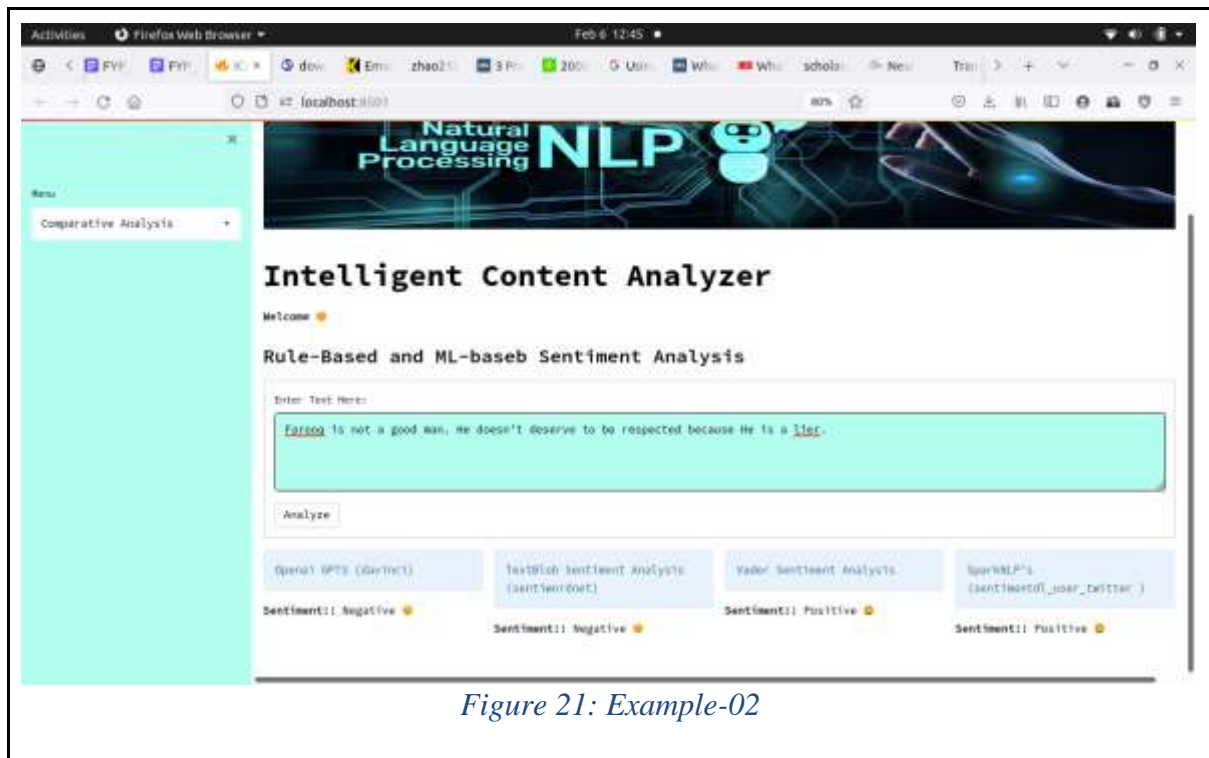
GPT3: Negative

SparkNLP: Positive

Text Blob: Negative

Vader: Positive





*Figure 21: Example-02*

### Comments/Observation:

GPT3 and TextBlob were able to categorise the text correctly and Vader and SparkNLP failed on a straightforward text.

### Example-03:

**Text:** You think you deserve congratulations for this kind of work?

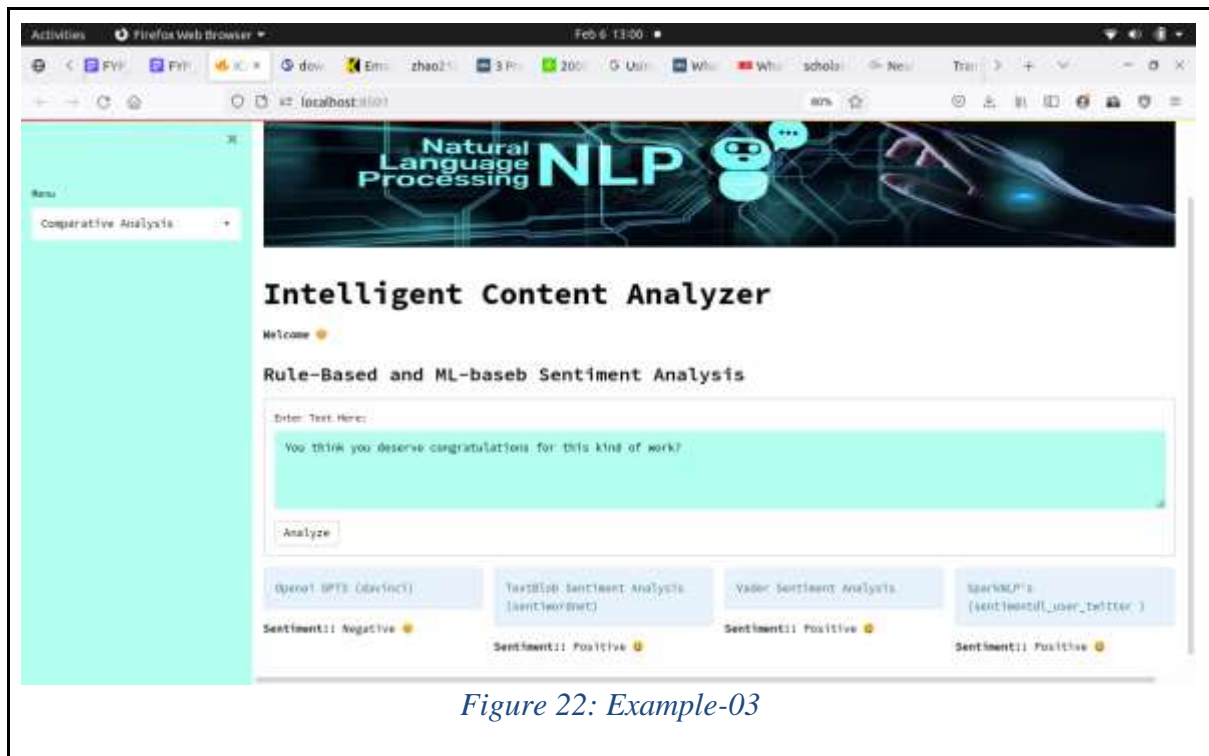
### Results:

GPT3: Negative

SparkNLP: Positive

Text Blob: Positive

Vader: Positive



*Figure 22: Example-03*

#### **Comments/Observation:**

Only GPT3 was able to correctly classify the given straightforward text.

