Text Sentiment Analysis using Disaster Allied Tweets

Selvakumar G1, Adithi Mookambal2, Madhumidha3

School of Computer Science and Engineering (SCOPE)1,2,3

Vellore Institute of Technology1,2,3

Chennai, India

# **Abstract**

Social media platforms have played a key role in the expansion of information exchange and as a significant source of news for all types of industries and personal matters over the last decade. We can better understand the current opinions and values that people have around them by extracting and analyzing data from social media, and we can help the government or any other organization that analyses this type of data and approaches a solution to any problems that occur to keep an ideal around us. Modern algorithms from machine learning, deep learning, and transfer learning technologies can be used to analyze and learn more about data as it becomes more widely available, resulting in better outcomes.

***Keywords: Tweets, Sentiment Analysis***

# **Introduction**

Twitter is a social media platform that allows users to send and receive short messages known as tweets. Tweets can be up to 280 characters long and can include text, images, or links. Tweets can also be interacted with by liking, commenting, or retweeting them. There is no specific data available on the average number of tweets posted on Twitter each day because this varies depending on a few factors such as the number of users, the events and trends being discussed, and the time of year. Twitter, on the other hand, is estimated to receive millions of tweets per day from its users. With over 330 million monthly active users as of the fourth quarter of 2021, Twitter has a large and active user base. The platform is especially popular for news and information sharing, as well as discussions about current events, politics, and pop culture. Businesses and organizations also frequently use it for marketing and customer service.

Twitter data has been widely used for various purposes in artificial intelligence (AI) and machine learning (ML). Among the most common applications are

1. Sentiment analysis: Machine learning models have been trained using Twitter data to classify the sentiment expressed in tweets (positive, negative, or neutral). This has a wide range of applications, including marketing, customer service, and political analysis.
2. NLP: Twitter data has been applied to train machine learning models for tasks like language translation, text summarization, and text classification. Twitter data was utilized to investigate patterns and trends in social interactions, such as how information spreads on the platform and how users connect with one another.
3. Trend analysis: Twitter content is used to assess trends and patterns in popular topics and hashtags on the platform.
4. Event detection: Twitter data is being used in real-time to identify and track events and trends such as natural disasters and major news events.

Overall, Twitter's large volume of data and real-time nature make it a valuable resource for AI and ML applications, especially in natural language processing and social media analysis.

Twitter data has frequently been used to study and comprehend the social impact of pandemics and natural disasters. For example, researchers have examined how people cope with and respond to the COVID-19 pandemic, including how they share information and express their emotions, using Twitter data. Similarly, Twitter data has been applied to investigate how people react to natural disasters such as hurricanes and earthquakes, as well as how they seek and aid one another.

There are several ways that Twitter data can be clouded or biased during pandemics and disasters. One issue is that not everyone has equal access to the internet or social media, so the data may not be representative of the entire population. Additionally, people may be more likely to tweet about certain topics during a pandemic or disaster, which can skew the data. For example, during the COVID-19 pandemic, there may be more tweets about the virus and its impact on daily life, which could make it difficult to accurately analyze trends and patterns.

During pandemics and disasters, Twitter data can be clouded or biased in a variety of ways. The data may not be representative of the entire population because not everyone has equal access to the internet or social media. Furthermore, during a pandemic or disaster, people may be more likely to tweet about specific topics, which can skew the data. During the COVID-19 pandemic, for example, there may be more tweets about the virus and its impact on daily life, making it difficult to accurately analyze trends and patterns.

**Data pre-processing**

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:

* Getting the dataset
* Importing libraries
* Importing datasets
* Finding Missing Data
* Encoding Categorical Data
* Splitting dataset into training and test set
* Feature scaling

To create a machine learning model, the first thing we required is a dataset as a machine learning model completely works on data. The collected data for a particular problem in a proper format is known as the dataset. In order to perform data pre-processing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data pre-processing: NumPy, pandas, matplotlib

Now we need to import the datasets which we have collected for our machine learning project. But before importing a dataset, we need to set the current directory as a working directory. To set a working directory in Spyder IDE or any other anaconda editors. The next step of data pre-processing is to handle missing data in the datasets. If the dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.

Categorical data is data which has some categories such as, in our dataset; there are two categorical variable, **Country,** and**Purchased.** Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So it is necessary to encode these categorical variables into numbers. We divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose, if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. Feature scaling is the final step of data pre-processing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range.

**Text Vectorization**

A pre-processing layer which maps text features to integer sequences. This layer has basic options for managing text in a Keras model. It transforms a batch of strings (one example = one string) into either a list of token indices (one example = 1D tensor of integer token indices) or a dense representation (one example = 1D tensor of float values representing data about the example's tokens). This layer is meant to handle natural language inputs. To handle simple string inputs (categorical strings or pre-tokenized strings) see [tf.keras.layers.StringLookup](https://keras.io/api/layers/preprocessing_layers/categorical/string_lookup#stringlookup-class). The vocabulary for the layer must be either supplied on construction or learned via adapt(). When this layer is adapted, it will analyze the dataset, determine the frequency of individual string values, and create a vocabulary from them. This vocabulary can have unlimited size or be capped, depending on the configuration options for this layer; if there are more unique values in the input than the maximum vocabulary size, the most frequent terms will be used to create the vocabulary.

