International Journal of Innovative Research in Science and Engineering
Volume 9, Issue 04, 2023
www.ijirse.com
ISSN: 2454-9665

Disaster Tweet Classification Using BERT Model

K. Vasavi, G. Chandra Sekhar, M. Avinash,
M. Jhansi Dharmesh, P. Jagadeesh, Mrs. V.Prathima
Department of Information Technology, Tirumala Engineering College

ABSTRACT

With today's technology, each person's online footmark opens the door for a largetreasure trove of information that can be used for numerous purposes that varies from analyzing market trends to understanding the general emotion of the people. The labels of every Tweet reflect different types of disaster-related information, which have different potential usagein emergency response. Whenever a person tweetsa message which was about an emergency or an impending disaster and this was recognized immediately by our BERT models, we would be able to react as fast as possible which helps to savelives of people. BERT is an open-source machine learning framework for natural language processing (NLP). The main aim of our project is todistinguish if a tweet talks about a real disaster or not.

Keywords— SVM , BERT Model , Google collab , NSP , MLM.

INTRODUCTION

The traditional approach for the text classification is generally make use of Logistic Regression, Support Vector Machine, Naïve Bayes, Gradient Boosting and other fundamental machine learning models. These methods require manual feature engineering process to encode textsequences in vector form, therefore can be fed into classifiers, which might suffer from the problem of typing error and generates noisy words to the corpus. The performance of the model depends on the way of encoding text into vector forms. Here, understanding the context of words is important to analyze a tweet intention. For example, a tweet like this "she says that no one is to blame for her illness. It was just an accident of nature.". Even though it contains the word, "accident" it does not mean any danger or emergency; rather, it is used to describe the normal conversation. Let's assume another tweet like this, "he had an accident at the factory", I got injured". Here, the word "accident" means disaster, and thetweet describes an emergency. The two examples show that one word could have multiple meanings supported its context. Therefore, understanding the context of words is important to analyze a tweet's sentiment and intention. Different researchers proposed different methods to understand the meaning of aword by representing it in embedding or vector Neural network-based methods such as Skip-gram, Fast Text are popular for learning word embeddings from large word corpus and are used for solving differing types of NLP tasks. These methods are also used for sentiment analysis of Twitter data However, those embedding learning methods provide static embedding for a single word in a document. Hence, the meaning of the word, "accident" would remain the same in the above two examplesfor these methods.

Existing model

- Social media analysis is an active, interdisciplinary research field which has enabled the researchers to gain unique perspectives into human and data behaviors.
- Most of the existing methodologies have been tried to detect the situational information during the disaster. The authors used uni-gram features for classifying the tweets into user-defined categories during the disaster.
- Most of the models are classifying the tweet as disaster by using the main keywords in that text, but most of the time those are not disaster.

PROPOSED MODEL

- Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing pre-training developed by Google.
- Unlike the existing model, BERT makes use of bidirectional learning to gain context of words from both left to right context and right to left context simultaneously.
- BERT is designed to help computers understand meaning of ambiguous language in text by using surrounding text to establish context.
- BERT introduced contextual word embeddings (one word can have a different meaning based on the words around it).

REQUIREMENT SPECIFICATION

Functional Requirements:

- The model should accept the input in the form of the text and the data having the keywords are identified and the data is processed.
- The output provided will be in the form of 0’s and 1’s so that 1 means that the data is disaster and the 0means that the tweet is not disaster . The data is collected and then trained with the CNN algorithm and data is preprocessed and then will be ready for testing.

Non-functional requirements:

- Reliability: This software will work reliably for alltype of tweets.
- Functionality: This software will deliver on thefunctional requirements mentioned in the document.
- Performance : Every tweet is classified as disaster ornot with high accuracy.

REQUIREMENT SPECIFICATION(Cont.)

Hardware Requirements:

- Ram:4GB
- Disc Space:5GB
- Processor : Intel i3Software Requirements:
- Python3
- Google Colab
- Numpy

- Seaborn
- keras

METHODOLOGY

- Collection of Data
- Data Visualization
- Data Preprocessing
- Testing and Training Model
- Word2Vec and SVM
- Classification with TF-IDF and SVM
- Building the BERT Model

METHODOLOGY(Cont.)

1. ***Collection of Data***.

- The data set is collected so that the data is trained andtested for further processing

. The data set contains data of previous tweets.So that thecollected data is used to train the deep learning models

- The dataset we have used is a labelled dataset thetconsists of a label whether it is disaster or not

2. ***DATA PREPROCESSING:***

- The text features in the train and test sets are noisy .
- One way to clean the feature is by removing the :1.Remove URLs

2. Remove emojis 3.Remove html content 4.Remove punctuation’s

3. Data Visualization:

- We will be visualizing the data. To know whatpercent of data is target which means the disaster one . The other percent know the data which isn’t
- We will be visualizing the data in the form ofRegular bar graph ,pie graph, word cloud.

