International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 03 | Mar 2020 www.irjet.net p-ISSN: 2395-0072

Event Notifier on Scraped Mails using NLP

Ashish Sangale[1], Mrunal Nalawade[2], Ameya Brahmankar[3], Shruti Agrawal[4]

[1,2,3]Student, Dept. of Information Technology, Vidyalankar Institute of Technology, Mumbai
[4]Professor, Dept. of Information Technology, Vidyalankar Institute of Technology, Mumbai

---

Abstract - *In today's hectic schedule one often tends to miss important event or meeting because a reminder on calendar went un-noticed. The system is designed by keeping in mind the efforts and time taken to continuously browse the mails. This system aims at notifying the user about any event received on their mail by setting a reminder on google calendar. The event can be an appointment, movie booking, meeting, keynote event and anything else. This is done by integrating Gmail API using which mail is scraped. From this mail, event related data is extracted using NLP and stored onto a csv file. Further, Calendar API is integrated, and reminder is set on the calendar using the data from csv file. Google itself filters any spam mails, so the user need not worry about the reminders of fake events. Details about the event will be set to the calendar which will make attending the event easier for the user as they will get the event related information at one place.*

Key Words: Gmail API, Natural language processing, OAuth, Chunking, POS tagging, Google Calendar, IOB tagging.

1. INTRODUCTION

The Number of mails received per day are quite high and it is tedious task to go through each mail. Event notifier will do the needful. Firstly, it will check for new mails after specified time. When it finds one, it will scrape the data from the mails. Further it will recognize the event and other event related details. It will extract subject of mail, sender's email id and message snippet. From message snippet, time and address are retrieved. For retrieving address, message is divided into chunks, which are further assigned with IOB tags. Using this tag, address is identified. This extracted information will be saved to file. From this information, it will set a reminder for that event. The extracted information will also contain the address of the event, which will also get saved to the reminder on the Google calendar and on one click the user can even get directions to the event. All the scraping of data is done after a specific time interval that is after a specific time interval it will check for any new mail and scrap only those new mails and not all, so the problem of huge data getting scrapped is also solved.

2. Literature Survey

2.1 Web Scraping

Data proves to be very useful in the field of marketing, scientific or academic research and even for data analysis. Collecting data directly from website is a very tedious task and to deal with such limitation web scraping techniques and tools are introduced. With the use of these tools and techniques extracting information has become much easier. Now the data can be easily visualized and analyzed for further use. Web Scraping has been very useful in the proposed system for scraping mails in .csv file. Web scraping is a subset of web mining technology. Web mining is at the intersection of Information Retrieval and Information Extraction. Both have important roles to trace and mine valuable information out of unstructured data. When talking about Web Scraping, it is mining information from different and unstructured websites and transforming it into comprehensible structure like spreadsheets, database or .csv file. Various types of data like house pricing, stock pricing, sports scores, reports etc. can be collected using web scraping. From operation point of view web scraping is like copy paste operation just the difference is that the job is done in an organized way by the computer itself. Web scraping software are being developed which are just a computerized program to do the manual copy paste work. Large amount of data is being collected at a click and stored in a local database or a spreadsheet for analysis [1]. Among the various tools available for scraping Scrapy is the tool used in the proposed system.

2.2 Natural Language Processing (NLP) using NLTK

The term Natural language processing of computer science, artificial intelligence and computational linguistic focuses on how the interaction between computers and human languages takes place. NLP also includes the ability to draw useful insights from data obtained from mails, videos and other unstructured material. Various aspects of NLP include Parsing, Machine translation, Language modelling, Machine learning, Semantics Analysis. NLP is the use of the system and its capability to process sentences in a natural language rather than specialized artificial computer language such as python, java.

NLTK is a collection of program modules, data sets, tutorials and exercises, covering symbolic and statistical

natural language processing. NLTK is based on python and distributed under the GPL open Source license. NLTK is a collection of various modules namely Parsing module, Tagging module, Finite State Automata, Type checking, Visualization, Text classification, etc. Core modules define basic data types that are used, and the remaining modules are just the devoted to specific task. NLTK uses several text corpora. A text corpus is a large collection of structured text [3].

Following are the Corpus that are being used in the proposed system

Brown Corpus – This corpus consists of one million words of American English. This was the first Corpus that could be used in computational linguistic processing.

Gutenberg Corpus – Gutenberg corpus consists of 14 texts chosen from Project Gutenberg which is largest collection of e-books. This corpus contains total 1.7 million words.

Stopwords Corpus – Apart from the regular words we use there are certain words which perform important grammatical functions but are unlikely to be interesting. These words are called as Stop words. NLTK consist of 2400 stop words across 11 different languages.

3. Methodology

3.1 Integrating Gmail API

Gmail API is the latest and still underutilized public API space, which provides RESTful access to a user's Inbox messages, Inbox configuration, message labels, and the ability to draft and send messages on a user's behalf. This toolkit makes it possible to implement a full web-client that replicates functionality as of Gmail.

