



# Event Detection and Extraction from Twitter

## Using NLP

By- Pratishtha Sharma (IDS2022004)

Project Link- <https://github.com/Pratishthaaaa/Event-Detection-and-Extraction-from-Twitter>

# **Problem Formulation:**

## **Defining our problem statement...**

- Event Detection
- Event Extraction from Twitter

# Dataset

# Twitter tweets with event classification

## From Kaggle

- Data contains 1104 rows and 3 columns
- Event column has 14 unique values
- Class columns have 4 different values

A text	=	A event	=	A class	=
RT @MichaelDeLau... 1% RT @Devinder_Shar... 1% Other (1082) 98%		rescued 12% tsunami 9% Other (869) 79%		disaster 28% alarming 26% Other (511) 46%	
3.6 aftershock at 4:05 am. I was asleep and the movement of my bed woke me up. Not so scary but I wi...		earthquake		disaster	
Good Night my friends, As per USGS, the new 5.9 quake in this morning caused the 7-day aftershock fo...		earthquake		disaster	
Yea it wasnt that long she killed me in like 6 or 7 seconds but not sure would have to <a href="https://t.co...">https://t.co...</a>		earthquake		disaster	

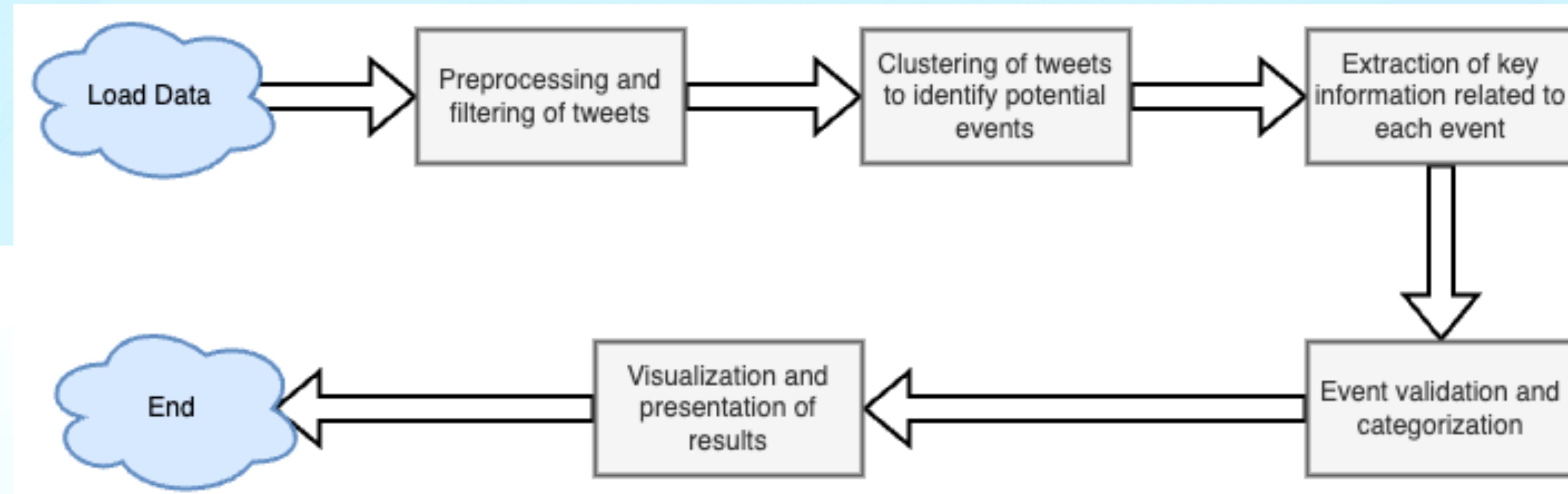
# Event detection and extraction ?

- Event detection and extraction is the process of automatically identifying and extracting relevant events from unstructured text data, such as social media posts, news articles, or other textual sources. An event can be any occurrence that is of interest to the user, such as a natural disaster, a sporting event, a political rally, or a product launch.
- Event detection involves identifying the relevant events from a large volume of unstructured text data, while event extraction involves extracting relevant information related to those events, such as the time, location, and sentiment. This information can be useful for various applications, such as predicting stock prices, analyzing public opinion, and predicting social trends.
- NLP techniques, such as named entity recognition, sentiment analysis, and topic modeling, can be used to identify and extract events from text data.