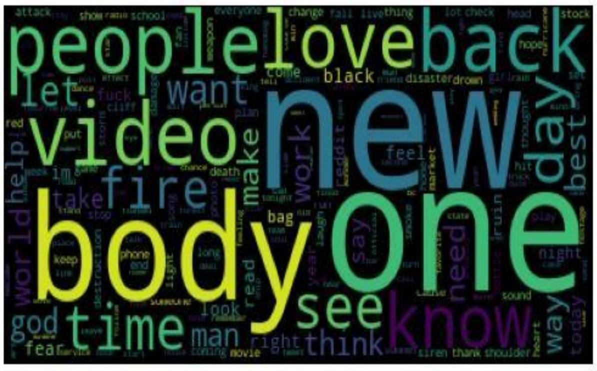

Fig . Words most used in non-target dataset

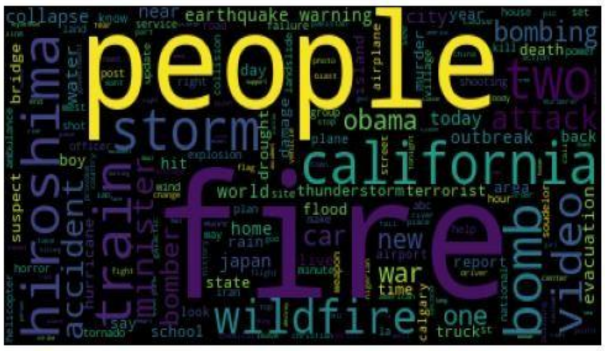

Fig . Words most used in target dataset

4. Testing and Training Model:

- For better accuracy we train our model with differentepochs.
- Hyper Parameters For Learner Modellearner.fit_onecycle(lr =2e-5,epochs =3)
- Learning rate (lr) –The rate at which the model learnsthe data.
- Epochs -Indicates the number of passes of the entire training dataset goes through the model or can be said as no of iterations of the dataset goes in the model.
- Batch Size –It is the no of data set points the modelconsiders .Here we have put the batch size as 64

5. Word2Vec and SVM (Existed model):

- From this vectors we convert them into numpy byusing the default method to_numpy().
- From this column and with target column we split thedata into train and split and we develop our SVM model.
- After the model developed we predict the results forother data and find accuracy for this model.
- The accuracy of this model is 57%.
- For to find the accuracy for this Word2vec and SVMmodel we use spacy large model.
- **SpaCy** is an open-source and free library for **Natural Language Processing** (NLP) in Python having a lot ofin-built functionalities. It’s becoming popular for processing and analyzing data in NLP.

6. Vectorization is a process of converting the text data into a machine-readable form. The words are represented as vectors. With this we created a new column as vector Classification with TF-IDF and SVM (Existed model)

- TF-IDF stands for **“Term Frequency — Inverse Document Frequency”**.
- This is a technique to quantify words in a set of documents. We generally compute a score for each word to signify its importance in the document and corpus.
- This method is a widely used technique in Information Retrieval and Text Mining.
- Term Frequency measures the frequency of a word ina document. This highly depends on the length of the document and the generality of the word.
- tf(t,d) = count of t in d / number of words in d.

- DF is the number of documents in which the word is present.
- df(t) = occurrence of t in N documents
- IDF is the inverse of the document frequency which measures the informativeness of term t.
- idf(t) = log(N/(df + 1))
- TF-IDF = Term Frequency (TF) * Inverse DocumentFrequency (IDF)
- tf-idf(t, d) = tf(t, d) * log(N/(df +1))
- **Terminology**
- t — term (word)
- d — document (set of words)
- N — count of corpus
- corpus — the total document set

7. Building the BERT Model:

- BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model in which everyoutput element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection.
- Historically, language models could only read text input sequentially -- either left-to-right or right-to-left - - but couldn't do both at the same time.
- BERT is different because it is designed to read in both directions at once. This capability, enabled by theintroduction of Transformers, is known as bidirectionality.

Conclusion:

Bidirectional Encoder Representations from Transformers (**BERT**) is a transformer-based machine learning technique for natural language processing (NLP) pre- training developed by Google. Twitter is one of highly sociableand authentic channel.For to identify the tweet is disaster or not ,we use BERT model for to do that task.The main application ofthis project to identify the tweet is disaster or not and help thosepeople immediately as soon as possible. From this project we conclude that BERT model gives a good accuracy for to predict the new tweet classification or text classification as compared to the other models. . In Future we are going to make this modelmore efficient and better .

References:

- https://www.kaggle.com/c/twitter-sentiment-analysis2
- https://towardsdatascience.com/workflow-of-a-machine-learning-project-ec1dba419b94
- https://towardsdatascience.com/tf-idf-for-document- ranking-from-scratch-in-python-onreal- world-dataset-796d339a4089
- https://www.tutorialspoint.com/uml/uml_standard_diagrams
- https://www.tutorialspoint.com/google_colab/what_is_goog le_colab.htm