#### **Example-04:**

**Text:** I love killing humans. I feel passionate about killing nice and beautiful ones. I consider this a great service for the human race. I feel great after doing that.

#### **Results:**

GPT3: Neutral

SparkNLP: Positive

Text Blob: Positive

Vader: Positive

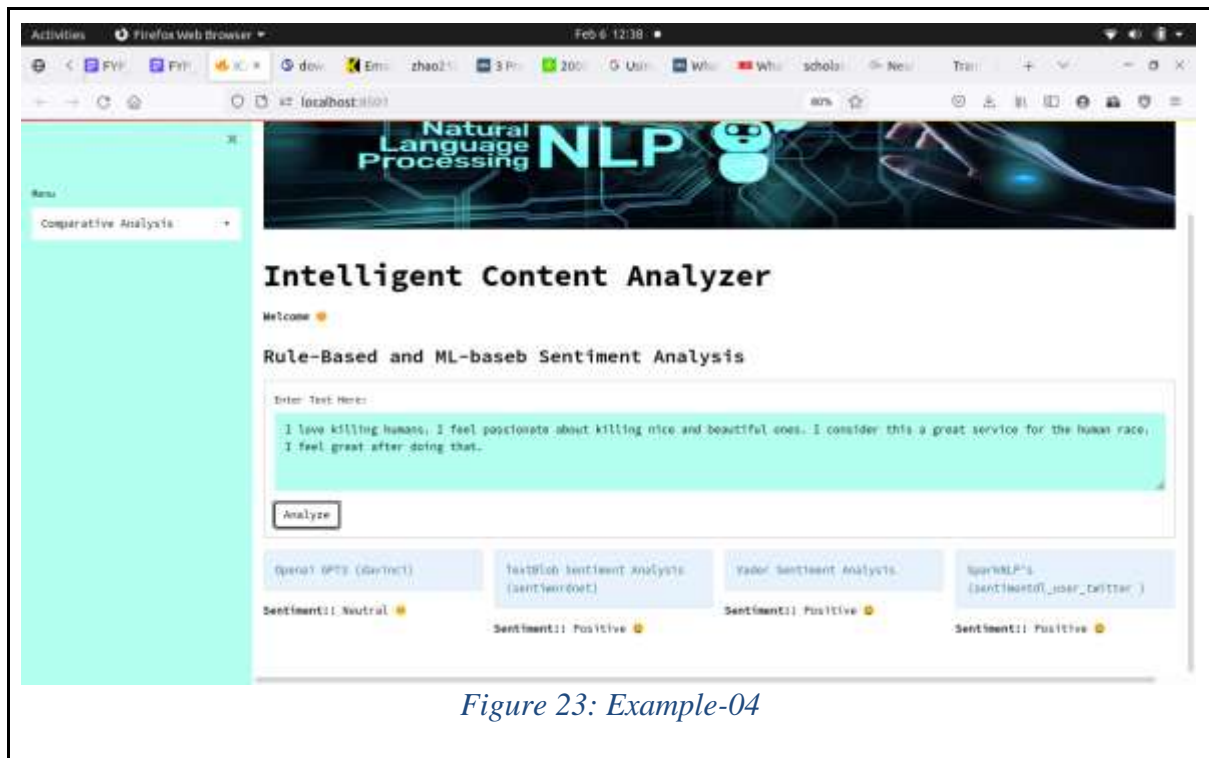


Figure 23: Example-04

#### Comments/Observation:

All systems failed.

## 5 Observation/Results of sentiment analysis systems

From our pool of lexicon-based sentiment analysis systems, Vader outperforms TextBlob as it is specialised on social media data. Rule-based/lexicon-based sentiment analysis approaches perform well on straightforward positive and negative pieces of text unless you give them a sarcastic comment (or a comment that is communicating negative meaning using positive words) to analyse sentiment. Lexicon-based systems do not work on such text/tweet if there is a complex pattern between words of the text. These models also don't deal with words or lexicons they don't have in their dictionary. For example, lexicon-based sentiment analysis systems (VADER, TextBlob) classify "I love killing humans. I feel passionate about killing nice and beautiful ones." as positive text although it is a clearly negative piece of text. But machine-learning-based systems (GPT3, SparkNLP model) for sentiment analysis performs better than lexicon-based systems as they recognize patterns. This is because the machine-learning models learn patterns from the dataset and then apply that learning during testing to make predictions on unseen data.

Although SparkNLP's model is trained on a large Twitter dataset (Sentiment140) with 1.6 million tweets, It still does not perform very well on the tweets. This problem with SparkNLP is because of the poor quality of the dataset it is trained on.

## 5.1 Comparative Analysis

We have observed the flaws in the state-of-the-art Rule-based and ML/Deep Learning based systems using a self developed tool from the above examples. These systems are fragile and fail miserably on the unseen textual data. Rule-based systems fail because they are unable to take into account the context of the words and the complex patterns of human language. Using a list of lexicons is not enough to correctly classify a piece of text for a good sentiment analysis system. We have observed that most popular Deep Learning based systems for sentiment analysis also fail. They fail because they lack quality training data. Remember SparkNLP's model (sentimentdl\_user\_twitter) was trained on Sentiment140 and We have already discussed the problems with this dataset under datasets section.

For Rule-based (Textblob, Vader, etc), We can safely say that they fail because they work on lists of lexicons and fail to take into account the complexities of human languages. Rule-based systems are not perfect and they fail but at least they are explainable and we know why they fail and we can express that in common human understandable terms. On the other hand, for Deep Learning-based systems (GPT3, BERT, etc), We can't say much about why they fail or why they work because they lack explainability. Working of Deep learning based systems can't be expressed in common human understandable terms. These systems have learnability (e.g. In example-03, GPT3 was able to correctly classify the text while all other systems failed because GPT3 knew that the pattern of the language in the text was negative. GPT3 knew that because He has learned that during its training.) but lacked explainability/interpretability. The reason Deep learning-based systems perform better than rule-based systems is that they are learnable. Deep learning systems can learn/remember the hidden patterns in the data. That's why the world is using them extensively in many areas compared to the rule-based systems. On the other hand, Rule-based systems are explainable as mentioned but lack learnability.

