
# Extracting Events from Messages

## with NLP

BY- Gomtesh Jain (IDS2022006)

https://github.com/gomtesh/Extracting-Events-from-Messages

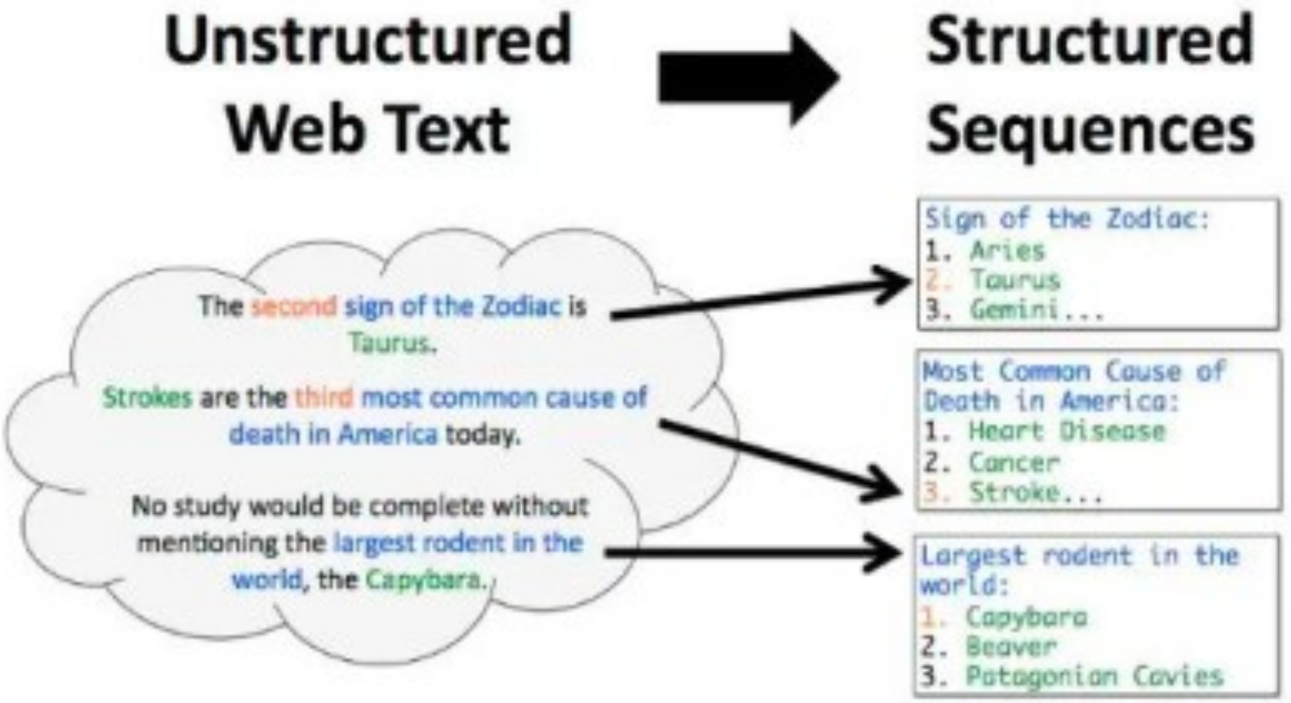
# Problem Formulation:

## Defining our problem statement...

- Extracting Events from Text Messages
- Extracting Events from Emails with NLP

# What is information extraction?

- The process of sifting through unstructured data and extracting vital information into more editable and structured data forms is known as information extraction.



# What is Event Extraction ?

- Is the process of gathering knowledge about periodical incidents found in texts, automatically identifying information about what happened and when it happened.
- Example -
  - The criteria 8 coordinators meeting will be at 4.10pm today in conference hall. Inconvenience is regretted
    - Subject: The criteria 8 coordinators meeting
    - Place: conference hall
    - Date: 16-04-2016
    - Time: 4.10pm

# Dataset

# The Enron Email Dataset

## From Kaggle

- The Enron email dataset contains approximately 500,000 emails generated by employees of the Enron Corporation. It was obtained by the Federal Energy Regulatory Commission during its investigation of Enron's collapse.

| | file | message |
|---|---|---|
| 0 | allen-p/_sent_mail/1. | Message-ID: <18782981.1075855378110.JavaMail.e... |
| 1 | allen-p/_sent_mail/10. | Message-ID: <15464986.1075855378456.JavaMail.e... |
| 2 | allen-p/_sent_mail/100. | Message-ID: <24216240.1075855687451.JavaMail.e... |
| 3 | allen-p/_sent_mail/1000. | Message-ID: <13505866.1075863688222.JavaMail.e... |
| 4 | allen-p/_sent_mail/1001. | Message-ID: <30922949.1075863688243.JavaMail.e... |

