EVENT INFORMATION EXTRACTION FROM E-MAIL AND UPDATING EVENT IN CALENDAR

Aswar Shreyas[1], Gaikwad Priyanka[1], Merlyn Pearl[1], Shinde Swapnal[1]

[1] *Students, B.E., Dept. of Computer Engineering, D.Y.Patil College Of Engineering, Akurdi, Pune, Maharashtra, India*

ABSTRACT

Digitization has sneaked into every aspect of our life. These days most of the people receive a very large number of emails. Emails are also used to send information about events and invitations. People add these events, invitations to a calendar or reminder like app manually. But checking the email from email inbox and adding the event details manually to a calendar or reminder app is bit time taking. This paper, presents a NLP based web browser extension which can detect events and extract them to add into a calendar or reminder autonomously with user consent. The email text goes through a lexical analyzer and it splits the email text into tokens. These tokens are taken as input by the tagging phase. Then a parser constructs parse tree, using a specific grammar, which gives us event details such as time, place, date etc. After extraction, these event details can be mapped into a calendar if the user wants.

Keywords : - *Event Extraction, NLP, Grammar, NLTK, Tokens, API, Extension.*

1. INTRODUCTION

In this present digitalizing world, every internet user is receiving a large number of E-mails. Emails are a much easier and faster way to communicate than the traditional mail. Common people receive a lot of text messages or emails as a notification for them about certain events and generally these people do add the events to calendar or a reminder like application in order to notify as if the numbers of such messages are large. People use various things like keeping a note of the events in a diary, adding the event to the calendar or reminder like app manually so that it will notify on the day even if they forget about it. Adding to calendar makes alerts on the mobile on or before the event and meeting date.

As the world is so hectic, manually reading the mail and searching for event and then adding data to a calendar requires a lot of time. E-mails can be of many types: Personal messages, community events, official notification type etc. The main domain of our concern is the email messages which arrive with an event in it. It includes messages having a date, time, event, and place or either of these present which indicates an event.

Our paper aims at automating the process of extraction of event and adding it on to the calendar. Using this system, the manual efforts of opening mail, reading it, understanding the event information, and manually making calendar entries will be eliminated. By making use of natural language processing we can classify the event details in the arriving email into subject, date, time and place if any of the combination is present. We take the event details and maps on to the calendar with the appropriate date provided in the message. Thus the crucial part is extracting the subject, date, time and place.

The extraction is mainly grammar based. Natural language processing plays a very important role in the extraction process. A grammar includes certain rules which are to be considered to include and exclude words based on the tags. An extension was developed in order to implement this by integrating python code. In this system, once the user logs in with his/her login credentials, the emails will get retrieved one by one and the data will be analyzed and event will get extracted. We've performed a number of observations to find the common pattern in the emails to define the grammar. Grammar helps us to construct a tree and this tree will result into various extractions. Thus we get our event mapped on to the calendar and notify the user at right time.

2. LITERATURE REVIEW

The numerous research efforts has been made to resolve the issue of event extraction from the given plain text data, from social websites, blogs, huge web documents and Emails. On a daily basis, most internet users will send and receive numerous emails of varying types thereby generating massive amounts of digital communication. At present, it is the user's task to filter and archive all of his email data which, depending upon the average number of messages he receives, may be an extremely time consuming task. Often important information is not properly archived or noted and, as a result, lost in a sea of email communication.

In [1] authors have focused on extracting events from the Social media sites like Facebook and Twitter. They are the first to present formal definition of social events. They figured the tasks and challenges in implementing such sys- tem. For the same they proposed mainly two solutions for extracting the event information from social media site : (1) an unsupervised content segmentation framework to extract event phrases (2) utilize external knowledge bases to detect fine-grained event locations and unveil their background information.

Over the years, Information Extraction (IE) has become increasingly popular as a tool for a vast array of applications. In [2] authors have surveyed different methods for event extraction from text for decision support systems. They gave summarization of event extraction techniques for textual data, distinguishing between data-driven, knowledge-driven, and hybrid methods, and presented a qualitative evaluation of these.

In [3] authors have defined the concept of important event and proposed an efficient methodology for performing event detection from large time-stamped web document streams. They also proposed an efficient algorithm for detecting all important events from a document stream. The algorithm when tested on 7 million blog posts, accurately detected events, represented the detected events in a rich format.