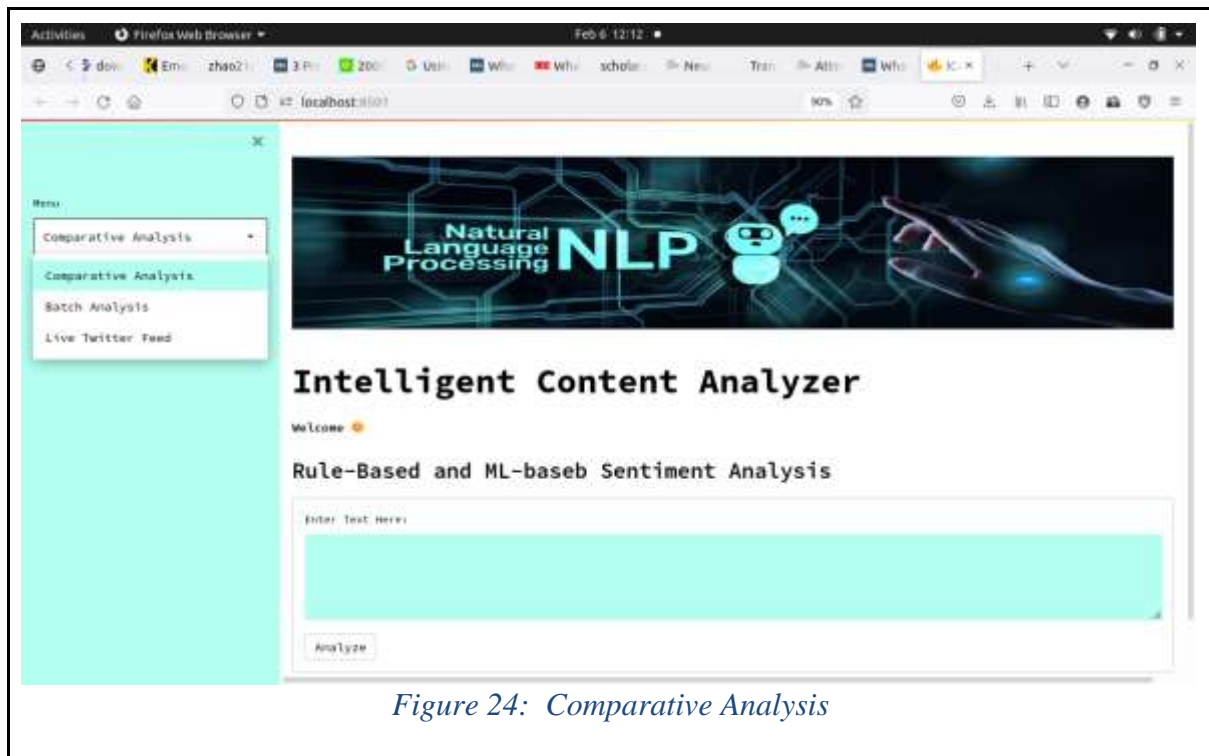
## 5.2 Effect of Granularity of Dataset

The granularity of the dataset has an enormous impact on the performance of language models. We have trained two state-of-art language models(GPT2, BERT) on our dataset(RETWEET dataset). Dataset of three-granularity has low accuracy than the dataset with two sentiment granularity levels. BERT model comes up with an accuracy of 41% on three-granularity levels, but the same model gives more than 95% accuracy in the case of two-granularity levels of sentiment. Even, OpenAI's GPT-2 model has an accuracy of 53% on triple-granularity but accuracy of 75% on double-granularity. So, we have observed and concluded that increasing the number of granularity levels in sentiment in a dataset directly affects the accuracy of a model. We think this problem with the text sentiment classification is due to the increasing number of boundaries because, with the increase in granularity levels of the dataset, a model has to learn more boundaries.

