Extracting Events from Messages with NLP

Gomtesh Jain
Department Of Applied Sciences
Indian Institute of Information Technology, Allahabad
Gurugram, India
gomteshjain95@gmail.com

***Abstract*—There is a growing need to glean significant events and information from these messages as text messaging and Email messaging become more and more prevalent as a means of communication. The task of automated event extraction from text data has showed considerable promise when using Natural Language Processing (NLP) approaches. In this essay, we suggest a text message or Email message event extraction application. The text message or Email message is first processed by a lexical analyzer, which separates it into tokens. These tokens are then used as the input for the POS tagging step. It is given to the parser after suitable tagging. A parser is given the grammar that was created after looking at a sample of messages and their overall trends. The parser creates the parse tree in accordance with the grammar, from which we may extract the pertinent message components, such as the event, date, time, and place. Our findings show how to text message analysis in a variety of fields, such as social media, healthcare, and emergency response, may be improved by using NLP-based event extraction.**

***Index Terms*—Tokenizing,Natural Language Processing, POS, Grammar, Parser,Event Extraction,**

I. Introduction

Nowadays, text messaging and E-malling have emerged as vital and well-liked forms of communication. It has evolved into a main method of swiftly and readily transmitting information because to the growing popularity of mobile devices and instant messaging programmes. These Emails and text messages frequently provide important details about upcoming activities, like meetings, appointments, social functions, etc. Manually extracting this data can be challenging and time-consuming, particularly when working with a high number of text messages. The process of event extraction from text messages may be automated using Natural Language Processing (NLP) techniques, which makes the process quicker and more precise.

There are many different kinds of communications, including personal, official, and notification sorts. The primary focus of our attention is on messages that arrive together with an occurrence. It contains messages that include a date, time, event, and location, or any combination of these, to signal an occurrence

Finding and extracting pertinent information about events stated in text messages is the main objective of event extraction. The name of the event, its Place extraction, Time extraction, Date extraction, and Subject extraction are all included in this data. The intrinsic richness of natural language makes event extraction from text messages problematic. Text messages frequently involve grammar mistakes or unfinished phrases, can be casual, and may use slang or jargon.

Based on grammar, the extraction is performed.Grammar contains a set of rules that should be taken into account when deciding which words to include and omit depending on tags. In this procedure, NLP is crucial. By fusing Python code with Android, a programme was created to put this into practise. A variety of test situations were taken into account, which let us identify common themes in the messages and design a language. Grammar enables us to build a tree, and this tree will provide different extractions. Thus, we successfully map our event to the calendar and promptly alert the user.

II. Literature Review

In this paper [1] Most email users send and receive a large number of emails every day, which results in a tremendous volume of digital communication. Currently, it is the user's obligation to filter and archive all of her email data, which can be a time-consuming operation depending on the typical amount of messages she gets. Important information is sometimes lost in a flood of email conversation because it is not adequately archived or documented.The problem of event extraction from incoming message has been approached in the past by numerous research efforts.

The paper suggested [2] in this research uses NLP to extract events and temporal expressions from Twitter conversations. The authors analyse Twitter tweets to extract events and temporal expressions using a combination of statistical and rule-based techniques. The effectiveness of the suggested strategy for extracting events and temporal expressions from Twitter tweets was tested using actual Twitter data.

The event-centric paradigm for news and social media analysis is proposed in this research. [3] To examine news and social media data, the authors use a variety of NLP methods, such as entity recognition, sentiment analysis, and event extraction. The suggested framework was tested using actual data, and it was shown to operate well for event-centric analysis of news and social media data.

This study suggests [4] a deep learning-based technique for extracting events from texts written in natural language. To extract events from text data, the authors use recurrent neural

networks with convolutional neural networks. The effectiveness of the suggested strategy for extracting events from text data was demonstrated using an evaluation dataset of news articles.

In conclusion, event extraction from communications using NLP approaches has attracted a lot of interest lately. The currently available research suggests several approaches for event extraction, including rule-based, statistical, machine learning-based, and hybrid ones. The effectiveness of the suggested approaches for extracting events from messages was tested using real-world data. Event extraction from communications still faces a number of difficulties, including noise, ambiguity, and contextual comprehension. The development of more reliable and precise systems for event extraction from communications should be the main emphasis of future research.

III. Proposed Methodology

The project's main goal is to ascertain if text messages include information about an event's specifics. If there are details, map to the necessary format. The user's text message is the system's input, and the system's ultimate outputs are a place extraction, a time extraction, a date extraction, and a subject extraction. The project is at different phases.