A noticeable work has been done in [4], in which the authors have presented a mechanism to ex- tract the events related to Foodborne disease in nearby areas by using algorithms like fixed and dynamic context window selecting. They used Weibo API to crawl the tweets and get the keywords and developed SVM (support vector machine) to reduce the noisy data.

3. SYSTEM ARCHITECHTURE

The main aspect is to determine if an event is specified in a text message or email. If events details are present, then map the details to calendar by extracting them from the message. The input for the system is the email received by the user on his/her mail account and the final output is a calendar event. There are different stages of the project as shown in the figure.

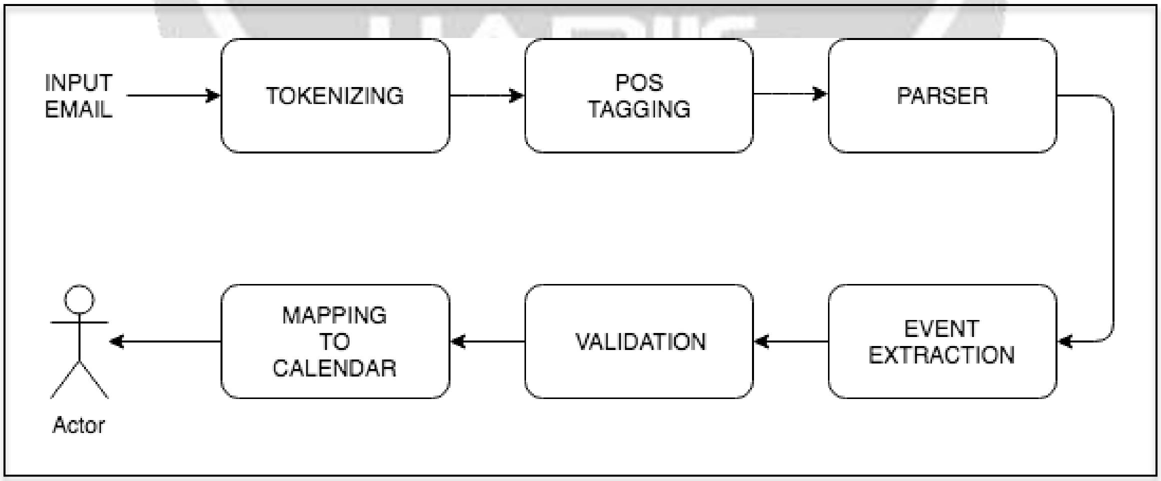

Fig.1 – System Architecture Pipeline

3.1 Retrieving Email Contents

At client side, after enabling the extension, user will be asked to login to Gmail account if not already logged in and will ask for permissions. If already logged in, then it will retrieve authentication details stored in JSON file i.e. storage.json. The client id and scope is defined to get access to Gmail API. Then it will scan the inbox and retrieve unread emails and send them to server.

3.2 Tokenizing

At server side-The email obtained by the user s stored as a string and it contains a group of words. The string is to be split into tokens. For doing this we use a word_tokenize function of NLTK library nltk.tokenize. The word_tokenize function takes the string as the input and separates each words as tokens by checking the whitespace and periods. Each of the words are stored in an array. The array is returned by the function. These tokens thus obtained are passed to POS tagger. The tokens obtained contains words, brackets. Normal split() function separates the words based on white space only.

3.3 POS – Tagging

Noun, verb, propositions, etc. are the different part of speech tagging. Different part of speech tagging should be introduced in addition to the common POS tagging present in the English grammar for making date and time processing easy. For adding the additional POS tagging to the grammar we have defined a list of regular expression as re. The list of POS tagging is shown in the table.

Table 1- : POS Tagging

SAL	Salutation
EX	Expression
PUN	Punctuation
DATESPEC	Date Specified
DATEUNS	Date Unspecified
CON	Conjunction
DT	Determinant
VB	Verb
MD	Modal
DAYS	Day
MONTH	Month
TM	Time
NNS	Noun
IN	Proposition

An example is shown below.

Text message :
‘there will be a meeting today’
After POS Tagging :
[(‘there’,’EX’),(‘will’,’VB’),(‘be’,’VB’),(‘a’,’DT’),(‘meeting’,’NN’),(‘today’,’DATESPEC’)]

3.4 Parser

Word which are obtained are after POS tagging may not be contain event information. These words are to be eliminated. Grammar is defined so as to eliminate the words which are of no use. This module offers a generic date/time string parser which is able to parse most known formats to represent a date and/or time. ‘dparser’ module attempts to be forgiving with regards to unlikely input formats, returning a date time object even for dates which are ambiguous.

