International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Twitter Sentiment Analysis: An Unsupervised Approach

Shila Jawale[1], Deep Patil [2], Akshata Chirmade [3], Sanchita Bavdekar [4]

[1]*Asst.Professor, Information Technology, Datta Meghe College of Engineering, Navi Mumbai, India*
[2,3,4]*BE Student, Information Technology, Datta Meghe College of Engineering, Navi Mumbai, India*

Abstract: *Twitter is a very popular small blogging site where thousands of people exchange their thoughts every day in the form of tweets. Opinion investigation (Sentiment analysis) of Twitter data is a field that has been of attention over the last decade which involves dissecting "tweets" and the content of these expressions. Our Project presents the idea of segregating the Twitter data into specific clusters using unsupervised techniques and further classifying the tweets into positive, negative, or neutral. In this project, twitter data undergoes pre-processing phase and has been trained using an efficient word embedding model in such a way that, it becomes capable of testing the tweets and evoking the necessary emotions in feeds Tweets.*

Keywords—Sentiment Analysis, Stopwords, Fast text, Word2Vec.

1 INTRODUCTION

Today, social media platforms such as Twitter or Facebook have gained great popularity among many readers.

Twitter is a microblogging platform where massive instant messages (i.e.tweets) are posted every day. Sentiment analysis is also known as Opinion mining. This allows people to share and express their opinions regarding daily issues and send messages around the world in a simple way. The aim here is to find out the sentiments behind a particular text or group of text inputs given by the user on different subjects and topics. Twitter feed analysis for emotional analysis has become a major research and business activity. Twitter Sentiment Analysis (TSA) tackles the problem of analyzing the tweets in terms of the opinion they express.

In order to extract emotions from tweets, emotional analysis is used. Due to this reason, Twitter is used as an informative source by many organizations, institutions, and companies. The results from this can be used in many areas such as analyzing and monitoring mood changes, event-related emotions or release of a particular product, analyzing public view of government policies, etc.

Severe research has been done on Twitter data, for analyzing the tweets and their respective polarity. In this paper, we aim to review some research in this domain and study how to perform sentiment analysis on Twitter data using K-Means which is an unsupervised machine learning algorithm.

1.1 PROBLEM STATEMENT

Based on the dataset provided, segregate the Twitter data into specific clusters using unsupervised techniques.

1.2 OBJECTIVE

The main objective of this project is to implement an unsupervised algorithm for the automatic classification of text (raw data in the form of tweets) into specific clusters and further classify them into positive, negative, and neutral. Performing sentiment analysis to determine the attitude of mass is positive, negative, or neutral towards a subject of interest and then representing it in a graphical representation.

2 LITERATURE REVIEW

KMeans Clustering

The Kmeans algorithm is a repetitive algorithm that attempts to divide a database into separate small K-groups (collections) where each data point belongs to only one group. Here the clusters are formed of data points that are very similar to each other. Also, every cluster is different from the others. This algorithm finds the current centroid and then repeats the procedure till the optimal centroid is produced. It is known presumptuously how many collections there are. Also known as a flat clustering algorithm. It provides data points in a collection in such a way that the total square distance between the data points and the cluster's centroid (mathematical interpretation of all data points in that collection) is minimal.

We have implemented our project using the KMeans algorithm as it is one of the efficient algorithms as far as unsupervised techniques are concerned and predominantly works best in the case of unlabelled data.

2.1 Word Embedding Models

As our project mainly focuses on unsupervised techniques to be used, for purely unlabelled data it was important to use a word embedding model for word representations in vector form. While conducting a literature survey, we researched different word embedding models like Word2Vec, GloVe, and the fast text models.

Word2Vec

Word2Vec model creates vectors of the words that are distributed numerical representations of word features – these word features could comprise words that represent the context of the individual words present in our vocabulary. Word2Vec, a word embedding methodology, enables similar words to have similar dimensions and, consequently, helps bring context. For example, if we consider the sentence – "Sita likes to run in the park.", there can be pairs of context words and target words. If we consider a context window size of 2, we will have pairs like ([Sita, to], likes), ([to, in], run), ([likes, run], to), etc. An in-depth reading model can attempt to predict these target words based on contextual words.