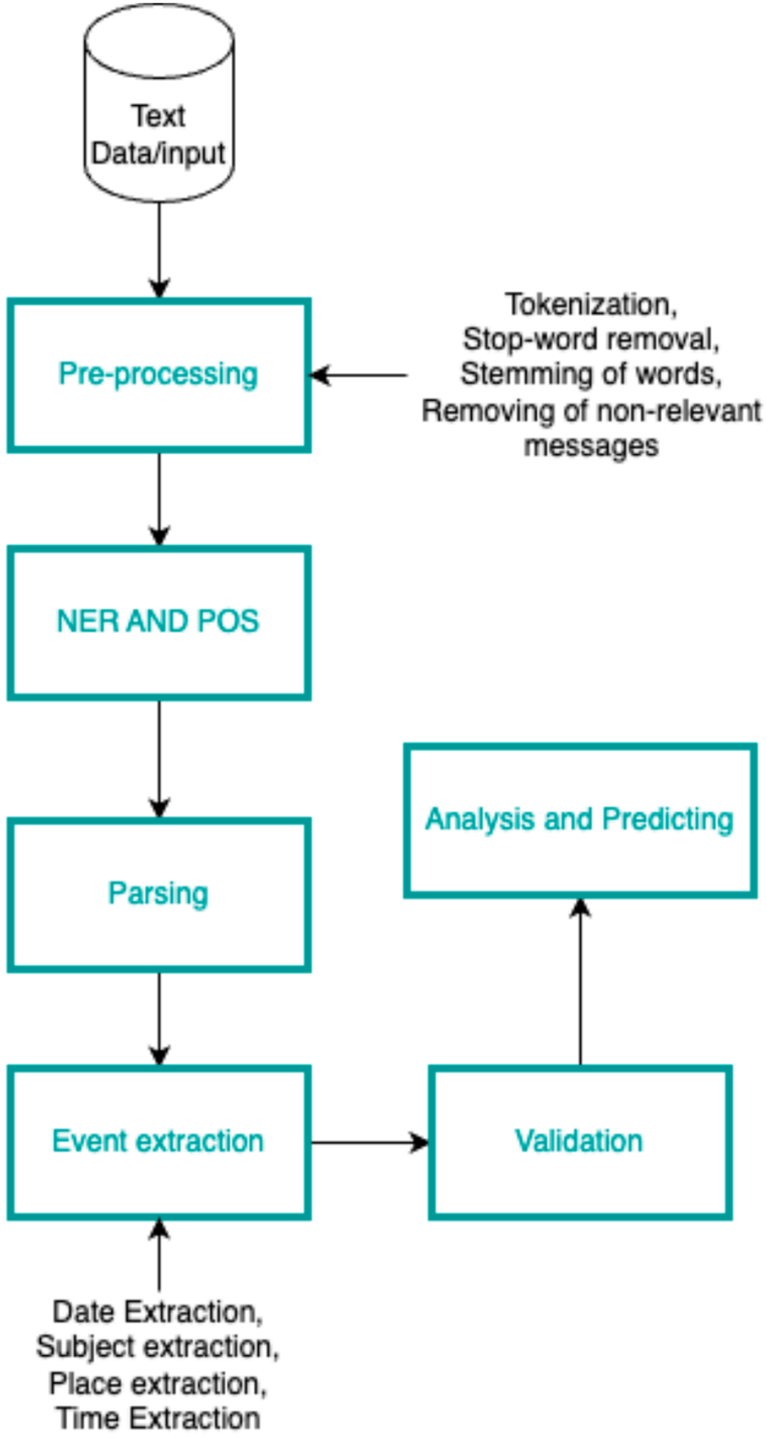
```
allen-p/_sent_mail/10.


Message-ID: <15464986.1075855378456.JavaMail.evans@thyme>
Date: Fri, 4 May 2001 13:51:00 -0700 (PDT)
From: phillip.allen@enron.com
To: john.lavorato@enron.com
Subject: Re:
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Phillip K Allen
X-To: John J Lavorato <John J Lavorato/ENRON@enronXgate@ENRON>
X-cc:
X-bcc:
X-Folder: \Phillip_Allen_Jan2002_1\Allen, Phillip K.\'Sent Mail
X-Origin: Allen-P
X-FileName: pallen (Non-Privileged).pst

Traveling to have a business meeting takes the fun out of the trip.  Especially if you have to prepare a presentation.  I would su

As far as the business meetings, I think it would be more productive to try and stimulate discussions across the different groups

My suggestion for where to go is Austin.  Play golf and rent a ski boat and jet ski's.  Flying somewhere takes too much time.
```