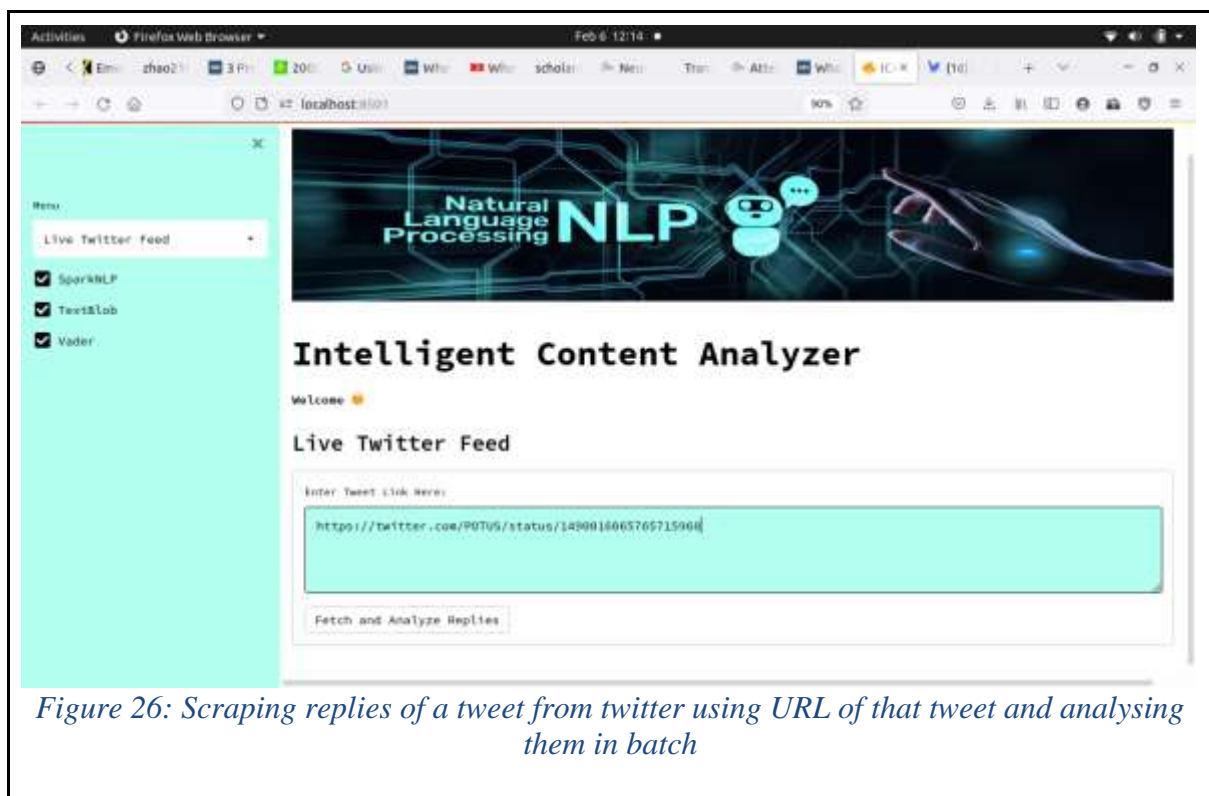
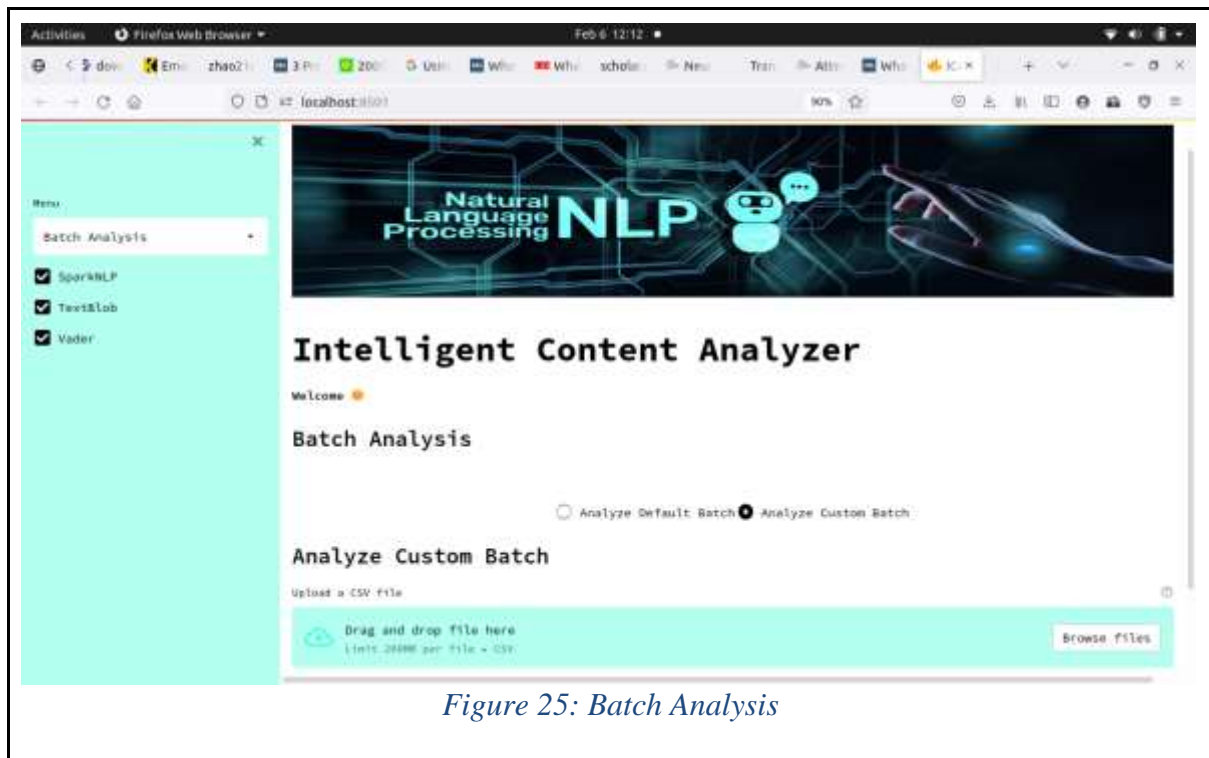
## 6 Implementation

### 6.1 Comparative Analysis Tool

Following are the details about the tool we have developed to perform our comparative analysis in real-time:



*Figure 24: Comparative Analysis*



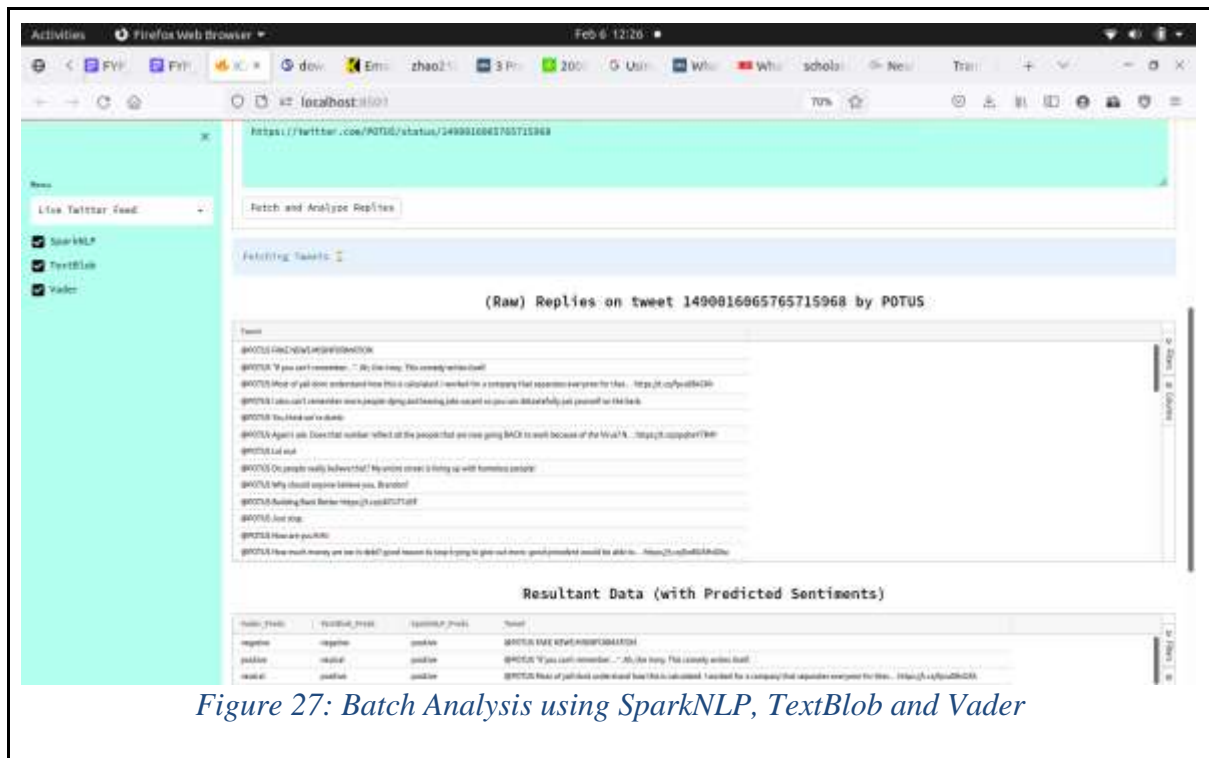


Figure 27: Batch Analysis using SparkNLP, TextBlob and Vader

**Note:** We haven't added OpenAI GPT3 here in batch analysis due to the usage limitation as they only allowed us to use it in a limited manner.

