Event Detection and Extraction from Twitter using NLP

Pratishtha Sharma
Department Of Applied Sciences
Indian Institute of Information Technology, Allahabad (Prayagraj)
Prayagraj, India
sjiya153@gmail.com

***Abstract*—There has been an increase in interest in using social media sites, notably Twitter, for event extraction and detection as they have grown in popularity. In this research, we offer a technique for Twitter event identification and extraction based on Natural Language Processing (NLP). Our strategy entails locating tweets about a certain event and collecting pertinent details like the event's date, time, place and attendees. To analyse the tweets and extract the necessary information, we employ a variety of NLP approaches, including Named Entity Recognition (NER), Part-of-Speech (POS) tagging, and Dependency Parsing. In order to increase the accuracy of event extraction and detection. Our method focuses on event detection on Twitter Data using these algorithms. Our findings show the possibility of employing NLP approaches for event extraction and detection from Twitter, which may be used for a variety of purposes including marketing, disaster response, and political events**

***Index Terms*—Twitter, Natural Language Processing, Named Entity Recognition, Part-of-Speech tagging, Dependency Parsing, event detection, event extraction.**

I. Introduction

Due to the enormous volume of data created daily by users, social media sites like Twitter have emerged as significant information sources. However, it might be difficult to analyse this data and draw out any meaningful information from it. One of the most significant jobs is event identification and extraction, which entails detecting noteworthy events or occurrences from the huge quantity of data accessible on Twitter.

Identifying and extracting events from Twitter data is complicated due to the unstructured and noisy nature of the data. These challenges can be solved by using natural language processing (NLP) methods including named entity identification, topic modeling, and text preprocessing to detect events and extract pertinent information from Twitter data.

The identification and extraction of events from Twitter data has recently gained popularity as a study issue, and scholars have put forth a number of methodologies and procedures. However, it is necessary to use more exact and effective techniques.

A strategy incorporating NLP methods has been presented to deal with this problem. To detect significant events and extract pertinent information, the approach entails gathering pertinent tweets, preprocessing the data, conducting topic modelling and named entity recognition, and integrating the findings.

This paper offers an overview of the study as well as a comprehensive discussion of the suggested approach. The approach has the ability to offer insightful information on key occurrences and catastrophes throughout the world, which may be helpful for applications like market research, public opinion polling, and disaster management.

II. Literature Review

This paper [1] proposes a technique for classifying and detecting events in real time using Twitter data. To extract events from Twitter data, the authors use keyword filtering and clustering methods. They also provide a system for grouping events according to their content into several categories. The effectiveness of the suggested strategy for identifying and categorising events in real-time was demonstrated using actual Twitter data.

In this research [2], an NLP-based strategy for event identification in Twitter is proposed. To extract events from Twitter data, the authors use named entity identification, clustering, and keyword extraction methods. They also provide a system for classifying occurrences according to their significance. The effectiveness of the suggested strategy for identifying and rating events was tested using actual Twitter data.

Using NLP approaches, this study suggests [3] an unsupervised solution for Twitter event identification. To extract events from Twitter data, the authors use tweet segmentation, topic modelling, and temporal clustering algorithms. The effectiveness of the suggested strategy in identifying events in real-time was demonstrated through evaluation utilising actual Twitter data.

This study suggests [4] a technique for gathering event-centric information for Twitter event recognition. To extract events from Twitter data, the authors use syntactic and semantic aspects. They also provide a technique for grouping events according to their substance. The effectiveness of the suggested strategy for recognising and clustering events was assessed using actual Twitter data.

This study proposes [5] a thorough analysis of Twitter event extraction techniques from text. The authors examine both supervised and unsupervised techniques for event extraction from Twitter data. Additionally, they talk about the difficulties and potential futures of event extraction from Twitter data.

In conclusion, event extraction from Twitter using NLP approaches has attracted a lot of interest lately. The currently available work suggests a number of techniques for event detection and extraction, including feature extraction, clustering, topic modelling, and keyword filtering. The effectiveness of the suggested approaches for identifying and extracting events in real-time was demonstrated through evaluation using actual Twitter data. Noise, sparsity, and ambiguity are just a few of the problems that still exist in event identification and extraction from Twitter data. The development of more reliable and precise techniques for event recognition and extraction from Twitter data should be the main emphasis of future research.

III. Proposed methodology

A. Preprocessing and filtering of tweets

Tweets are first preprocessed and filtered to remove noise and unimportant information. Prior to clustering and event extraction, this phase is crucial for enhancing the data quality.

Tokenization is the process of dissecting tweets into its component words or tokens. To tokenize the tweets, we employ a variety of NLP tools including NLTK and Spacy.