# Introduction

- Event detection and extraction from Twitter is a crucial task in natural language processing (NLP)
- Twitter is a rich source of real-time data, and event detection and extraction using NLP can help in various fields, such as disaster response, market research, and political analysis
- In this presentation, we'll explore the challenges of event detection and extraction from Twitter, the role of NLP in this process, different methodologies used, evaluation metrics, and applications

# Project Flow chart



# Preprocessing and filtering of tweets

- Tokenization of tweets
- Stop-word removal
- Stemming of words
- Named Entity Recognition (NER)
- Filtering of non-relevant tweets

# NER

## Named Entity Recognition

- Named Entity Recognition (NER) is a natural language processing (NLP) technique that involves identifying and classifying entities in text into predefined categories such as person, organization, location, date, time, and others.
- NER can be used to automatically extract information from large collections of unstructured text, such as news articles, social media posts, and scientific papers. It is an important component of many NLP applications, such as information extraction, question answering, text summarization, and sentiment analysis.
- The task of NER involves identifying the boundaries of entities and assigning them to the appropriate categories. For example, in the sentence "John Smith works at Google in New York", the named entities are "John Smith" (person), "Google" (organization), and "New York" (location). NER algorithms typically use machine learning techniques, such as rule-based systems, statistical models, and deep learning architectures, to perform this task.

# Visualizing A dependency parse or named entities in a text

yea be not long kill like second ORDINAL sure would, honored essay AfterShock PERSON mark th anniversary Alvin Tofflers Future Shock Compendium ORG  
essay, ticket on SALE Aftershock nowplaying khdradio, come we aftershock, oO I think I feel aftershock, a quick comparison ongoing M Puerto Rico GPE earthquake  
sequence M Central California LOC sequenc, why Puerto Ricans NORP refuse go home aside aftershock ground southwest Puerto Rico GPE , inception the  
Social Network Easy a Black Swan Toy Story ORG the kid all right aftershock, Puerto Rico GPE hit another magnitude aftershock KYMA, Puerto Rico GPE hit  
another magnitude aftershock KYMA ORG , a hard hit, Musharaf verdict mild aftershock extension one CARDINAL worry come extention, Puerto Rico GPE hit  
another magnitude aftershock KYMA, Kumera PERSON yellow Aftershock PHFOSSIL PERSON feature bankbtn Good Living ORG magazine Photography Stylist  
shino, Puerto Rico GPE hit another magnitude aftershock KYMA ORG , please pray PuertoRico GPE , Puerto Rico GPE hit another magnitude aftershock  
KYMA ORG , YakovZ

# A dependency parse or named entities in a text

- ORDINAL: "first", "second", etc.
- 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
- CARDINAL: Numerals that do not fall under another type
- **TIME**: Times smaller than a day
- **DATE**: Absolute or relative dates or periods
- **EVENT**: Named hurricanes, battles, wars, sports events, etc.

# Clustering of tweets to identify potential events

## Event Detection

- Preprocessing the dataframe
  - Preprocessing text
  - Label Encoder for event and class
  - Cleaning the data
- Using DECISION TREE ALGORITHM created a model
- Created an identifier function to identify event
- Created a classifier function to identify class
- Created a check function to get predictions using same model
- Created Dataframe as a result having columns text and predictions.

```

1 def identifier(text):
2     earthquake = ['earthquake', 'aftershock', 'aftershocks']
3     tsunami = ['tsunami', 'flood']
4     fire = ['wildfire', 'fire']
5     tornedo = ['tornedo', 'twister', 'cyclone']
6     thunderstorm = ['thunderstorm', 'storm']
7     bombing = ['suicide bombing', 'bomb', 'bomber', 'explosive']
8     snowstorm = ['snowstorm']
9     sandstorm = ['sandstorm']
10    explosion = ['explosion', 'blast']
11    rescued = ['rescued', 'recovered', 'survived']
12    christmas = ['christmas', 'xmas']
13    eid = ['eid', 'eid ul fitr', 'eid ul adha']
14    ramadan = ['ramzan', 'ramadan']
15    independence = ['independence day', 'independence', 'day of independence']
16    str = text.lower()
17    if any(x in str for x in earthquake):
18        return 1
19    if any(x in str for x in tsunami):
20        return 12
21    if any(x in str for x in fire):
22        return 13
23    if any(x in str for x in tornedo):
24        return 11