## 7 Deep AI Algorithm:

### 7.1 Logical Analysis:

Logical Analysis is basically the classification of text based on logic. Logic deals with the text that is in the form of an argument and classifies whether an argument is valid or invalid. This is the test for the structure of the argument.

### 7.2 Why Logical Analysis?

We want to perform logical analysis to filter out the illogical text and give the end user the option to see negative but logical comments as well. We wanted to go one step further from the sentiment analysis to introduce a more richer measure. A lot of work has been done on sentiment analysis but there is no work done on logical analysis of the text.

### 7.3 Argument:

The set of initial statements (premises) followed by conclusion is called an argument.



Example:

**If Elon Musk buys Twitter, then Twitter will become more transparent. Elon Musk has bought Twitter. Therefore, Twitter has become more transparent.**

1<sup>st</sup> Premise: **Orange**

2<sup>nd</sup> Premise: **Green**

Conclusion: **Red**

## 7.4 Argument's Validity:

An argument is valid if and only if in every case where all the premises are true, the conclusion is true. Otherwise, the argument is invalid <sup>12</sup>.

## 7.5 Argument Classification Model

The initial idea was to use a hybrid approach (ML based + Rule based) for logical classification. It was our hypothesis that classifying a text is in the form of an argument or not would help us in the future to develop the Deep AI Model for logical analysis because it is necessary for a text to be in the form of an argument before the logical analysis. That's why we have performed multiple experiments to develop a good approach.

### 7.5.1 Dataset:

We have used a self-curated dataset for argument classification model. Our dataset consists of 200 labeled retweets. It is a completely balanced dataset with 100 retweets that are in the form of an argument and 100 retweets that are not in the form of an argument.

### 7.5.2 Bag of Words + SVM:

In our 1<sup>st</sup> experiment, we used the classical bag of words representation and trained and SVM on our self-curated dataset. We achieved 63% accuracy.

### 7.5.3 Bert Representation + SVM:

In our 2<sup>nd</sup> experiment, we used the Bert Representation and trained and SVM on our self-curated dataset. It was our hypothesis that Bert Representation will improve our results, which is exactly what happened, and we achieved 83% accuracy.

---

<sup>12</sup>[https://math.libretexts.org/Courses/Monroe\\_Community\\_College/MTH\\_220\\_Discrete\\_Math/2%3A\\_Logic/2.6\\_Arguments\\_and\\_Rules\\_of\\_Inference](https://math.libretexts.org/Courses/Monroe_Community_College/MTH_220_Discrete_Math/2%3A_Logic/2.6_Arguments_and_Rules_of_Inference)



## 7.6 Order Sentence Model:

As we know that, a typical argument consists of a number of premises followed by a conclusion. The human language is complex and it is not fixed that the premises will necessarily be coming before the conclusion. Therefore, we wanted to know what happens if there is a change in the order means conclusion in coming before the premises or the in the middle of the sentence instead of coming at the end.

### 7.6.1 Experimentation:

We have used BERT as end-to-end classifier in this experiment. We used our previous self-curated dataset for this experiment. By training BERT with default order of sentences, we achieved 88% accuracy while by training BERT with shuffled order of sentences, we achieved 91% accuracy. These results indicate that order did not matter much which is a bit surprising for us because argument is all about structure. Therefore, We believe more experiments needs to be done to understand the reason of these results.

## 7.7 Logical Classification:

In the world of logic, there are few very common forms of valid and invalid forms of arguments. The ones that we are using are given below:

Valid Forms	
<b>Modus Ponens</b> $p \rightarrow q$ $p$ $\therefore q$	<b>Modus Tollens</b> $p \rightarrow q$ $\sim q$ $\therefore \sim p$
Invalid Forms (Fallacies)	
<b>Inverse Error</b> $p \rightarrow q$ $\sim p$ $\therefore \sim q$	<b>Converse Error</b> $p \rightarrow q$ $q$ $\therefore p$

Examples:

If the new coalition government fails to run the state, then elections need to be held. The new coalition government has failed to run the state. Therefore, elections need to be held. (**Modus Ponens**)

If you interfere in the internal affairs of my country, I will not support you. I am supporting you. Therefore, You have not interfered in the internet affairs of my country. (**Modus Tollens**)  
If I help you, I will not be able to vote. I did not help you. Therefore, I was able to vote. (**Inverse Error**)

If I do not vote for you, I will be helping Biden. I have helped Biden. Therefore, I have not voted for you. (**Converse Error**)

## 7.8 What is Deep AI Model?

It is an agent-based human in the loop model for classifying a text as logical or illogical. Following are the basic ingredients of Deep AI Model.

- It consists of 5 agents (2 logically valid agents based on the above logically valid forms of argument, 2 logically invalid agents based on the above logically invalid forms of argument and 1 human agent).
- The 4 agents that are based on various forms of arguments have their respective localized memories. Memory is made up of two elements, (a) a set of representative arguments of their specific type, and (b) against each argument in the memory of a agent, an array of distribution which for agent 1 is initialized by the first index being set 1, and the rest of three indices having the value 0. This array represents the “belief strength” of an agent for a specific argument in its memory, as well as representing an implicit footprint for all its interactions with other agents. For agent-2, the second index will have 1 and all other indices will have a value of 0. See Table 1 and Table 2.
- The 5<sup>th</sup> agent is the human agent. It provides ground and resolves the contended cases. It is assumed that human agents have sufficient knowledge about the valid and invalid forms of argument to give a verdict in favor of any agent.
- A round of contest is played among the agents when a new argument is presented as input to the system as per the algorithm presented in Figure 29, where each agent plays by matching the input argument with those arguments that are stored in its localized memory. If a match is found then agent casts a vote. The human agent also votes in some specific conditions.
- The agents that participate in the voting for the input argument update their belief strength arrays against their matched arguments stored in their respective localized memories.