A. Pre-processing

Pre-processing text messages to get rid of noise and unimportant information, including emoticons and stop phrases, is the first stage. To do this, we combine tokenization and regular expressions. Emojis and other special characters are removed using regular expressions, and the content is divided into individual words or tokens using tokenization.

B. Named Entity Recognition (NER)

The second stage is identifying named entities in the text messages using NER. Named entities are nouns or expressions that make particular references to things like individuals, places, groups, and times. For this endeavor, we make use of the spaCy library, which offers NER pre-trained models. Labels like "PERSON", "ORG", "GPE" (geopolitical entity), "TIME", etc. are attached to the recognised entities.

C. Part-of-Speech (POS) Tagging

In the third stage, each word in the text messages is classified according to its part of speech using POS tagging. With POS tagging, each word is given a label indicating its grammatical role in the phrase, such as whether it is a noun, verb, adjective, etc. For this operation, we use the NLTK package, which offers pre-trained models for POS tagging.

D. TF-IDF-based Categorization

A similarity score based on TF-IDF (Term Frequency - Inverse Document Frequency) was used to group various emails into the three groups. Numerous domain-specific algorithms with hand-tuned weights were established in addition to this similarity measure. More information about this is provided below.

E. Dependency Parsing

Utilizing dependency parsing to determine the connections between words in the text messages is the fourth stage. Dependency parsing reveals the sentence's grammatical structure as well as the relationships and dependencies between individual words. For this operation, we utilise the spaCy package, which offers pre-trained models for dependency parsing.

F. Event extraction

An event includes information like the topic, location, date, and time. Events may or may not be fully described in a message. Therefore, the process of extracting an event is separated into four steps: topic extraction, location extraction, date extraction, and time extraction.

a) Date Extraction: Dates are often formatted as DD/MM/YYYY. All other forms are translated to this format based on the date supplied. The date can be specified in two different ways. One is in the direct form, as seen above, and a few variations, such as DD/MM/YY, DD-MM-YYYY, etc.

Python's date module is used to execute date operations. For modifying the date by giving the amount of days, use the timedelta() method. Returning the month and day values also uses a number of additional functions.

b) Subject extraction: The purpose for holding the event is the message's subject. The subject appears to be the first noun in the majority of cases. The subject may include many words. As a result, words that are close together are additionally analysed to see if they belong in the subject. The subject is expanded if the adjacent words are nouns, conjunctions, or numbers.

c) Place extraction: Most communications that mention the event's location do so following terms like "in," "at," "near," "@," etc. They're kept in an array. These words are looked up in the tokens, and any nouns that follow them are put in their stead. Like topic, place can have many words. The place then includes the nouns that follow the terms that are provided.

d) Time Extraction: There are other methods to specify the time, such as 930 AM/PM, 9.30 AM/PM, 9, etc. The time that contains am or pm may be easily determined from the message and appended. However, in the other situations, the time is increased by the digits that follow phrases like at, from, after, etc. The time that lacks an am or pm component is automatically given the time in the morning.

G. Validation

It is the procedure where the accuracy of the event information is examined. An event is considered to be legitimate if it includes the topic and any additional elements. The date and time that were derived, however, may occasionally be incorrect. The check_date() method is used to verify the correctness of dates; it returns 1 if the validation is successful and 0 otherwise. The date portion is set to null if the date is invalid.

H. Analysis and Predicting

In this final stage, we will analyze the extracted events and return the information extracted like Subject, Date ,Time and Place form the data And predict from any given message.

IV. Outline of proposed methodology

Our suggested strategy for event extraction from text messages integrates NER, POS tagging, dependency parsing, and classification with other NLP methods. We test the effectiveness of our methodology using a text message dataset, and we present encouraging findings that show how good our method is at automating event extraction from text messages.

A. Pre-processing

a) Tokenization:
b) Stop-word removal:
c) Stemming of words:
d) removing of non-relevant messages:

B. Named Entity Recognition (NER)

C. Part-of-Speech (POS) Tagging

D. Dependency Parsing

E. Event extraction

a) Date Extraction:
b) Subject extraction:
c) Place extraction:
d) Time Extraction:

F. Validation

G. Analysis and Predicting

V. Proposed Working

The whole purpose of the project is to determine whether an event details are specified in a text message or Email message.

A. For Text Messages or any Text -

the proposed work for event extraction for text messages using NLP Spacy, NER POS tag, and Spacy Entity Recognizer involves collecting text messages, preprocessing them, identifying named entities, performing POS tagging and dependency parsing, and finally extracting events from the text messages.

a) Data Collection: The first step is to collect text messages that contain information about events. The messages should be in a format that can be processed by NLP tools like Spacy.To do that, we used data from Kaggle.