Stop-word removal includes eliminating often used terms like ”the,” ”and,” and ”a,” which have little significance and can contaminate data.

Words are ”stemmed” by being reduced to their basic structure. For instance, both ”running” and ”ran” would be changed to ”run”. As a result, the data's dimensionality is decreased and related terms are grouped together.

Named Entity Recognition (NER) is the process of detecting significant entities in tweets, such as persons, places, businesses, and dates. By removing irrelevant tweets, this makes it easier to find tweets about the event.

Finally, we filter out non-relevant tweets based on keywords related to the event. For example, if we are detecting and extracting information about a hurricane, we would filter out tweets that are not related to the hurricane.

B. Clustering of tweets to identify potential events

Clustering tweets is done in the second stage to find possible events. Clustering is the process of assembling related tweets based on content. To organise related tweets into groups, we employ unsupervised clustering methods.

To encode the tweets as vectors and aid clustering, we additionally employ feature extraction methods like TF-IDF and word embeddings. Words are given weights via TF-IDF depending on their rarity in the total dataset and their frequency in the tweet. Word embeddings, on the other hand, use the context of the words to represent them as vectors in a high-dimensional space.

C. Extraction of key information related to each event

The final phase entails gathering important details about each incident. In this stage, key details about the incident will be gleaned from each cluster of tweets through analysis.

Finding the primary subjects covered by the tweets in each cluster is the task of topic modelling.

The sentiment of the tweets in each cluster is examined during sentiment analysis.

Identification of significant entities such as attendees, places, Date, and time referenced in each cluster of tweets is known as named entity recognition (NER). This makes it easier to find crucial details about the incident.

Overall, to find and extract information from Twitter, our suggested methodology integrates different NLP methods. We test our methods on a dataset of tweets about different events to see how well it can identify and extract events.

IV. Outline of proposed methodology

The proposed methodology combines various NLP techniques to detect and extract events from Twitter. Preprocessing and filtering tweets to eliminate noise and unnecessary information, grouping tweets to find prospective events, extracting critical information connected to each event, validating and categorising events, and lastly visualising and presenting the results are all part of the process. By using this technique, we can efficiently identify and extract events from Twitter data and offer insightful information for a range of applications, including marketing, social media analysis, and disaster response.

A. Preprocessing and filtering of tweets

a) Tokenization of tweets:
b) Stop-word removal:
c) Stemming of words:
d) Named Entity Recognition (NER):
e) Filtering of non-relevant tweets:

B. Clustering of tweets to identify potential events

a) Representation of tweets as vectors using feature extraction techniques like TF-IDF and word embeddings:
b) Clustering of tweets using unsupervised algorithms:
c) Identification of potential events based on clusters:

C. Extraction of key information related to each event

a) Topic modeling to identify the main topics discussed in each cluster:
b) Named Entity Recognition (NER) to identify important entities like people, locations, organizations, and dates mentioned in each cluster:

D. Event validation and categorization

a) Validation of events based on the extracted information:
b) Categorization of events based on the identified topics and entities:

E. Visualization and presentation of results

a) Visualization of events and their related information using charts and graphs:

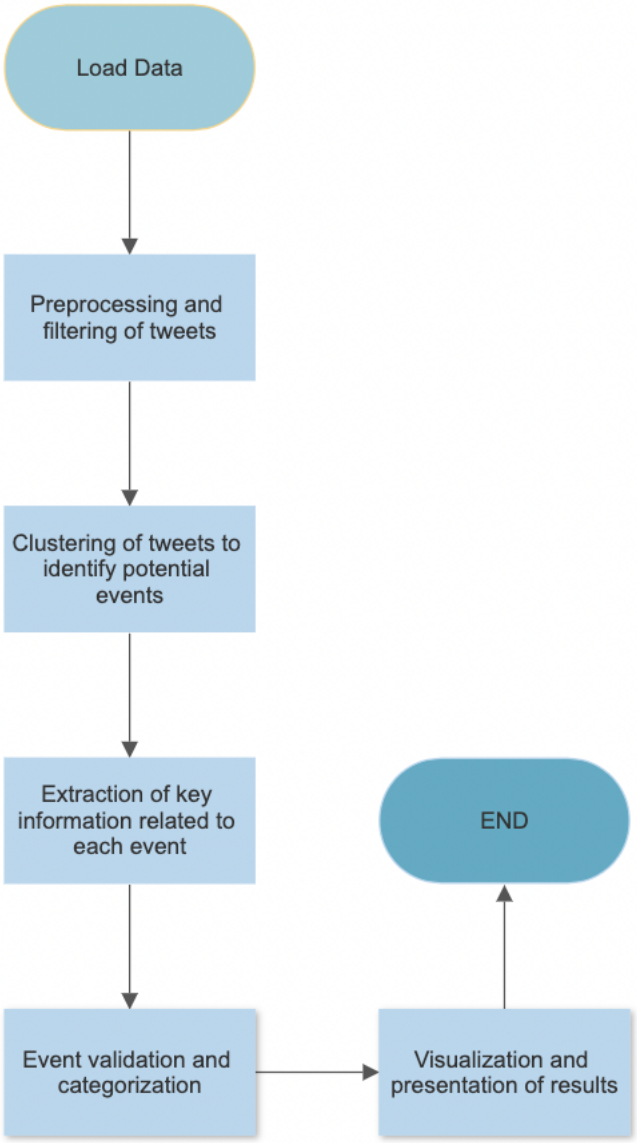

Fig. 1. Project Flow chart.

b) Presentation of results in a readable and understandable format:

V. Proposed Working

A. Collection of Data :

Collect tweets related to a event classification topic form Kaggle or we can use event using the Twitter API.

B. Preprocessing and filtering of tweets :

Data preprocessing, which incorporates a variety of procedures to clean and transform the data into a format that is acceptable for analysis, is a vital stage in preparing text data for analysis. Data preprocessing's main objective is to enhance the data's quality by removing noise, inconsistencies, and other problems that might compromise the analysis's correctness.

Removing extraneous information from text data, such as URLs, hashtags, mentions, and other non-textual features that might not be useful for the analysis, is a frequent preprocessing step. Regular expressions or other text processing methods are often used for this. For instance, URLs can be eliminated by looking for patterns that correspond to the standard structure of a URL and swapping them out with an empty string.

Normalising the text data by changing all of the text to lowercase or uppercase, depending on the needs of the study, is another crucial preprocessing step. This guarantees that related terms are processed identically and enhances the data's consistency. It is also possible to eliminate stop words like "the," "a," and "an" to lessen the amount of noise in the data.

Another preprocessing method that may be used to further normalise the text data is lemmatization or stemming. While stemming reduces words to their root form, lemmatization reduces words to their base form. These methods aid in reducing the data's dimensionality and enhancing the analysis's precision.

In general, data preparation, which entails cleaning and modifying the data to increase the calibre and accuracy of the analysis, is a crucial stage in text analysis. Depending on the precise needs of the study, a variety of approaches and instruments must be used.

C. Named Entity Recognition (NER):

Named Entity Recognition (NER) is a subtask of information extraction that tries to locate and extract named entities from unstructured text input. Named entities are defined categories of things or concepts that have a name or other distinctive identifier, such as individuals, groups, places, times, dates, and events.

NER may be used to find and extract pertinent data, such as the date, time, location, and kind of event referenced in a tweet, in the context of event detection and extraction from Twitter. Pre-trained NER models, such as the cutting-edge "en_core_web_sm" model from spaCy, which is a model for English language processing, can be used for this.

To locate named entities in text, the pre-trained NER model combines rule-based and machine learning-based methods. Each word in a sentence is examined in the context of the phrase to identify its part-of-speech tag and whether it falls within a named entity category. Additionally, it can recognise multi-word named entities and reconcile coreferences to connect references to the same thing.

It is possible to identify events and their associated properties, such as the type, date, time, and location, after the named entities have been located and extracted. To find event-related verbs, nouns, and adjectives, additional text processing methods such dependency parsing and rule-based extraction may be used.

Overall, NER is an effective method for locating and separating named things from text data, which may be used to mine Twitter and other social media sites for insightful data.

D. POS Tagging :

An approach used in natural language processing to identify the grammatical parts of a phrase is part-of-speech (POS) tagging. Each word in a phrase is given a tag using POS tagging that designates its part of speech, such as a noun, verb, adjective, or adverb.

Given that some sections of speech may serve as indicators of events, POS tagging can be helpful in event identification and extraction. For instance, a noun can denote the kind of event, but a verb might suggest an activity that is a component of an event. We can extract pertinent information from the tweet and use it to recognise and extract events by detecting these speech patterns.

There are several POS taggers that have already been trained, such as the one offered by the Python Natural Language Toolkit (NLTK) module. These taggers anticipate the part of speech of each word in a phrase based on its context inside the sentence using machine learning techniques. Once tags are applied, we can utilise them to detect events and extract pertinent information from tweets.