# Emails Events
## From Kaggle

- The Enron email dataset contains approximately 16500 Text messages

| | text | label |
|---|---|---|
| 0 | It's the first Monday after I walked out on my... | 1 |
| 1 | Our work schedule encompassed five intense day... | 1 |
| 2 | The highways were quiet on the Monday mid-morn... | 1 |
| 3 | He left early each Friday afternoon, often ret... | 1 |
| 4 | The difference between her before her Monday n... | 1 |

# Project Pipeline



# Pre-processing

- Tokenization
- Stop-word removal
- Stemming of words
- removing of non-relevant messages
- Passing it to nlp = spacy.load(‘en_core_web_sm')

# Named Entity Recognition

## NER

- The goal of NER is to identify and extract named entities from a given text
- Named entities include people's names, locations, organizations, dates, and more
- NER is important for many NLP applications, such as information retrieval, text classification, sentiment analysis, and machine translation
- Machine learning techniques, such as statistical models and deep learning algorithms, are used for NER
- Popular NER tools and frameworks include SpaCy, NLTK, Stanford NER, and AllenNLP.

```
1 displacy.render(doc, style="ent", jupyter=True)
```

BEIRUT ORG , LEBANON GPE ( 10:20 P.M. TIME ) – The Russian Air Force ORG has launched several airstrikes over the eastern countryside of Aleppo GPE tonight TIME , hitting several of the Islamic State's ORG ( ISIS ORG ) positions between Deir Hafer PERSON and Al-Bab PERSON . Russian NORP jets traveled from the Hmaymim Military Airport FAC in southwest Latakia GPE to the Aleppo GPE Governorate tonight TIME in order to aid the Syrian NORP and Turkish NORP armies currently battling with the Islamic NORP State forces in the Al-Bab Plateau FAC and Deir Hafer Plain PERSON . According to local reports, Russian NORP jets primarily focused on the road leading from Al-Bab GPE to Deir Hafer PERSON ; this area is where the Syrian Arab Army ORG is currently attacking the Islamic NORP State forces.

Russian NORP jets are still launching airstrikes this minute TIME , forcing the Islamic NORP State terrorists to to avoid launching counter-attacks against the Syrian NORP and Turkish NORP armies in east Aleppo GPE .

# Part-of-speech and Dependency tags

- `ADJ`: adjective, e.g. big, old, green, incomprehensible, first
- `ADP`: adposition, e.g. in, to, during
- `ADV`: adverb, e.g. very, tomorrow, down, where, there
- `AUX`: auxiliary, e.g. is, has (done), will (do), should (do)
- `CONJ`: conjunction, e.g. and, or, but
- `CCONJ`: coordinating conjunction, e.g. and, or, but
- `DET`: determiner, e.g. a, an, the
- `INTJ`: interjection, e.g. psst, ouch, bravo, hello
- `NOUN`: noun, e.g. girl, cat, tree, air, beauty
- `NUM`: numeral, e.g. 1, 2017, one, seventy-seven, IV, MMXIV

## spacy.explain("GPE")

```
{'CARDINAL': 'Numerals that do not fall under another type',
 'DATE': 'Absolute or relative dates or periods',
 'EVENT': 'Named hurricanes, battles, wars, sports events, etc.
 'FAC': 'Buildings, airports, highways, bridges, etc.',
 'GPE': 'Countries, cities, states',
 'LANGUAGE': 'Any named language',
 'LAW': 'Named documents made into laws.',
 'LOC': 'Non-GPE locations, mountain ranges, bodies of water',
 'MONEY': 'Monetary values, including unit',
 'NORP': 'Nationalities or religious or political groups',
 'ORDINAL': '"first", "second", etc.',
 'ORG': 'Companies, agencies, institutions, etc.',
 'PERCENT': 'Percentage, including "%"',
 'PERSON': 'People, including fictional',
 'PRODUCT': 'Objects, vehicles, foods, etc. (not services)',
 'QUANTITY': 'Measurements, as of weight or distance',
 'TIME': 'Times smaller than a day',
 'WORK_OF_ART': 'Titles of books, songs, etc.'}
```