*Table 1: Agent-1's (Modus Ponens) initial memory*

ID	Comment	1	2	3	4
0	We know that if Trump uses a prevent defence strategy during the super bowl week then He will fail. He is using the prevent defence strategy. Therefore, He will fail.	1	0	0	0
1	If the new coalition government fails to run the state, then elections need to be held. The new coalition government has failed to run the state. Therefore, elections need to be held.	1	0	0	0
2	If Elon Musk buys Twitter, then Twitter will become more transparent. Elon Musk has bought Twitter. Therefore, Twitter has become more transparent.	1	0	0	0
...	...	1	0	0	0

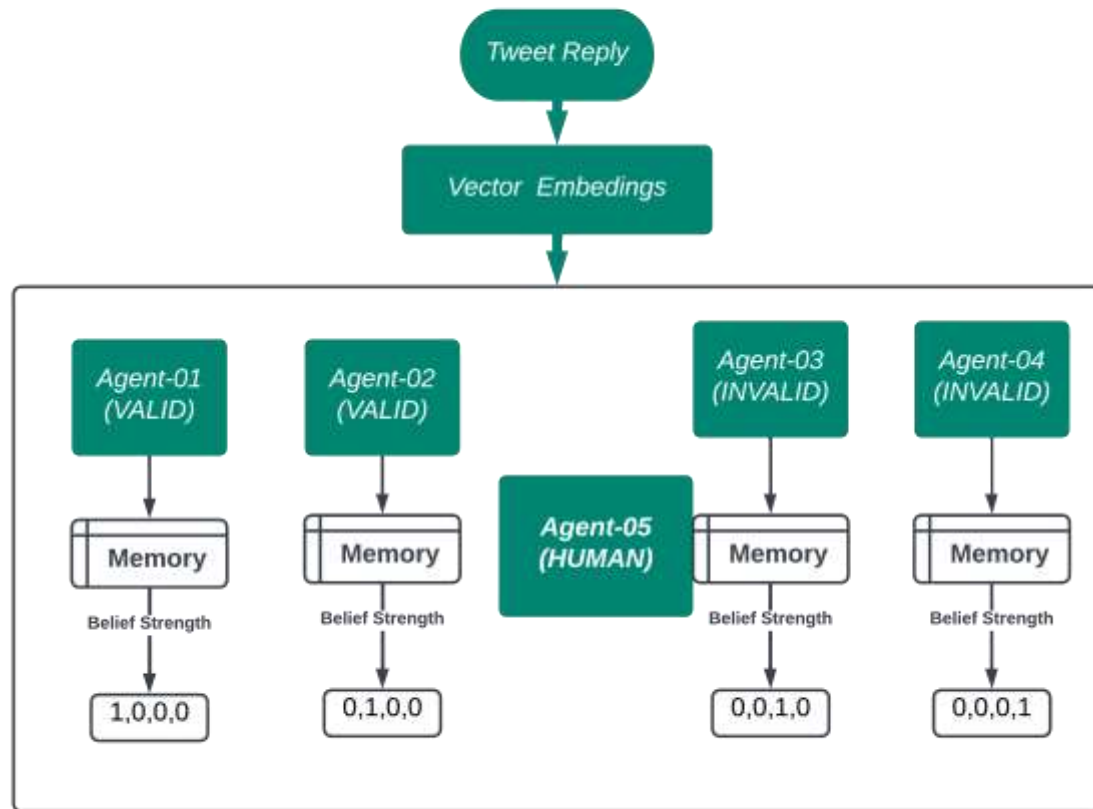
*Table 2: Agent-2's (Modus Tollens) initial memory*

ID	Comment	1	2	3	4
0	If you interfere in the internal affairs of my country, I will not support you. I am supporting you. Therefore, You have not interfered in the internet affairs of my country.	0	1	0	0
1	If you play poorly, you will not win the game. You have won the game. Therefore, you have not played poorly	0	1	0	0
2	If I go on vacation, I will not perform well in exams. I have performed well in exams. Therefore, I have not gone on vacation.	0	1	0	0
...	...	0	1	0	0

### 7.8.1 Visual Representation:

The following figure pictorially represents our human-centered Deep AI Model:

## Human-centered agent-based Deep AI Model



*Figure 28: Pictorial View of our Deep-AI Model*

### 7.8.2 Algorithm:

These following instructions are for each agent-x, where x represents the agent ID and also the logical form:

In short, each time a new argument is given as input to our system then that new argument is matched with the respective arguments already stored in each agent's memory. If no match by any agent then the input argument is passed on to the human agent, and e.g, the human agent decides that the given input argument is in the form of Modus Ponens and belongs to Agent-1, then this is the final output, and this current argument, and its associated belief strength array (1,0,0,0) gets added to agent 1's localized memory. If only one agent is able to find a matching argument, then that agent's ID (form of the argument) would be the final output of the system. (A particular Argument form) value for the input argument. E.g., it was agent 2 that found a matching argument in its localized memory and its matched argument had the current belief strength array of (1,4,0,0) then it would be strengthened to (1,5,0,0). Now, against this specific matching argument agent 2's belief strength value is 5 for future references unless it further gets evolved. In case if multiple agents find a match in their respective memories, then there are two possibilities:

1. If there is one agent amongst the contending agents which has a higher belief strength value then the case gets settled and this agent wins.
2. Or if there are multiple competing agents with the same belief strength values. In this situation, the human agent interferes and decides the fate of the input argument.

Either way, whichever agent wins, all the contending agents update their corresponding matched argument's belief strength arrays at the winning agent's index. The Deep AI algorithm is given below:

**Algorithm 1: Human-centered Agent-based Deep AI Algorithm**

```

if I am an agent-(x), on my turn, I find the best match between the input comment and the
comments within my memory then
  if no match found, pass and wait here till the end of round then
    if output is x, add the comment to my memory, with appropriate initial belief
    strength array, with all 0s, except for index x having a value of 1;;
    else if output is not x, ignore;
  else
    if match found, I propose the following, and wait here till the end of round:
      • my ID as output x
      • my vote's explanation (matching strength)
      • my vote's belief strength

    if contention by one or more agent then
      If there is perfect matching or a single maximum value for competing belief
      strengths, that agent's ID is the output ;
      In case of a tie for maximum belief strength values, human judge is invoked;
    else
      no contention by any other agent, x is the winning output
    end
    I update my belief strength array at winning agent's index, against the matched
    comment in my memory
  end
else
  I am human agent, I get to decide in case no agent outputs, or in case of competing
  belief strength values;;
end

```

*Figure 29: Algorithm for our Deep-AI Model to classify logic of a statement/tweet*

**Note:** The above algorithm is inspired from a work titled, “An interactive multi-agent reasoning model for sentiment analysis: a case for computational semiotics” [12].