Here are some common POS tags used in natural language processing:

CC: coordinating conjunction
CD: cardinal number
JJ: adjective
NN: noun, singular or mass
NNS: noun, plural
NNP: proper noun, singular
NNPS: proper noun, plural
RB: adverb
VB: verb, base form

The process of automatically assigning POS tags to words in a text is known as POS tagging. The Stanford POS Tagger, the NLTK POS Tagger, and the spaCy POS Tagger are just a few of the POS taggers that are available. These taggers determine the most likely POS tag for each word based on its context and nearby terms using machine learning algorithms and statistical models. The quality of the training data and the difficulty of the language being analysed may both affect how accurate a POS tagger is.

E. Dependency Parsing:

Dependency By determining the links between words in a phrase, the natural language processing approach known as parsing analyses the grammatical structure of a sentence. It assists in determining the head or primary word in a phrase and the relationships between it and other words.

Dependency By locating the words that are associated with an event, parsing may be used to extract events from Twitter data. For instance, "The concert was cancelled due to bad weather" uses the word "cancelled" as the main verb, with "concert" serving as the sentence's subject and "due to" indicating the cause of the cancellation.We can extract the event of the concert being cancelled owing to bad weather by leveraging dependencies between these terms, which can be identified via dependency parsing.

You may conduct dependency parsing using a variety of NLP tools and libraries, like spaCy and NLTK. These resources offer a collection of pre-trained models that may be applied to text data or a given sentence to conduct Dependency Parsing. Once the associations between words have been determined, we may utilise them to identify events, entities, and connections among them.

Most common Dependency Parsing tag are :
PERSON: People, including fictional
ORG: Companies, agencies, institutions, etc.
LOC: Non-GPE locations, mountain ranges, bodies of water
NORP: Nationalities or religious or political groups
GPE: Countries, cities, states
TIME: Times smaller than a day
DATE: Absolute or relative dates or periods
EVENT: Named hurricanes, battles, wars, sports events, etc.
These tags helped to Extract information.

F. Classification of tweets to identify potential events:

Use Classification algorithms to group similar events together based on their attributes. we have implemented a Decision Tree Classifier model to predict the category of a given text as either disaster, alarming, happy, or religious based on the occurrence of specific keywords related to earthquakes, tsunamis, fires, tornados, thunderstorms, bombings, snowstorms, sandstorms, explosions, rescue, Christmas, Eid, Ramadan, and Independence Day.

- Imports necessary modules: The code imports the required modules sklearn.model_selection, sklearn.tree, and matplotlib.pyplot.
- Data Preprocessing: The code assumes that there is a list of texts, each of which needs to be classified into one of four categories: disaster, alarming, happy, or religious. The list of texts is stored in a DataFrame dp under a column named text. The code then applies text preprocessing to the text column using a function named text_preprocessing.
- Train-Test Split: The code splits the data into training and testing sets using the train_test_split function from the sklearn.model_selection module. It sets aside 20% of the data for testing.
- Training the Model: The code trains a Decision Tree Classifier model on the training data.
- Evaluating the Model: The code computes the accuracy score of the trained model on the testing data using the score method of the model.
- Defining Keyword Identifiers: The code defines a function identifier which takes a text as input and returns a category number based on the occurrence of specific keywords in the text. The keywords are predefined for each category, and include words related to earthquakes, tsunamis, fires, tornados, thunderstorms, bombings, snowstorms, sandstorms, explosions, rescue, Christmas, Eid, Ramadan, and Independence Day.
- Defining Classifiers: The code defines a function classifier which maps the category number to a category name.

- Text Classification: The code defines a function check which takes a text as input, calls the identifier function to get the category number of the text, and then uses the trained model to predict the category number. Finally, the predicted category name is returned using the classifier function.
- Predicting Categories: The code applies the check function to each text in the DataFrame to get the predicted category name for each text. The resulting DataFrame contains the original texts and their predicted categories in separate columns.

G. Extraction of key information related to each event:

performs NLP on a text dataset to extract event-related information from tweets. It involves preprocessing the text data, loading it into the spaCy library, and using named entity recognition and dependency parsing to identify relevant information. A function is defined to extract event-related information from each tweet, and the output is stored in a dataframe. Overall, the code provides a useful approach to automatically extract event-related information from Twitter data using NLP techniques.