```

```

1 def check(x):
2     try:
3         a = classifier(model.predict([[identifier(x)]]))
4         return a
5     except:
6         print("")

```

# Code and Output:

```

1 def classifier(x):
2     if x == 1:
3         a = 'disaster'
4         return a
5     if x == 0:
6         a = 'alarming'
7         return a
8     if x == 2:
9         a = 'happy'
10        return a
11    if x == 3:
12        a = 'religious'
13        return a

```

	text	predictions
0	today earthquake occur lahore	disaster
1	sense thunderstorm	alarming
2	Eid mubarak muslim	religious

# **Extraction of key information related to each event**

## **Event Extraction**

- Load Lib and Spacy models
- Created a function to extract event
  - Define the event-related POS tags and named entity labels
  - Extract the event-related information from the tweet
  - Return the all the attached information to pandas data frame
- Save CSV file

# Code and Output:

```
2 def extract_event(tweet):
3     doc = nlp(tweet)
4
5     # Define the event-related POS tags and named entity labels
6     event_pos_tags = ["NOUN", "VERB", "ADJ"]
7     event_ner_labels = ["DATE", "TIME", "GPE", "LOC", "EVENT"]
8
9     # Extract the event-related information from the tweet
10    event = {}
11    for token in doc:
12        if token.pos_ in event_pos_tags or token.ent_type_ in event_ner_label:
13            if token.ent_type_ in event_ner_labels:
14                if token.ent_type_ == "DATE":
15                    event["date"] = token.text
16                elif token.ent_type_ == "TIME":
17                    event["time"] = token.text
18                elif token.ent_type_ in ["GPE", "LOC"]:
19                    event["location"] = token.text
20                elif token.ent_type_ == "EVENT":
21                    event["type"] = token.text
22            else:
23                if token.pos_ == "NOUN":
24                    event["type"] = token.text
25                elif token.pos_ == "VERB":
26                    event["action"] = token.text
27                elif token.pos_ == "ADJ":
28                    event["description"] = token.text
29
30    return event
31
```

```
1 extract_event(tex_list[813])
```

```
{'type': 'year', 'date': 'christmas', 'description': 'new', 'action': 'sayup'}
```

# The Challenges of Event Detection and Extraction from Twitter

- Twitter data poses several challenges for event detection and extraction, such as the high volume, velocity, and variety of data
- Traditional keyword-based approaches are limited in their ability to detect and extract events accurately due to the nuances of language and context
- NLP can help overcome these challenges by providing techniques such as entity recognition, sentiment analysis, and topic modeling

# The Role of NLP in Event Detection and Extraction

- NLP provides several techniques that can be used for event detection and extraction, such as named entity recognition, sentiment analysis, and topic modeling
- Named entity recognition helps identify relevant entities in tweets, such as people, organizations, and locations, which can help identify events
- Sentiment analysis can help identify the sentiment of tweets related to an event, which can provide insights into the public opinion on the event
- Topic modeling can help identify relevant topics and themes in tweets, which can help in event detection and extraction

# Applications of Event Detection and Extraction from Twitter

- Event detection and extraction from Twitter has several applications, such as disaster response, crisis management, market research, and political analysis
- For example, event detection and extraction can be used to predict the spread of diseases during a pandemic, monitor social unrest during protests, and predict the outcome of elections
- NLP can help in these applications by providing insights into public opinion, identifying emerging trends, and providing real-time information

# Conclusion

- 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
- Overall, event detection and extraction from Twitter using NLP is a powerful tool that can provide valuable insights and real-time information for various fields.



**Thank You!!**