GloVe

The GloVe model is an unsupervised learning algorithm for obtaining vector representations for words. This is accomplished by making a word map in a logical area where the distance between words is related to semantic similarity. Training is done on integrated global word counts that take place together in the corpus, and the resulting presentations show interesting sub-structures of the vector space name. The advantage of GloVe is that, unlike Word2vec, GloVe relies not only on location statistics (word-for-word information) but integrates global statistics (co-occurrence of words) to determine word vectors.

Fasttext

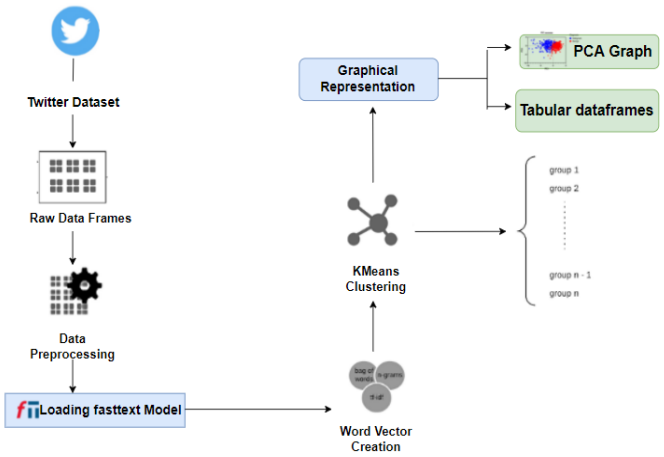
The fasttext model is a word embedding model for efficient classifications and word representations in vector form. FastText supports continuous word bag training (CBOW) or Skip-gram models using incorrect samples, softmax, or hierarchical softmax loss functions.

FastText is able to achieve really good performance of word representation and sentence division, especially in the case of rare words through character-level knowledge. Each word is represented as a bag of letters n-grams over the word itself, so for example, in the word matter, which has n = 3, the fastText presentations of the character n-grams are added as boundary markers to separate the word ngram from the word itself.

Out of these 3 models studied, we have implemented a fasttext model as it can be useful when the 'words' in the model aren't words for a particular language, and character level n-grams would not make sense. The most common use case is when we put in ids as words. During the model review, fastText reads the weights of each n-gram and the total word token. Word2vec and GloVe both fail to provide any vector representation in words that are not in the model dictionary. This is a great advantage of this approach.

3 METHODOLOGY

This section of the article sums up the methodology that has been applied and implemented for our project. It is divided into different processes such as data collection, data preprocessing, embedding model, and data analysis. The Natural Language Processing Toolkit (NLTK) which is a python-based platform is extensively used. The flow which this project follows is as below :



3.1 Data Collection

Data collection is defined as collecting and analyzing data to validate and research using some techniques. Firstly, a regulated dataset of Twitter is collected and used for the implementation. The dataset used nearly contains about 2 lakh rows, which is purely unlabelled.

3.2 Data Pre-processing

After collecting the data, the next step is to preprocess the data. This is an important phase in text processing as the data that is present is in its raw form containing a lot of special characters, URLs, hashtags & unnecessary symbols. Pre-processing refers to the transformation applied to the data before feeding it to the learning

algorithms. Thus, the Twitter data needs to undergo pre-processing. In this process, the stopwords and the lemmatized words are removed. The goal is to clean tweets to make them easier to read by a machine. There are many techniques out there for cleaning text. We have done text cleaning by lemmatization, stemming, and stop words.

In stemming, the words are reduced to their word stem that affixes to suffixes and prefixes or to roots of words known as a lemma. In simple words stemming to lower a word into a basic word or stem in such a way that the same words fall under the common stem. In both steam and lemmatization, the inflectional forms and words which are related to word-formation are converted to their basic form. (Ex: am, are, is => be / dog, dogs, dog's, dogs' => dog.) These reduce the corpus size and its complexity, allowing for simpler word embedding (am, are, and share the same exact word vector). Stop words are some common words that don't provide any weight for a machine's understanding of the given text. Examples of such stop words are; a, and, the, it, etc

NLTK supports termination name removal, and you can find a list of suspended names in the corpus module. To remove punctuation marks in a sentence, you can split your text into words and delete the word when it comes from the syllabus provided by NLTK.