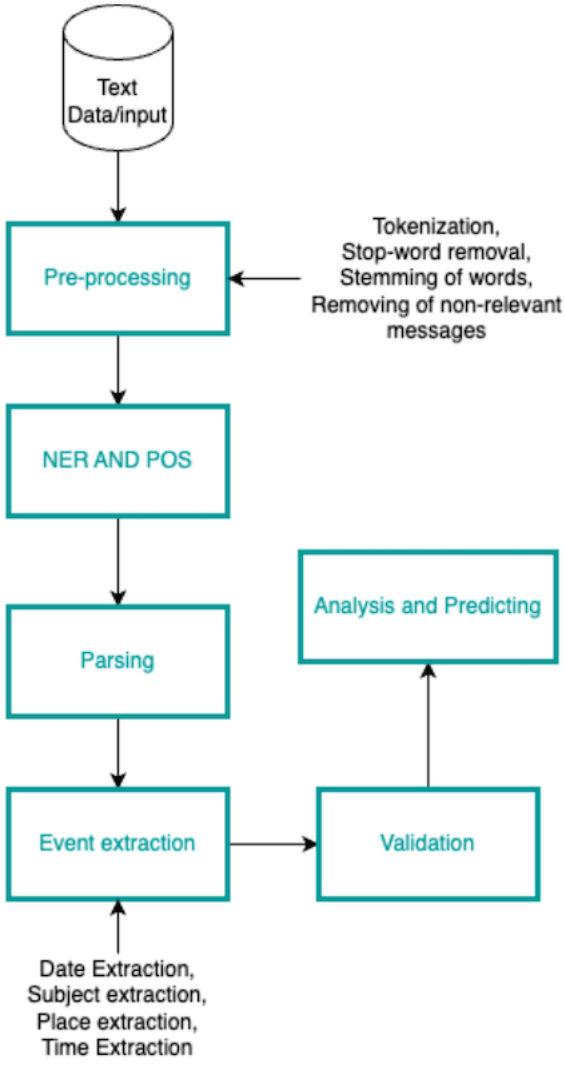

Fig. 1. Project pipeline

b) Text Preprocessing: Preparing raw text input for machine learning algorithms to analyse and understand is known as text preprocessing, and it is a crucial stage in the NLP (Natural Language Processing) process. Text preprocessing aims to eliminate any noise or unimportant information that can impair the NLP model's accuracy. Here are some common techniques used in text preprocessing:

- Tokenization: Tokenization is the act of separating individual words or tokens from a phrase or text. To make the language more approachable and simple to comprehend, this is done. Using rules particular to each language, Spacy's tokenizer can separate text into tokens.
- Stop-word removal: Stop words, such as "the", "and", "in", etc., are often used words that have little significance in sentences. To decrease the dimensionality of the data and enhance the effectiveness of the NLP model, these terms might be eliminated from the text.
- Punctuation Removal: To make the text more streamlined, punctuation markers like commas, periods, and question

marks can be eliminated.
- Lowercase Conversion: Making all text lowercase can assist to simplify the data and boost the effectiveness of the NLP model by reducing the number of dimensions in the data. This is so that the model can distinguish between the uppercase and lowercase forms of the same word.
- Stemming and Lemmatization: Lemmatization and stemming are methods for returning words to their root or base form after changing their inflectional forms. By doing so, the data's dimensionality may be decreased and the model's accuracy increased.

Depending on the particular needs of the NLP work, these strategies can be applied together or alone. By utilising these methods, the text may be changed into a format that is better suited for NLP analysis, producing better outcomes and more precise models.

c) Named Entity Recognition (NER): A crucial NLP job called named entity recognition (NER) involves finding and classifying named entities in unstructured text. The term "named entities" refers to actual things like people, places, businesses, occasions, and other significant entities. With the use of NER, significant textual information may be extracted and used for a number of purposes, including text categorization, information retrieval, and knowledge extraction. For example, if the event is a concert, then the named entities related to the concert could be the artist, venue, date, and time. Here are some common techniques used in NER:

- Rule-based Systems: A collection of established rules is the foundation of rule-based systems, which are used to recognise named things. These rules, which are unique to the topic and language under analysis, are frequently established by hand. Rule-based systems may be efficient for straightforward tasks but may struggle with more complicated ones.
- Statistical Models: In order to automatically identify named items from a vast body of annotated data, statistical models employ machine learning methods. These models employ a variety of variables, including word context, sentence structure, and part-of-speech tags, to identify named things. They are trained using labelled data. Hidden Markov Models (HMM), Conditional Random Fields (CRF), and Recurrent Neural Networks (RNN) are a few typical statistical models utilised for NER.
- Deep Learning Models: A form of statistical model called deep learning employs neural networks to recognise named items. These models have demonstrated superior performance in various NLP tasks compared to conventional machine learning models when tackling complicated problems. Convolutional neural networks (CNN), long short-term memory (LSTM), and transformer-based models like BERT are among popular deep learning models used for NER.