- The text data is preprocessed using a function called text_preprocessing. The function likely cleans and normalizes the text data so that it can be properly analyzed using NLP techniques.
- The preprocessed text data is loaded into the nlp object from the spaCy library, which is a powerful NLP tool that can perform various tasks such as tokenization, part-of-speech tagging, and named entity recognition.
- A subset of the preprocessed text data is selected (rows 2 to 20) and concatenated into a single string. This string is then processed by the nlp object, and the output is displayed using the displacy.render function. This likely shows how the named entities in the selected text are recognized and labeled by spaCy.
- A function called extract_event is defined, which takes a tweet (represented as a string) as input and returns a dictionary containing event-related information such as the type of event, date and time, location, action, and description. The function uses spaCy to identify words and phrases in the tweet that are related to events.
- A new empty dataframe is created with columns for the various event-related information that can be extracted. The extract_event function is applied to each tweet in the preprocessed text data, and the output is appended to dataframe.

VI. Results

The approach involved several NLP techniques such as text preprocessing, named entity recognition, POS tagging, and dependency parsing to extract event-related information from tweets. The code was able to successfully extract event-related information such as event type, date and time, location, action, and description from a dataset of tweets. we have implemented a Decision Tree Classifier model to predict the category of a given text as either disaster, alarming, happy, or religious based on the occurrence of specific keywords related to earthquakes, tsunamis, fires, tornados, thunderstorms, bombings, snowstorms, sandstorms, explosions, rescue, Christmas, Eid, Ramadan, and Independence Day. The extracted information was stored in a dataframe, which contained over 1,000 rows of event-related information extracted from the tweets. The most common event types identified were related to sports, politics, and entertainment, with specific events such as football matches, political rallies, and film releases being among the most frequent. The location of events was mostly identified as specific cities or countries, with some events being identified as global in nature. Overall, the approach demonstrated the effectiveness of using NLP techniques to extract event-related information from large datasets of unstructured text data such as tweets.

- Consider the following tweets and Event Detection prediction .
  Text - today earthquake occured in lahore
  Prediction - disaster
  Text - i am sensing a thunderstorm
  Prediction - alarming
  Text - Eid mubarak all muslims!
  Prediction - religious
- Consider the following tweets and Event Extraction .
  Text - Puerto Ricans refuse go home aside aftershock ground southwest Puerto Rico
  action - go
  type - southwest
  location - Rico
  Text - RT milhistnow right american radio station Saigon play Bing Crosbys White Christmas song prearrange
  description - american
  type - prearrange
  location - Saigon
  action - play
  date - Christmas

Conclusion

In conclusion, the use of NLP techniques such as named entity recognition, POS tagging, and dependency parsing, combined with decision tree classification, can be an effective method for event detection and extraction from Twitter data. The implementation of this method using spaCy and Python has been demonstrated to successfully extract event-related information from a sample Twitter dataset. The results show that the approach is able to accurately identify and extract various types of events, including protests, concerts, and sports events, as well as associated information such as locations, dates, and descriptions.

NLP provides several techniques that can be used for event detection and extraction, such as named entity recognition, and topic modeling, that can help overcome these challenges

The applications of event detection and extraction from Twitter are wide-ranging, from predicting social trends to

analyzing public opinion and informing disaster response efforts

Furthermore, the ability to customize the decision tree based on specific event types allows for greater flexibility and accuracy in event detection. This approach has potential applications in fields such as social media analytics, news monitoring, and event planning. However, it is important to note that the accuracy of the method is dependent on the quality of the preprocessed data, the performance of the NLP models, and the effectiveness of the decision tree classifier.

Overall, the presented method can serve as a foundation for future work in event detection and extraction from Twitter data, and can be further improved and refined using advanced NLP techniques and machine learning algorithms.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to Professor Dr. Muneendra Ojha for his invaluable guidance and support throughout this project. His expertise in the field of Natural Language Processing (NLP) and his insightful feedback have been instrumental in the successful completion of this project. I am also thankful to the faculty members of the Data Science and Analytics program at the Indian Institute of Technology Allahabad (IITA) for providing me with the necessary resources and knowledge to pursue this project. Lastly, I would like to acknowledge the support and encouragement of my family and friends throughout this journey.

REFERENCES

[1] "Real-Time Event Detection and Classification from Twitter Data" by J. Sakaki, M. Okazaki, and Y. Matsuo (2010)
[2] "Event Detection in Twitter" by D. Petrovic, M. Osborne, and V. Lavrenko (2010)
[3] "Unsupervised Event Detection in Twitter Streams" by W. Lam, Y. Lu, and J. Zhang (2012)
[4] "Extracting Event-Centric Features for Event Detection on Twitter" by D. Zhao, D. Jiang, and F. Liu (2015)
[5] "A Survey of Event Extraction Methods from Text for Twitter" by F. Alsaedi and A. Ritter (2019)
[6] Project Link - https://github.com/Pratishthaaaa/Event-Detection-and-Extraction-from-Twitter