3.5 Event Extraction

An event contains details like subject, place, date and time. A message containing event may or may not contain all the details. So the event extraction process is divided into four parts namely subject extraction, place extraction, date extraction and time extraction.

3.5.1 Subject extraction

Subject of the message is the reason for which the event is held. In most of the cases the subject of the email and the subject of event is same. Hence we need to extract the email subject only.

3.5.2 Location extraction

Most of the messages the location of the event is specified comes after words like in, at, near, @,etc. These are stored in an array. The tokens are checked for these word and the noun appearing after these words are added to place. As is the case of subject, place can be of multiword. So the adjacent nouns appearing after the words specified are added to the place.

3.5.3 Date Extraction

The standard format used for date is DD/MM/YYYY. Date specified all the other forms are converted to this format. There are two ways to specify the date. One is in the direct form as shown above and some variants of it like DD/MM/YY, DD-MM-YYYY, DD-MM-YY, etc. The other ways are like today, tomorrow, this Sunday, next Friday, 1st of September, etc. for each of these case different method should be applied. For example for the case were “this Saturday” is specified we have to get value of the present day like for Sunday is 0, Monday is 1, so on. Then we have to get the difference in value of the day specified in the message and present day and add that difference to present date. For the case of “next Friday” we have to do the same procedure and then add 7 to the date. For the case of “1st September” we have to replace the date and month field of present date with the date specified in the message.

3.5.4 Time Extraction

Time can also be specified in many ways like 830 am/pm,8.30 AM/PM,8, etc. time which have am or pm can be directly identified from the message and can be added to the time. But for the other cases the digits appearing after words like at, from, after, etc. are added to time. By default am is given to the time which doesn’t have the am or pm part.

3.6 Mapping to Calendar

The extracted event is then added to Google calendar. To do so Google Calendar API is used. The client id created by project owner, Authentication credentials stored in JSON file created at login stage and the scope is defined to gain access to the calendar. The event details retrieved such as event name, date, location, time etc are stored in JSON format and are inserted into calendar with reminder on event day using insert function of API service.

4. EXPERIMENTAL RESULTS

The main goal of the proposed work is to develop an extension application in order to automate the process of event extraction from text messages and add them on to the calendar. We considered a large set of messages and from each we extracted the event date place and time if its present in that. Consider the following messages and the extraction process.

CASE 1 : When all details are specified.

I] The committee P, coordinators meeting will be held at 3.20pm today in conference hall. Inconvenience is regretted.

[('The', 'NN'), (committee, 'NN'), ('P', NN), ('coordinators', 'NN'), ('meeting', 'NN'), ('will', 'VB'), ('be', 'VB'), ('held', 'VB'), ('at', 'IN'), ('3.20pm', 'TM'), ('today', 'DATESPEC'), ('in', 'IN'), ('conference', 'NN'), ('hall', 'NN'), ('.', 'PUN'), ('Inconvenience', 'NN'), ('is', 'VB')]
(NP The/NN committee/NN P/TM coordinators/NN meeting/NN will/VB be/VB held/VB at/IN 4.10pm/TM today/DATESPEC in/IN conference/NN hall/NN ./PUN Inconvenience/NN is/VB)
Event: meeting,meet,see
Time : 3.20PM
Date: 28/5/2018
Loc: not found

II] Dear sir, a meeting of Research Council is scheduled from 12 PM to 1.20PM on 14th April in the Conference hall. All members are requested to attend. XYZ , HOD , ABCOLLEGE

