Event Extraction From Emails Using Natural Language Processing

1st Anand Geed
dept. of Applied Science
Indian Institute of Information Technology Allahabad
Prayagraj, India
IDS2022005@iiita.ac.in

***Abstract*—The project aims to extract relevant information from emails using natural language processing techniques. Specifically, it focuses on identifying and extracting events mentioned in emails, such as meetings, appointments, or deadlines. The system utilizes techniques such as part-of-speech tagging, named entity recognition, and dependency parsing to identify and extract event-related information from the email text. The extracted events are then structured and stored in a database, along with relevant metadata such as the event type, date, time, and location. The project has the potential to streamline email management and increase productivity by automating the process of identifying and tracking important events mentioned in emails.**

***Index Terms*—natural language processing technique, part-of-speech tagging, named entity recognition, dependency parsing**

I. Introduction

Email is one of the most common forms of communication used in today's world, and it has become an essential tool for business communication. However, managing email can be a time-consuming task, particularly when dealing with a large volume of messages. One way to streamline the email management process is to automatically extract important information from emails using natural language processing (NLP) techniques.

In this project, we focus on event extraction from emails using NLP. Specifically, we aim to develop a system that can automatically identify and extract events mentioned in emails, such as meetings, appointments, or deadlines. The extracted events are then structured and stored in a database, making it easier for users to manage and keep track of their schedules.

To achieve this goal, we employ various NLP techniques, including part-of-speech tagging, named entity recognition, and dependency parsing. These techniques allow us to identify and extract relevant information from email text, such as the date, time, location, and type of event. We also use machine learning algorithms to improve the accuracy of the system and enable it to handle a wide range of email formats and styles.

The core objectives of the proposed system include:

- Retrieve event information from emails in a real-time environment.
- Avoid wastage of time by manually opening the mail.
- Eliminate any possibility of human negligence.
- Enable customization capabilities for the information retrieved.
- Developing a user-friendly application.

Overall, the project has the potential to greatly enhance productivity and streamline email management by automating the process of identifying and tracking important events mentioned in emails.

II. Literature review

The problem of event extraction from incoming email has been approached in the past by numerous research efforts. Here we will discuss other proposed solutions to the problem at hand.

One study by Wang et al. (2017) focused on identifying event-related emails from a large corpus of email data. They used a combination of keyword matching and machine learning techniques to identify emails containing information about events, such as meetings and appointments. The study showed that their system was able to achieve high accuracy in identifying relevant emails, which is a crucial step in event extraction

Another study by Stolcke et al. (2019) focused on extracting event-related information from email text using NLP techniques. They used a combination of named entity recognition and syntactic parsing to identify and extract event-related information, such as the date, time, and location of the event. The study showed that their system was able to achieve high accuracy in extracting event-related information, even when dealing with complex email formats.

In 2000, Almgren and Berglund presented an approach for information extraction of seminar data [2]. Their architecture was similar to that proposed here, in that messages are first classified as seminar announcements before information extraction techniques are used to isolate specific information about the seminar such as date, location, speaker, etc. Our system broadens the scope of the event extraction problem and attempts to correctly extract meeting information not only from seminar messages, but also from less structured personal meeting proposals.

Overall, these studies demonstrate the potential of NLP techniques for event extraction from email. However, there is still room for improvement, particularly in handling complex email formats and dealing with noise in the email data.

III. DATASET

Emails are quite private, so email datasets were least available. One of the few available ones was the Enron Dataset. This dataset was collected and prepared by the CALO Project. It contains mails from about 150 users, mainly from the senior management of Enron. The corpus contains a total of about 0.5 million messages. This is the most complete email corpus available. The corpus is one of the few publicly available mass collections of real emails easily available for study, because such collections are typically bound by numerous privacy and legal restrictions which render them prohibitively difficult to access. The existing dataset was insufficient to give a higher performance owing to the following reasons:

- The Enron Dataset was huge, but the number of emails that matched the concerned scenario was very few.
- Event email classification requires emails containing events in order to set a clear demarcation from the other non- event mails.
- The dataset contained mainly informal conversation mails stating unclear events only.
- Using such a dataset would cause overfitting and misclassification.
- Enron dataset is useful for spam mail classification, sentiment analysis etc. and not for Event mail classification.

IV. METHODOLOGY

Email Parsing: Email messages are typically formatted in the MIME (Multipurpose Internet Mail Extensions) standard, which specifies how email messages should be formatted and transmitted over the internet. In order to extract the relevant text information from emails, the emails must first be parsed to extract the different MIME parts, such as the message body and headers, including the subject, sender, recipient, and date. This parsing can be done using existing libraries such as the Python email module.