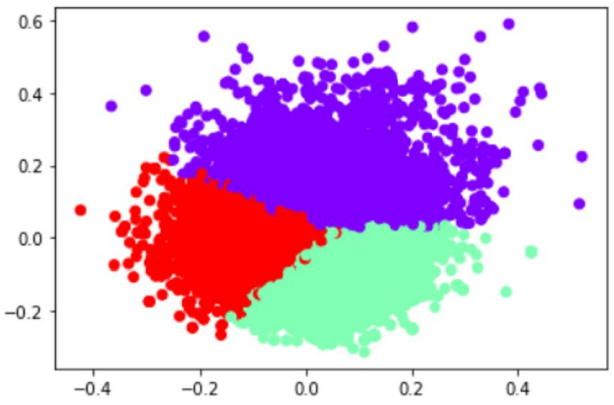
3.3 Loading The fasttext Model

The third step is training the cleaned or preprocessed data. In our case, once the data is cleaned it is trained using fasttext (which is a word embedding model) for efficient word representations in vector form. This model is considered a word bag model with a window that slides over the name because no internal structure of the word is considered. As long as the characters are inside this window, the order of the n-grams does not matter.

It is worthy to mention here that the fasttext model has nearly trained 45 million words from the dataset. The fastText model has worked well with rare words. So even if a word wasn't seen during training, it was later broken down into n-grams to get its embeddings.

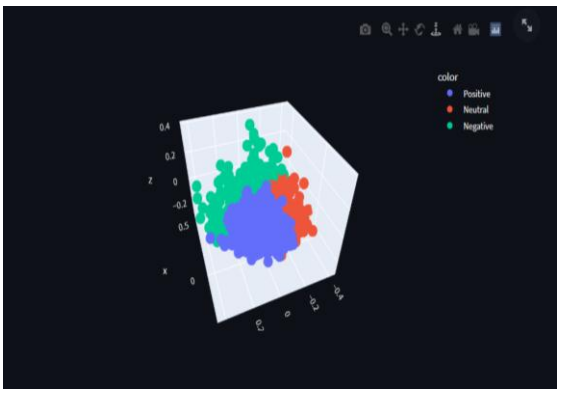
3.4 Applying KMeans Algorithm

Once the word vectors were created, the KMeans clustering algorithm was applied to them which resulted in the formation of clusters or groups based on the similarity found in their observations. In our case, 3 clusters were formed namely positive, negative and neutral. The graphical representation of clusters is given below :



3.5 Graphical Representation

The final representation is depicted in the form of a PCA Graph, 3D Graph, and also in the form of a tabular dataframe and visualized on streamlit.



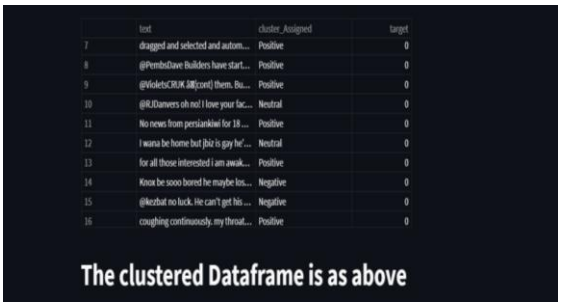


Fig: DataFrame with Assigned clusters.

4 . CONCLUSIONS

Twitter is a huge platform and source of improperly structured sentiment datasets that can be analyzed to produce trending emotions and many more. In Twitter sentiment analysis we inspect or mine each and every element of the tweet. This paper explains various steps involved in the analysis of Twitter sentiments along with the various tools that are used to perform Twitter sentiment analysis. It comprises steps like data

collection, text preprocessing, sentiment detection, sentiment classification, training, and testing of the model. For analyzing a tweet it is very necessary to know the morph and elements of the tweet. Each of these components and phases of sentiment analysis is briefly described in this review paper.

ACKNOWLEDGEMENT

Working on this project was a journey of immense knowledge and experience for us. However, this would have not been achievable without the contribution of all members. We would like to express our gratitude to Asst. Prof Shila Jawale for her guidance and valuable support.

REFERENCES

1. Analysis and Visualization of Twitter Data using k-means Clustering for International Conference on Intelligent Computing and Control Systems ICICCS 2017.

2. Analyzing Twitter Data using Unsupervised Learning techniques for Journal of Network Communications and Emerging Technologies (JNCET).

3. An Unsupervised Fuzzy Clustering Method for Twitter Sentiment Analysis for 2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions.

4. Automatic Unsupervised Polarity Detection on a Twitter Data Stream for 2014 IEEE International Conference on Semantic Computing.