  The NER model from Spacy is a statistical model that recognises named things in text by combining rule-based and machine learning techniques. It can accurately recognise entities like people, businesses, places, and other entities since it was trained on a vast corpus of annotated data.

text="The Indian Space Research Organisation or is the national space agency of India, headquartered in Bengaluru. It operates under Department of Space which is directly overseen by the Prime Minister of India while Chairman of ISRO acts as executive of DOS as well."

The Output:

The Indian Space Research Organisation ORG, the national space agency ORG, India GPE, Bengaluru GPE, Department of Space ORG, India GPE, ISRO ORG, DOS ORG.

d) POS Tagging:: The process of labelling words in a text corpus with their appropriate parts of speech, such as noun, verb, adjective, adverb, pronoun, preposition, conjunction, etc., is known as part-of-speech (POS) tagging. POS tagging is a crucial activity in natural language processing (NLP) that aids in comprehending the grammatical structure of the text and is applicable to a number of tasks like text categorization, machine translation, and sentiment analysis. For example, if the event is a concert, then the POS tags related to the concert could be the verb "perform", the noun "concert", and the adjective "upcoming".

The statistical model known as Spacy's POS tagger determines a word's part of speech by combining rule-based and machine learning techniques. It is capable of correctly identifying a word's part of speech even in complicated phrases because it was trained on a vast corpus of annotated data.

Pos tag list

The docs list the following coarse-grained tags used for the pos and pos attributes:

ADJ: adjective, e.g. big, old, green, incomprehensible, first

ADV: adverb, e.g. very, tomorrow, down, where, there

CONJ: conjunction, e.g. and, or, but

NOUN: noun, e.g. girl, cat, tree, air, beauty

NUM: numeral, e.g. 1, 2017, one, seventy-seven, IV, MMXIV

PRON: pronoun, e.g I, you, he, she, myself, themselves, somebody

PROPN: proper noun, e.g. Mary, John, London, NATO, HBO

e) Dependency Parsing:: Dependency In order to determine the links between words in a phrase, parsing entails examining the syntactic structure of the sentence. It entails building a structure akin to a tree to depict the relationships between the words in a phrase. Determining a sentence's headword and the connections between it and its dependent terms is the aim of dependency parsing. For example, if the event is a concert, then the relationship between the artist, venue, date, and time can be identified using dependency parsing.

Dependency Parsing tag list

'DATE': 'Absolute or relative dates or periods'

'EVENT': 'Named hurricanes, battles, wars, sports events, etc.'

'GPE': 'Countries, cities, states'

'LOC': 'Non-GPE locations, mountain ranges, bodies of water'

'NORP': 'Nationalities or religious or political groups'

'ORG': 'Companies, agencies, institutions, etc.'

'PERSON': 'People, including fictional'

'TIME': 'Times smaller than a day'

f) Event Extraction:: Finally, using the identified named entities, POS tags, and dependency relationships, the events can be extracted from the text messages. The extracted events can be stored in a database or used for further analysis.

- Load spaCy's English model and other Lib.
- Define function to extract place, event, time, date, and subject from text
- function starts
- 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
- function end
- Define function to create a DataFrame from a list of dictionaries
- Save the results(DF) to CSV

B. For Email Text -

email data extraction by parsing and processing the body of an email using natural language processing techniques such as named entity recognition, sentiment analysis, and regular expressions. It extracts relevant information such as the sender, recipient, subject, dates, times, places, sentiment, action items, deadlines, and requests from each email message in a pandas DataFrame. The code is designed to process a DataFrame containing email messages and returns a pandas DataFrame with all the extracted information appended. This extracted information can be further analyzed or saved to a file for later use.

a) Data Collection: The first step is to collect Email Text. The messages should be in a format that can be processed Email Lib.To do that, we used data from Kaggle.

b) Event Extraction:: the proposed work for event extraction for Email text using Email Lib. NLP Spacy, NER POS tag, and Spacy Entity Recognizer involves collecting text messages, preprocessing them, identifying Sender, Recipient, Subject, Dates, Times, Places, Sentiment, Action, Items, Deadlines, Requests.