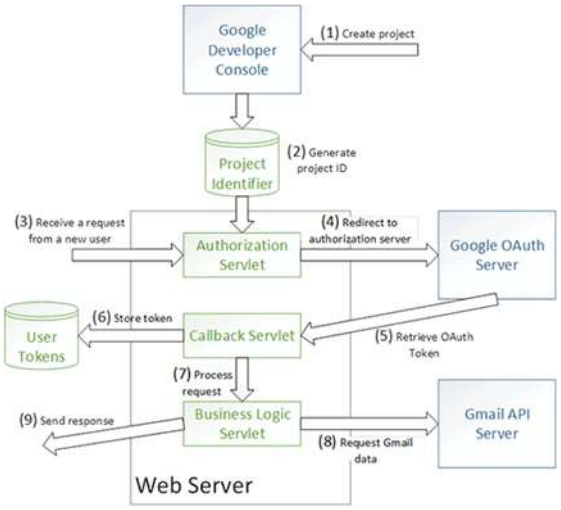

Fig -1: Integrating Gmail API [6]

3.2 Natural Language Processing

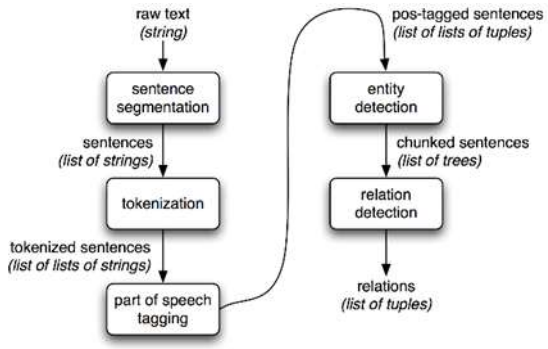

Fig -2: Natural Language processing

When using a semi-structured text where you can match some "labels" where address begins, then regular expression is a way to go. This process is fast and does not need a huge dataset of addresses to train address chunker, only predefined regular expression is required and then use it wherever it fails.

So, as regular expression is off the table, we can use Natural Language Processing to process text and extract addresses. As a classifier NLP can predict the class of chunk of text based on the previous observation.

An HMM tag is used to assign POS tags. This is done by finding out the mostly tag for each word in a sentence. HMM finds a tag sequence instead of finding tag for each separate word. In a given sentence w1,…, wn, a HMM based tagger chooses a tag sequence t1,….,tn that maximizes the following joint probability:

(t1 … tn, w1…., w) = P(t1 … tn)P(w1 … wn|t1..tn)

Maximum Entropy based taggers incorporate complex features which are unlikely to be done by the unigram and HMM based taggers. Conditional probability of Maximum entropy-based taggers can be given by

$$(w1 \dots wn|t1 \dots tn) \approx \prod^{P(t1 \dots tn)P(ti \mid Ci)Ni=1}$$

Where C1,...,Cn . C is the context of the word that contains the previous assigned tags before w.

3.3 Integrating Google Calendar API

Google uses OAuth2.0 for authorization. Users are redirected to Google OAuth Login URL where they authorize the application to manage their Calendar. After the user authorizes the Google Application, user will be redirected to a given redirect URL. An authorization code is passed as a GET parameter named code by Google. We must use this code and make an API call to get an access token. Using this token, we can set an event on google calendar.

Fig 3: Flowchart of Proposed System

4. Proposed Model

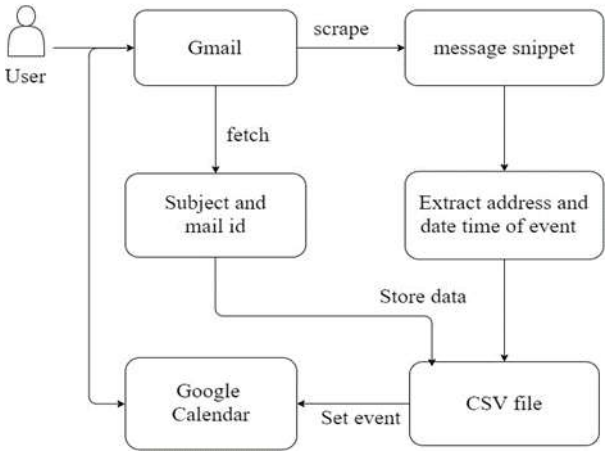

Fig 4: Block diagram of proposed system

4.1 Integrating Gmail API

Creating a new Gmail API project needs logging into the Google Developer Console using an existing Google account, choosing the option to create a new project, and selecting Gmail API as one of the APIs used in the project. Once the User is correctly logged in and new API is selected and used in the project, Google's monitoring systems will make sure to relate your application's API accesses to the project. To handle project identification technically, the Developer Console lets you create a file with all the project information. This file must be loaded into a GoogleClientSecrets object at runtime and pass the object when invoking the API methods. It is to be note that the project credentials are tied to a specific domain, so the developers need to get a different file for each domain they intend to support.