### 7.8.3 Key Aspects of the System:

Some key aspects of the Deep AI Algorithm are given below:

1. The system learns when new argument is added in the memory and when contending agents update their respective belief strength arrays

2. It is hoped that with the right conditions, the human agent's interference reduces with the passage of time, as the agents's own and shared experiences becomes more reasonable and richer. We tested the Deep AI Model on 100 unseen data instances (real tweets). And, following is the trend of learning:

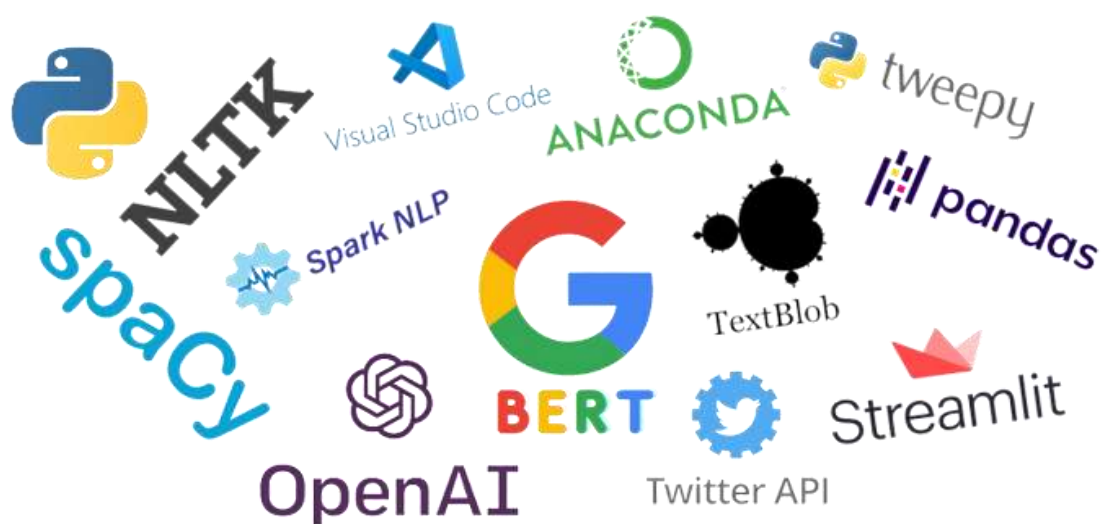
**$2 \rightarrow 5 \rightarrow 7 \rightarrow 17 \rightarrow 26 \rightarrow 33 \rightarrow 35 \rightarrow 47 \rightarrow 83$**

From our initial testing, We have observed the above trend which shows that the involvement of the human agent reduces over time. The human agent gets involved on the 2<sup>nd</sup> tweet, then on 5<sup>th</sup> tweet, ... then 83<sup>rd</sup>.

3. This system is explainable and we can explain the working of this system in human understandable terms.
4. The system can work with small or no data.
5. The system is human-centered.

## 6.2 Major Tools and Technologies

- Python3
- Streamlit
- Spark-NLP
- NLTK
- VADER
- spaCy
- TextBlob
- OpenAI CLI & API
- Tweepy
- Twitter API
- Pandas
- Anaconda



*Figure 30: Tools and technologies used*

## 7 Future Plan

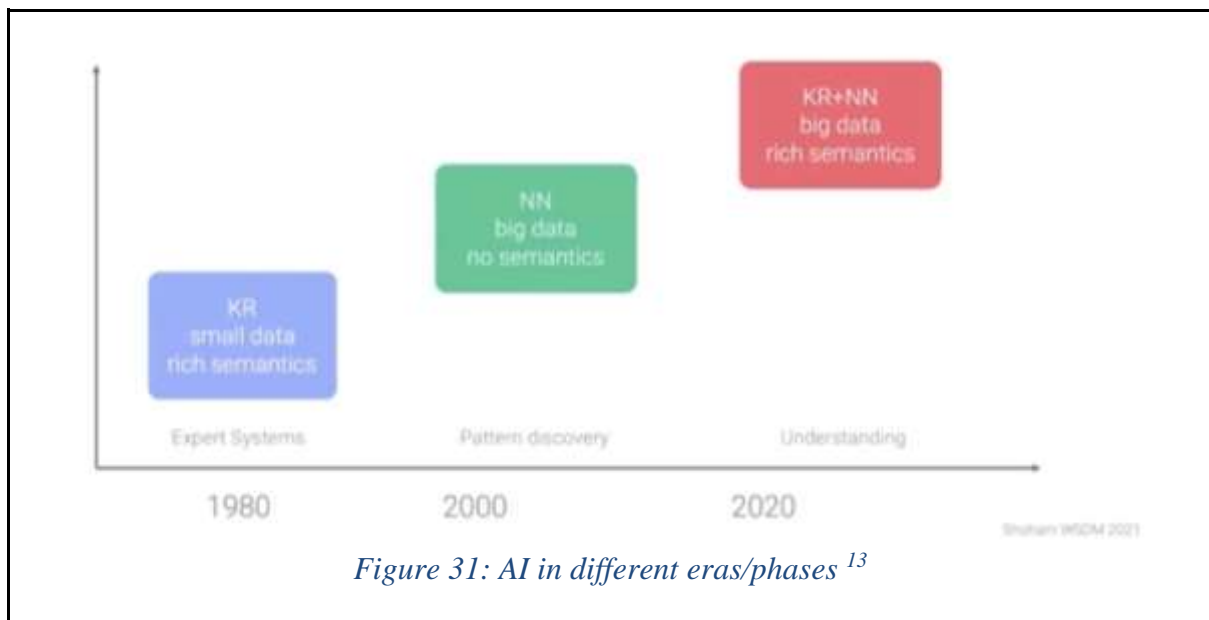
### 7.1 Challenges in AI

The computer scientist and a Professor Emeritus at Stanford University, Yoav Shoham in his keynote address at 14th ACM Conference on Web Search and Data Mining has stressed that Deep networks are not intelligent. They are dumb, and They lack the common sense of even the 5-year-old. There is a need to develop technology to make machines understand the language like humans. Big models like Bert, T5 become highly specialized to the datasets they trained on [13].