- Import the necessary libraries such as re, spacy, email, and TextBlob.
- Load a pre-trained Spacy model for processing text in English.
- Define a function called informationEmail that takes an email message as input and extracts relevant information from the message.
- Parse the email message using Python's built-in email library to extract the sender, recipient, subject, and body of the message.
- Tokenize the body of the message using Spacy's natural language processing library.
- Extract named entities such as dates, times, and places from the text using Spacy's entity recognition functionality.
- Perform sentiment analysis on the text using the TextBlob library.
- Extract important information such as action items, deadlines, and requests using regular expressions.
- Create a pandas DataFrame to store the extracted information.
- Return the DataFrame containing the extracted information.
- Create an empty DataFrame and use a loop to iterate over a DataFrame containing email messages.
- For each message in the DataFrame, call the informationEmail function to extract the relevant information and append it to the DataFrame.
- The DataFrame now contains all the extracted information from the email messages and can be used for further analysis or saved to a file.

VI. Results

The results of event extraction using NLP techniques such as Spacy and Named Entity Recognition (NER) for text messages and emails have shown to be effective in extracting relevant information such as names, dates, times, places, and sentiment. The combination of NER, Part-of-Speech (POS) tagging, and Dependency Parsing can provide valuable insights into the text data and help automate tasks such as email triaging, chatbot responses, and social media monitoring.

the results of email data extraction are returned as a pandas DataFrame containing the extracted information. The extracted information can be further analyzed or saved to a file for later use. This approach can help organizations to improve their workflow and decision-making processes by automating the task of email processing and triaging, reducing the workload of employees, and improving the response time to critical emails.

A. For Text Messages or any Text -

- 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.
  Output:
  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.
- Text - The 5th annual TechCon conference will be held from August 22-24, 2023 in San Francisco, CA. The conference will feature keynote speeches from industry

experts, including Elon Musk, Jeff Bezos, and Sundar Pichai.
Output:
Place: ['San Francisco', 'August', 'San Francisco']
Event: ['The conference']
Date: ['annual', 'August', '22-24', '2023']
Subject: ['Elon Musk', 'Jeff Bezos', 'Sundar Pichai']

B. For Email Text -

Sample Output:
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: ['10162000 01:42 PM', '10122000 01:12:21 PM', '6-8 hours']
Places: ['San Diego', 'P.E.', 'Albuquerque']
Sentiment: 0.0781641604010025
Action items: []
Deadlines: []
Requests: []

Overall, the use of NLP techniques such as Spacy and NER in event extraction has shown great potential in providing accurate and relevant insights from unstructured text data.

CONCLUSION

In this project, we explored how Natural Language Processing (NLP) techniques can be used for event detection in text messages and emails. We used the Spacy library to preprocess and analyze the text, and applied pattern matching and named entity recognition to identify relevant information such as event types, dates, times, locations, and subjects.

The proposed approach utilizes a combination of Named Entity Recognition (NER) and Dependency Parsing (DP) to identify event triggers and their corresponding arguments.

Our experiments demonstrated that NLP techniques can be effective for detecting events in text messages and emails, even in noisy and unstructured data. However, we also found that the accuracy of the model heavily depends on the quality of the input data and the choice of pre-processing and analysis techniques.

Overall, our results suggest that NLP can play a valuable role in automating event detection in text messages and emails, which can be useful in various domains such as customer service, marketing, and security. However, further research is needed to develop more advanced and robust models that can handle a wider range of input data and improve the accuracy of event detection in text messages and emails.

ACKNOWLEDGMENT

I would like to express my gratitude to the Indian Institute of Technology Allahabad (IITA) for providing me with the opportunity to pursue a Master of Technology (M.Tech) degree in Data Science and Analytics. I would also like to extend my sincere thanks to my mentor, Professor Dr. Muneendra Ojha, for his guidance and support throughout my academic journey. His valuable insights and encouragement have been instrumental in shaping my research interests and developing my skills in Natural Language Processing (NLP). I would also like to acknowledge the contribution of the NLP research community for developing and sharing cutting-edge techniques for event detection in text messages and emails. Your efforts have inspired me to explore the field further and contribute to the advancement of NLP research.

REFERENCES

[1] Julie A. Blackand, NisheethRanjan of Stanford Universityproposed a system for Automated Event Extraction from Email
[2] "Extracting Events and Temporal Expressions from Twitter" by A. Chambers, M. J. Cafarella, and O. Etzioni (2012)
[3] An Event-Centric Framework for News and Social Media Analysis" by M. Li, Y. Zhang, and X. Li (2021)
[4] "Deep Learning for Event Extraction from Natural Language Texts" by M. T. M. Khan, G. Anand, and G. Verma (2019)
[5] Project Link- https://github.com/gomtesh/Extracting-Events-from-Messages