The processing of each example contains the following steps:

1. Standardize each example (usually lowercasing + punctuation stripping)
2. Split each example into substrings (usually words)
3. Recombine substrings into tokens (usually ngrams)
4. Index tokens (associate a unique int value with each token)
5. Transform each example using this index, either into a vector of ints or a dense float vector.

**Embedding techniques**

The initial embedding techniques dealt with only words. Given a set of words, you would generate an embedding for each word in the set. The simplest method was to one-hot encode the sequence of words provided so that each word was represented by 1 and other words by 0. While this was effective in representing words and other simple text-processing tasks, it didn’t really work on the more complex ones, such as finding similar words. A word embedding not only converts the word but also identifies the semantics and syntaxes of the word to build a vector representation of this information. Some popular word embedding techniques include Word2Vec, GloVe, ELMo, FastText, etc. The underlying concept is to use information from the words adjacent to the word.

## Universal Sentence Encoder

One of the most well-performing sentence embedding techniques right now is the Universal Sentence Encoder. And it should come as no surprise from anybody that it has been proposed by Google. The key feature here is that we can use it for Multi-task learning. This means that the sentence embeddings we generate can be used for multiple tasks like sentiment analysis, text classification, sentence similarity, etc, and the results of these asks are then fed back to the model to get even better sentence vectors that before.

The most interesting part is that this encoder is based on two encoder models and we can use either of the two:

#### Transformer

#### Deep Averaging Network(DAN)

Both of these models are capable of taking a word or a sentence as input and generating embeddings for the same. Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of [artificial intelligence or AI](https://www.ibm.com/topics/artificial-intelligence)—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics—rule-based modelling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance you’ve interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes. Sentiment analysis is used to detect or recognize the sentiment which is contained in the text. This analysis helps us to get the reference of our text which means we can understand that the content is positive, negative, or neutral.

Looking at the current scenario, all the business tycoons need to have a lucid idea of what kind of response their product is receiving from the customers and how the changes can be incorporated based on the arising demands.

Following are the steps involved in the process of sentiment analysis-

1. Importing the dataset. The dataset can be obtained from the authentic resources and can be imported into our code editor using read\_csv.
2. The next crucial step is to find out the features that influence the sentiment of our objective.
3. Once we draw the conclusion based on the visualization, we can move on to the next step which is creating a 'wordclouds'.
4. The next step is to classify the reviews into positive and negative.
5. Now we will create wordclouds for both the reviews.
6. The amount of obtained wordclouds in the dataset can be understood with the help of bar graphs.
7. The model can be built using-
   1. First, clean the data and make sure all the pre-processing stages are followed.
   2. The next step is to split the data frame which contains only the required features.
   3. Create a bag of words which means go for vectorization where text can be converted into integer matrix.
   4. Now we will import logistic regression which will implement regression with a categorical variable.
   5. Now let's split our data into independent variable and target.
   6. Let's take the training dataset and fit it into the model.
   7. Next, we can take the test dataset and make the prediction.
   8. The final task is to test the accuracy of our model using evaluation metrics.

**Models**

*Gaussian Naive bayes*

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset. Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

*Convolutional Neural Network*

A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals. CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces. This characteristic that makes convolutional neural network so robust for computer vision. CNN can run directly on a underdone image and do not need any pre-processing. A convolutional neural network is a feed forward neural network, seldom with up to 20. The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer. CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes. With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces. The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

*Long Short-Term Memory*

LSTM stands for long short-term memory networks, used in the field of Deep Learning. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. LSTM has feedback connections, i.e., it is capable of processing the entire sequence of data, apart from single data points such as images. This finds application in speech recognition, machine translation, etc. LSTM is a special kind of RNN, which shows outstanding performance on a large variety of problems.

*Recurrent Neural Network*

RNN is a type of Neural Network where the output from the previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence. RNN have a “memory” which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

*Universal Sentence Encoder V4*

The Universal Sentence Encoder encodes text into high dimensional vectors that can be used for text classification, semantic similarity, clustering, and other natural language tasks. The pre-trained Universal Sentence Encoder is publicly available in TensorFlow-hub. It comes with two variations i.e. one trained with Transformer encoder and other trained with Deep Averaging Network (DAN). The two have a trade-off of accuracy and computational resource requirement. While the one with Transformer encoder has higher accuracy, it is computationally more intensive. The one with DNA encoding is computationally less expensive and with little lower accuracy.

**Conclusion**

The process of automatically analyzing text data to determine the sentiment or emotion expressed in it is known as text sentiment analysis. Deep learning techniques, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, have grown in popularity for text sentiment analysis because of their ability to learn complex patterns and relationships in data. Pay close attention to context and meaning: Researchers have concentrated on developing models that consider the context and meaning of words in a text, rather than just the words themselves. This is referred to as semantic sentiment analysis. And, because it can speed up training and improve performance, the use of transfer learning, which involves using pre-trained models on large datasets as a starting point for training on a new task, has become a popular approach in text sentiment analysis. Overall, these trends reflect a focus on improving the accuracy and robustness of text sentiment analysis models and developing models capable of better understanding the text's context and meaning.