We have also observed the problems highlighted by Yoav Shoham during our analysis of state-of-the-art NLP systems.

### 7.2 Future of AI

According to Yoav Shoham, To deal with the above challenge, Various efforts to merge the symbolic reasoning in deep networks are going on, and this is the future. He has bet that We are entering the third phase of AI which is expressed in the following figure:



**Note:** In the above figure, The first phase of AI ends around 2000. The second phase ends around 2020, and the third phase of AI starts in 2020.

<sup>13</sup> <https://www.ai21.com/blog/nlp-21st-century>

## 7.3 Deep AI Model's Development

The Deep AI Model is developed considering the challenges and future opportunities highlighted above and also highlighted by [14] and in [15]. Our Deep AI Model is human-centered. It is reasonable, explainable and learnable system.

## 7.4 Envisioned Product

The Deep AI Model can be embedded in a product (Intelligent Content Analyzer) in future after doing rigorous testing. The envisioned product will provide its users with the tools to have more control over their social media lives by intelligently analyzing the content of replies on the social media posts of its users.

### 7.4.1 Tentative Interface

The tentative interface of the envisioned product is given in the Figure 32.



*Figure 32: Prototype of our final product*



## 7.4.2 Tentative Business Plan:

The tentative business plan of the envisioned product is given below:

### 7.4.2.1 Business Model

Our model will be a subscription-based business model.

### 7.4.2.2 Competitors

There are no direct or indirect competitors.

### 7.4.2.3 Proposed customers

The regular twitter users in general and especially journalists, celebrities, politicians and public figures.

### 7.4.2.4 Advertising and promotions strategy

We will be utilizing “Google Ads” services and also use other social media platforms for online presence and marketing.

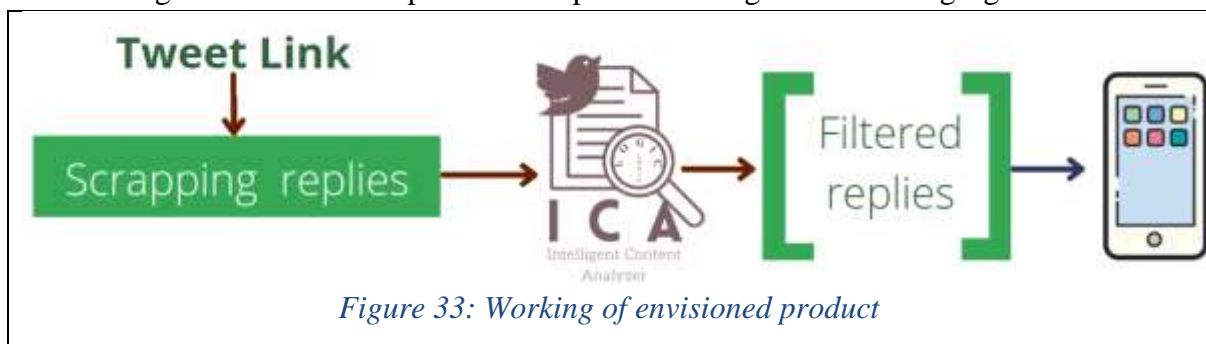
### 7.4.2.5 Pricing strategy

After a one week trial period, We will be providing two pricing plans:

- 1. Premium:** This will be for individual users and it will costs \$15 per month.
- 2. Business:** This will be for professionals and teams of 5 to 150 users and it will costs \$25 per month.

## 7.4.3 Working

The working of the envisioned product is expressed through the following figure:



## 7.5 Testing

The initial testing of Deep AI Model has showed good results but more testing is required to improve it and make it deployable.

## References

- [1] S. Loria, "textblob Documentation," *Release 0.16.0*, vol. 0, 2021.
- [2] C. Musto, G. Semeraro and M. Polignano, "A Comparison of Lexicon-based Approaches for Sentiment Analysis of Microblog Posts.," in *DART@ AI\* IA*, 2014.
- [3] A. Athar, "Sentiment Analysis: VADER or TextBlob?," January 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/01/sentiment-analysis-vader-or-textblob/>.
- [4] V. Bonta and N. K. N. Janardhan, "A comprehensive study on lexicon based approaches for sentiment analysis," *Asian Journal of Computer Science and Technology*, vol. 8, p. 1–6, 2019.
- [5] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," *arXiv preprint arXiv:1103.2903*, 2011.
- [6] GeeksforGeeks, "Python – Sentiment Analysis using Affin," November 2020. [Online]. Available: <https://www.geeksforgeeks.org/python-sentiment-analysis-using-affin/>.
- [7] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [8] O. G. Yalcin, "Sentiment Analysis in 10 Minutes with BERT and TensorFlow," November 2020. [Online]. Available: <https://towardsdatascience.com/sentiment-analysis-in-10-minutes-with-bert-and-hugging-face-294e8a04b671>.
- [9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell and others, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, p. 1877–1901, 2020.
- [10] A. Go, R. Bhayani and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N project report, Stanford*, vol. 1, p. 2009, 2009.
- [11] S. Tayebi Arasteh, M. Monajem, V. Christlein, P. Heinrich, A. Nicolaou, H. Naderi Boldaji, M. Lotfinia and S. Evert, "How Will Your Tweet Be Received? Predicting the Sentiment Polarity of Tweet Replies," *arXiv e-prints*, p. arXiv–2104, 2021.
- [12] J. Akhtar, "An interactive multi-agent reasoning model for sentiment analysis: a case for computational semiotics," *Artificial Intelligence Review*, vol. 53, pp. 3987-4004, 2020.
- [13] Y. Shoham, "NLP in the 21st Century," 2021. [Online]. Available: <https://www.ai21.com/blog/nlp-21st-century>.
- [14] J. A. Buolamwini, "Gender shades: intersectional phenotypic and demographic evaluation of face datasets and gender classifiers," 2017.
- [15] "Gathering Strength, Gathering Storms: The One Hundred Year Study on Artificial Intelligence (AI100) 2021 Study Panel Report," September 2021. [Online]. Available: [https://ai100.stanford.edu/sites/g/files/sbiybj18871/files/media/file/AI100Report\\_MT\\_10.pdf](https://ai100.stanford.edu/sites/g/files/sbiybj18871/files/media/file/AI100Report_MT_10.pdf).

- [16] A. Cuncic, "Mental Health Effects of Reading Negative Comments Online," January 2021. [Online]. Available: <https://www.verywellmind.com/mental-health-effects-of-reading-negative-comments-online-5090287>.