# Text Messages or Any Text

# Code Approach

```
# Extract information from spaCy's entities and noun chunks
for ent in doc.ents:
    if ent.label_ == 'GPE' or ent.label_ == 'LOC':
        place.append(ent.text)
    elif ent.label_ == 'PERSON':
        subject.append(ent.text)
    elif ent.label_ == 'TIME':
        time.append(ent.text)
    elif ent.label_ == 'EVENT':
        event.append(ent.text)
    elif ent.label_ == 'DATE':
        date.append(ent.text)
for chunk in doc.noun_chunks:
    if chunk.root.dep_ == 'nsubj' and chunk.root.head.pos_ == 'VERB':
        event.append(chunk.text)
    elif chunk.root.dep_ == 'pobj' and chunk.root.head.pos_ == 'ADP' and chunk.root.head.head.pos_ == 'VERB':
        place.append(chunk.text)
    elif chunk.root.dep_ == 'npadvmod' and chunk.root.head.pos_ == 'VERB':
        time.append(chunk.text)
    elif chunk.root.dep_ == 'pobj' and chunk.root.head.pos_ == 'ADP' and chunk.root.head.head.pos_ == 'VERB':
        date.append(chunk.text)
```

# Code Pipeline

- Load spaCy's English model and other Lib.
- Define function to extract place, event, time, date, and subject from text
  - Parse text with spaCy
  - Initialize variables to store information
  - Extract information from spaCy's entities and noun chunks
  - Create a dictionary to store the extracted information
- Define function to create a DataFrame from a list of dictionaries
- Save the results(DF) to CSV

# Results

- text = "The annual conference on machine learning will take place on May 10-12, 2023 at 10:00 AM at venu the San Francisco Marriott Marquis. Keynote speakers include Andrew Ng and Yoshua Bengio."

```
Place: ['San Francisco', 'May', '10:00 AM']
Event: ['The annual conference', 'Keynote speakers']
Time: ['10:00 AM']
Date: ['annual', 'May 10-12, 2023']
Subject: ['Andrew Ng', 'Yoshua Bengio']
Important things: Keynote speakers include Andrew Ng and Yoshua Bengio.
```

# E-mail Text

# Code Approach

```
# Parse email
msg = email.message_from_string(message)
sender = msg['From']
recipient = msg['To']
subject = msg['Subject']
body = msg.get_payload()

# Tokenize text
doc = nlp(body)

# Extract named entities
dates = []
times = []
places = []
for ent in doc.ents:
    if ent.label_ == 'DATE':
        dates.append(ent.text)
    elif ent.label_ == 'TIME':
        times.append(ent.text)
    elif ent.label_ == 'GPE':
        places.append(ent.text)

# Perform sentiment analysis
blob = TextBlob(body)
sentiment = blob.sentiment.polarity
# Extract important information
action_items = re.findall(r'(?:\n|^)[\*\-\+]\s+(.*?)(?=\n[\*\-\+]|$)', body, re.DOTALL)
deadlines = re.findall(r'(?:by|before|due\s+date)\s+(\w+\s+\d+)', body, re.IGNORECASE)
requests = re.findall(r'(?:can|could|would|please)\s+(.*?)(?=\?)', body)
```

# Code Pipeline

- Load spaCy's English model and other Lib.
- transform the email into correct form and get email body
- Define function to extract place, event, time, date, and subject from text
  - Parse email
  - Parse text with spaCy (Tokenize text)
  - Initialize variables to store information
  - Extract named entities
  - Perform sentiment analysis
  - Extract other important information
- Save the results(DF) to CSV

# Results

```
Sender: phillip.allen@enron.com
Recipient: zimam@enron.com
Subject: FW: fixed forward or other Collar floor gas price terms
Dates: ['3', '5', '7', '10 years', 'May through September']
Times: ['10/16/2000 \n01:42 PM', '10/12/2000 01:12:21 PM', '6-8 hours']
Places: ['San Diego', 'P.E.', 'Albuquerque']
Sentiment: 0.0781641604010025
Action items: []
Deadlines: []
Requests: []
```

# Conclusion

- Presented  different methods for extracting events from Text messages or Emails using Natural Language Processing (NLP) techniques.
- The proposed approach utilizes a combination of Named Entity Recognition (NER) and Dependency Parsing (DP) to identify event triggers and their corresponding arguments.

# THANK YOU!!!