[('Dear', 'SAL'), ('sir', 'SAL'), (',', 'NN'), ('A', 'NN'), ('meeting', 'NN'), ('of', 'IN'), ('Research', 'NN'), ('Council', 'NN'), ('is', 'VB'), ('scheduled', 'VB'), ('from', 'IN'), (12, 'TM'), ('PM', 'NN'), ('to', 'IN'), ('1.20PM', 'TM'), ('on', 'IN'), ('14th', 'DATEUNS'), (April, 'MONTH'), ('in', 'IN'), ('the', 'DT'), ('', 'NN'), ('Conference', 'NN'), ('hall', 'NN'), ('.', 'PUN'), ('All', 'NN'), ('members', 'NN'), ('are', 'NN'), ('requested', 'VB'), ('to', 'IN'), ('attend', 'NN'), ('.', 'PUN'),(XYZ, 'NN'), (',', 'NN'), (HOD, 'NN'), (',', 'NN'), ('ABCOLLEGE', 'NN')]
(NP,/NN A/NN meeting/NNof/INResearch/NN Council/NNis/VBscheduled/VBfrom/IN
12/TMPM/NNto/IN1.20PM/TMon/IN 14th/DATEUNSApril/MONTHin/INthe/DT/NNConference/NNhall/NN
./PUN All/NNmembers/NNare/NNrequested/VBto/INattend./NN XYZ/NN ,/NN HOD/NN ,/NN
ABCOLLEGE/NN),
Event :meetiing,meet,see
Date : 14/4/2018
Loc : error
Time : 12 PM-1.20PM

CASE 2 : When some details are not present.

I] All Infosys selects please be present at the Shantai hall by 10:00 am.

[('All', 'NN'), (Infosys, 'NN'), ('selects', 'NN'), ('please', 'VB'), ('be', 'VB'), ('present', 'NN'), ('at', 'IN'), ('the', 'DT'), ('Shantai', 'NN'), ('hall', 'NN'), ('by', 'IN'), ('10:00am', 'TM'), ('.', 'PUN')]
(NP All/NNInfosys/NNselects/NN please/VBbe/VBpresent/NNat/INthe/DTShantai/NN hall/NN
by/IN10:00am/TM./PUN)
Event : not found
Date : 28/05/2018
Loc :Shantai hall
Time : 10.00AM

II] There will be a meeting on 21-09-2018 at Dnyanprasad hall.

[('There', 'EX'), ('will', 'VB'), ('be', 'VB'), ('a', 'DT'), ('meeting', 'NN'), ('on', 'IN'), ('21-09-2018', 'DATESPEC'), ('in', 'IN'), ('Dnyanprasad', 'NN'), ('hall', 'NN'))]
(NPwill/VBbe/VBa/DTmeeting/NNon/IN21-09-2018/DATESPEC in/INDnyanprasad/NN hall/NN)
Event: meeting,meet,see
Date:21/9/2018
Loc:Dnyanprasad hall
Time: not found

CASE 3 : Personal messages.

I] Lets meet today.

[('lets', 'NNS'), ('meet', 'NN'), ('today', 'DATESPEC'), ('.', 'PUN')] (NP lets/NNS meet/NN today/DATESPEC ./PUN)
Event : meet,see
Date : 28/05/2018

Loc :not found
Time : not found

CASE 4 : Message with incorrect output.

I] My sister's engagement is on 4th April and wedding on 6th may. You and your family are invited.

[('My', 'NN'), ("sister's", 'NN'), ('engagement', 'NN'), ('is', 'VB'), ('on', 'IN'), ('4th', 'DATEUNS'), ('April', 'MONTH'), ('and', 'CON'), ('wedding', 'NN'), ('on', 'IN'), ('6th', 'DATEUNS'), ('may', 'MONTH'), ('.', 'PUN'), ('You', 'NN'), ('and', 'CON'), ('your', 'NN'), ('family', 'NN'), ('are', 'NN'), ('invited.', 'IN')]
(NP My/NN brother's/NN engagement/NN is/VB on/IN 4th/DATEUNS April/MONTH and/CON wedding/NN on/IN 6th/DATEUNS may/MONTH ./PUN You/NN and/CON your/NN family/NN are/NN invited./IN)
Event : engagement,wedding,marriage
Date : 4/4/2018
Loc: not found
Time: not found

Table 2- : Test-cases results

TESTS CASES	ACCURACY
All details present	28 of 30
Some details are not present	27 of 30
Incorrect input	9 of 10
Personal or informal messages	15 of 20

5. CONCLUSION

With this extension system for a browser the time consumed for manually logging to a email service provider, reading the invitation and setting an reminder for it is decreased. The extension can automatically provide this functionality. In future we can extend support to different browsers, also we can modify the basic extraction system which can be used extract different information from emails or any website.

7. REFERENCES

[1]. 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.
[2]. 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).
[3]. 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).
[4]. 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.
[5]. 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).