Text Cleaning: The extracted text may contain unwanted characters, such as special characters, punctuation, or HTML tags, which can affect the performance of the downstream tasks. Hence, it is important to clean the text by removing these unwanted characters and converting the text to a standardized format. This can involve techniques such as regex (regular expression) matching, HTML parsing, and language-specific processing such as stemming or lemmatization.

Text Normalization: Text normalization involves transforming the extracted text into a standardized format to ensure consistency and improve accuracy. This can involve tasks such as converting the text to lowercase, removing stop words (common words such as ”the” and ”is” that do not carry significant meaning), and converting numerical values to a standardized format (such as converting ”3:30 pm” to ”15:30”).

Named Entity Recognition (NER):An NER (Named Entity Recognition) system is used for event details extraction, based on the idea advocated in [7], with lot of customization and improvising. NER is a standard NLP problem which involves detection of named entities from a chunk of text,

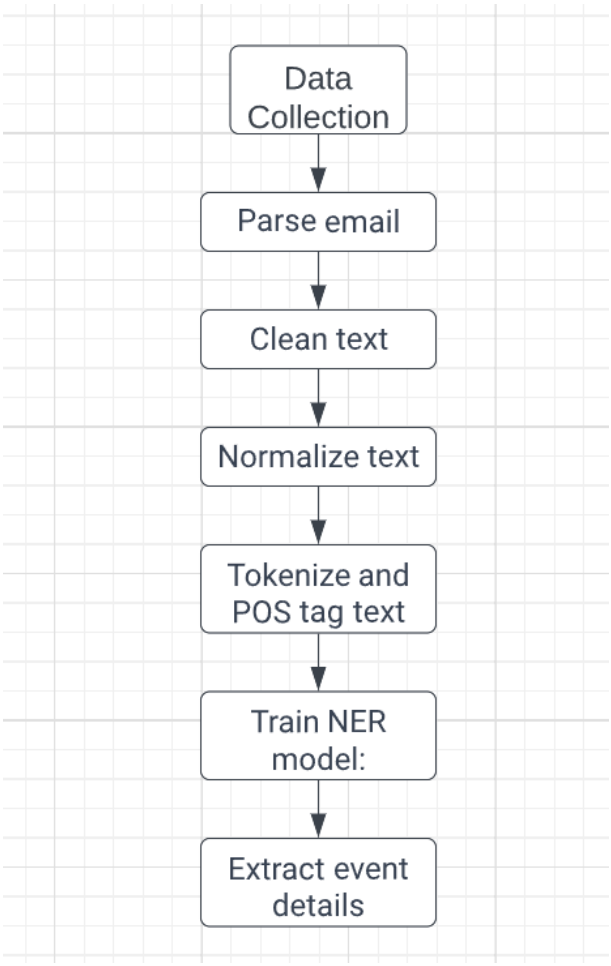

Fig. 1. Flow diagram of the implementation

and classifying them into predefined set of categories. SpaCy is an open-source library that provides an efficient system for NER in python, which can assign labels to groups of tokens which are contiguous. It provides a default model which can recognize a wide range of named or numerical entities, which include person, organization, language, event etc. Apart from these default entities, spaCy allows adding arbitrary classes to the NER model, by training the model to update it with newer trained examples. SpaCy first tokenizes the text into words or word embedding. Words are then split into features and then aggregated to a representative number. This number is then fed to a fully connected neural structure, which makes a classification based on the weight assigned to each feature within the text. The sentences were tokenized into words using the regex tokenizer which avoided the problems of mistokenizing while using the default NLTK tokenizer. Regular expressions were formed for various formats of dates, time etc. and python nltk.tokenize.regexp module was used. Each token was POS tagged using NLTK POS tagger. Since this is a supervised learning problem, the training data should be annotated manually. The dataset consists of the following tags- DATE, TIME, VENUE and LINK. The dataset follows a

BIO type tagging. The BIO format (beginning, inside, outside) is a common tagging format in named-entity recognition

V. RESULTS

Fig. 2. Output of event extraction from email

FUTURE SCOPE

Integration with other NLP techniques: As more businesses and organizations rely on email for communication, there is a growing need for automated systems to help manage the influx of information. Integrating event extraction with sentiment analysis and entity recognition can provide a more comprehensive understanding of the context and purpose of emails. For example, if an email mentions a meeting, event extraction can identify the date, time, and location, while sentiment analysis can identify the tone and mood of the email, and entity recognition can identify the people involved. Combining these techniques can provide more valuable insights into the content of emails and help automate tasks such as scheduling and follow-up.