4.2 Accessing the mails

Initially to invoke the API, it is necessary to create an instance of the Gmail API client. Using the client, it is possible to access the user's message threads, individual messages, Inbox labels, Gmail history, draft messages, and message attachments. All the resources are made available by executing appropriate commands on clientObject.users().resourceType() object (where clientObject and resourceType are placeholders).Once an API client is created, the API client code calls a method that lists down threads for a particular user (service.users().threads().list(USERNAME)), and then iteratively invokes a method to get individual threads (service.users().threads().get(USERNAME, thread.getId())). These incoming requests are passed to the Gmail API servers as HTTP GET and HTTP POST requests (For example, GET https://www.googleapis.com/gmail/v1/users/userId/threads/list retrieves the list of threads). The returned

objects are data structures with several complex fields. Consider as an example, each thread has a set of messages, while each message has an associated body, set of labels, unique thread and message identifiers, etc. Our callback takes the labels of a first message in each thread and compares the labels to a designated label.

4.3 Extracting address from text

To train the classifier a dataset of tagged sentences is needed. This Dataset must be in IOB format. The sentences must be tagged in a way such that they show classifier where address begins, continue and ends.

It creates a list containing number of tuples where the first element is a word and second is IOB tag:

O - Outside of address.

B-GPE - Begin of address string.

I-GPE - Inside address string.

The Probability P of lexical part of speech cannot be taken into account any contextual influences. Input representation of current tagged word is taken into consideration

Inij = (posj | wordi) , if i ≥ 0

There is lot of information of the preceding words, as they have been already tagged. Here the activation values are instead of lexical part of the speech probabilities:

Ini(t) = outj(t + 1), if i ≥ 0

4.4 Extracting Date from text

Datefinder is python module for locating dates inside text. This package is used to extract all sorts of date like strings from a document and turn them into datetime objects. This module finds the similar datetime strings and then uses the dateparser package to convert them to the datetime object.[8]

4.5 Creating event on Calendar

To create an event on the calendar, call the events.insert() method providing at least these parameters:

1. calendarId parameter is the identifier for calendar and can either be the email address of the calendar on which to create the event or a special keyword 'primary' which will use the calendar of the logged in user.

2. event parameter is the event to create with all the necessary details such as start and end. The start time and the end time are the only two required fields. Timed events are specified using the start.dateTime and end.dateTime fields. For all-day events, start.date and end.date start.date and end.date methods are used instead.

3. Set your OAuth scope.

5. Results

Table -1: Results

Mail	Extracted Information	Event set on calendar
Your appointment for dental check-up is on January 23rd, 2020 at 5:00 pm at Dr.Adsul, Bhatia Hospital, Gokhale road, Bandra.	Sender' email: dentalcare@gmail.com Subject: Dental Check-up Location: Bhatia Hospital, Gokhale road, Bandra Date and time: 23.01.2020 at 17:00	Date: 23.01.2020 Time: 17:00 Location: Bhatia Hospital, Gokhale road, Bandra Event: Dental Check-up Mail sent by: Dental Care
Your Yoga Camp has been booked for 10th May 2020 from 9:00 am onwards at Wellness Fitness Hub, Goregaon	Sender's email: wellnessfirnesshub@hotmail.com Subject: Yoga camp Location: Wellness Fitness Hub, Goregaon Date and time: 10.05.2020 at 09:00	Date: 10.05.2020 Time: 09:00 Location: Wellness Fitness Hub, Goregaon Event: Yoga camp Mail sent by: Wellness Fitness Hub
Your have been invited to creators meetup on Friday 28th February at	Sender's mail: youtubein@gmail.com Subject: Creators Meetup Location: St.Regis Lower Parel Date and Time: 28/03/2020 at 17:00	Date: 28.03.2020 Time: 5:00 Location: St.Regis

5:00 pm The event will be held at St.Regis Lower Parel		Lower Parel Event: Creators Meetup Mail sent by: Youtube India

6. CONCLUSIONS

The proposed system showcases ability of scraped data to set reminders on calendar. The Paper demonstrates how the Gmail API, Calendar API can be integrated into the proposed system. It covers the background activities in Gmail API and Calendar API that have enabled so much functionality.

This paper also demonstrates the use of OAuth 2.0 protocol for authentication and authorization purpose. As a future scope the proposed system can contribute to any booking applications or websites.

REFERENCES

[1] Prof. Anand Saukar, Prof. Kedar Pathare, Prof. Shweta Gode, 2018, 'An overview of web scraping techniques and tools', International Journal on Future Revolution in Computer Science & Communication Engineering Vol. 4 (4)

[2] Alabhya Farkiya, Prashant Saini, Shubham Sinha, Prof. Sharmishta Desai, 'Natural Language Processing using NLTK and WordNet', International journal of Computer Science and Information Technologies Vol. 6(6)

[3] Paul Nelson, Part of "Cruising the Data Ocean" series, NLP techniques to extract information.

[4] Edward Loper, Stewen Bird, ResearchGate, July 2002, NLTK (Natural Language Toolkit).

[5] Pratiksha Ashiwal, S.R.Tandan, Priyanka Tripathi, Rohit Miri, IJRASET June 2016, Web Information retrieval using Python and Beautiful soup.

[6] Gmail API Documentation (https://developers.google.com/gmail/api/guides)

[7] Google Calendar API Documentation (https://developers.google.com/calendar/overview)

[8] https://datefinder.readthedocs.io/