Specialized domains: In specialized domains such as healthcare and finance, event extraction can be customized to automate tasks such as appointment scheduling, billing, and financial tracking. For example, in healthcare, event extraction can help identify appointment details such as the date, time, and location, as well as relevant medical information such as the reason for the appointment. In finance, event extraction can identify important financial events such as payment due dates, deadlines, and investment opportunities. Tailoring event extraction to these domains can provide more accurate and useful extractions and help streamline tasks for professionals in these fields.

Sophisticated algorithms: As artificial intelligence and machine learning technologies continue to advance, event extraction algorithms can become even more sophisticated. These algorithms can adapt to different email formats and languages and improve accuracy and efficiency over time. This can lead to more accurate and relevant extractions, as well as faster processing times for large volumes of emails.

Intuitive user interfaces: To make event extraction more accessible and user-friendly, developers can create intuitive user interfaces and integrate the technology with other productivity tools such as project management software. This can help users easily manage their events and tasks without having to switch between different applications. By integrating event extraction with other productivity tools, businesses can increase efficiency and productivity, as well as reduce the risk of errors or miscommunications.

CONCLUSION

Extracting events from emails can be a challenging task, as emails often contain unstructured and informal language. However, by using techniques such as natural language processing (NLP) and machine learning (ML), it is possible to automatically extract events from emails with a high degree of accuracy.

The extracted events can be used for a variety of purposes, such as scheduling meetings, tracking project progress, and analyzing customer interactions. Additionally, the extracted events can be integrated with other systems, such as calendars and project management tools, to facilitate efficient workflows.

Overall, event extraction from emails has the potential to streamline communication and improve productivity in various domains, making it a valuable application of NLP and ML technology.

REFERENCES

[1] Rita McCue. “A Comparison of the Accuracy of Support Vector Machine and Naive Bayes Algorithms In Spam Classification,” University of California at Santa Cruz, 2009.

[2] Shashank Senapaty. “Detection and Extraction of Events from Emails,” Department of Computer Science, Stanford University, Stanford CA, December 12, 2008.

[3] Julie A. Black and Nisheeth Ranjan. “Automated Event Extraction from Email,” Stanford University, 2004.

[4] Ola Amayri, Nizar Bouguila. “A Study of Spam Filtering Using Support Vector Machines,” Artif. Intell. Rev.. 34. 73-108. 10.1007/s10462-010-9166-x, 2010.

[5] Aneesh G. Nath, Krishnanth V, Kevin Biju Mathew, Pranav T S, Sarath Gopi. “NLP based Event Extraction from Text Messages,” International Conference on Future Technology in Engineering, 2016.

[6] Aswar Shreyas, Gaikwad Priyanka, Merlyn Pearl, Shinde Swapnal. “Event information extraction from email and updating event in calendar,” Vol-4 Issue-3 2018, IJARIIEISSN(O)-2395-4396.

[7] Eleni Partalidou, Eleftherios Spyromitros-Xioufis, Stavros Doropoulos, Stavros Vologiannidis, Konstantinos I. Diamantaras. “Design and implementation of an opensource Greek POS Tagger and Entity Recognizer using spaCy,” IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2019.

[8] Konstantinos N. Vavliakis, Andreas L. Symeonidis, Pericles A. M , “Event identification in web social media through named entity recognition and topic modelling”, in K.N. Vavliakis et al. / Data Knowledge Engineering 88 (2013).

[9] Hao Li, Heng Ji, Lin Zhao, “Social Event Extraction: Task, Challenges and Techniques”, in IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2015.

[10] Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, Franciska de Jong, Emiel Caron, “A Survey of event extraction methods from text for decision support systems”, in F. Hogenboom, et al. / Decision Support Systems 85 (2016).

[11] Wenjuan Cui, Pengfei Wang, Yi Du, Xin Chen, Danhuai Guo, Jianhui Li, Yuanchun Zhou, “An algorithm for event detection based on social media data”, in W. Cui et al. / Neurocomputing 254 (2017).

[12] Jingsheng Deng, Fengcai Qiao, Hongying Li, Xin Zhang, Hui Wang, “An Overview of Event Extraction from Twitter”, in International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, 2015.