Food Truck Locator

Shaozhuo Jia
College of Engineering
San Jose State University
San Jose, USA
shaozhuo.jia@sjsu.edu

Phani Teja Kantamneni
College of Engineering
San Jose State University
San Jose, USA
phaniteja.kantamneni@sjsu.edu

Amey Totawar
College of Engineering
San Jose State University
San Jose, USA
amey.totawar@sjsu.edu

Rajeshri Dattatraya Sonwane
College of Engineering
San Jose State University
San Jose, USA
rajeshridattatraya.sonwane@sjsu.edu

***Abstract*—A natural language processor that specializes in creating events for food truck vendors automatically and consolidate all the events at one place with concise, but refined information about that particular vendor. Ultimately, allowing the vendor to publicize his events and the user to enjoy. The Google API based map embedded in a web application points to the markers extracted from the vendor tweets as well as custom vendor events to pop up the events in the nearby area.**

***Index Terms*—Natural Language Processing, Twitter API, Google Maps API**

I. Introduction

The main purpose of the project is to create a platform that will generate google maps of the Food trucks for the user. This project will gather and analyze data from two major sources to provide a map of local food trucks events for everyone to browse. We are fetching Twitter's new feeds and then processing it with the help of Natural Language processing. We are storing that data i.e. Location, Time, Twitter handler's name, Original tweet, etc into the database. We have used Amazon's instance for storing our MySQL database.

Also, food truck vendor can create their event in our platform by login with their twitter credentials into our system. Vendor can add, view, edit and list all the events that belongs to it. The user will get to see all the location/markers on maps for the Food truck once he/she visit our website. User is able to filter the events based on the location, date, radius, etc.

II. Methodology

This system will display food truck events on a map for all users to browse by fetching its database.

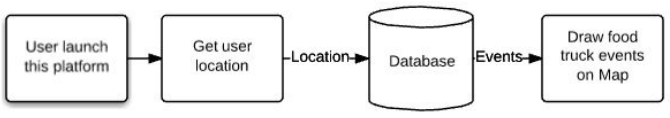

Fig. 1. System overview

1. User launch this platform
2. The platform asks for user's location
3. The location information is used to fetch events from the database
4. Queried events are draw on a map where user can interact

This system will be gathering data from two major data sources: A. Food truck vendor's twitter feed; B. Food truck vendor create event on this platform

A. *Food truck vendor's twitter feed:*

Whenever a new post was created on twitter by a food truck vendor, a subsystem connected to that twitter stream is notified, the following actions will take place to process and evaluate the incoming tweet for potential event creation:

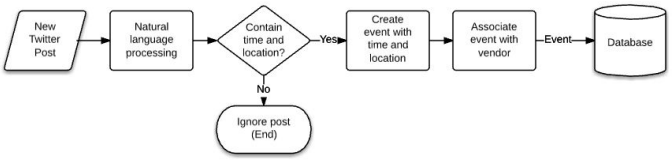

Fig. 2 . Fetching and storing Food trucks events from Twitter new feeds

1. A new twitter post was delivered by twitter API via Twitter4J
2. The post is then sent to a NLP (natural language processing) engine for contextual understanding
    a. The text of the post is first tokenized and processed through the Named-entity recognition (NRE) engine (provided by Stanford CoreNLP project)
    b. The tokenized words and NRE results are then being processed by our specialized location exactor
    c. The location extractor will evaluate the inputs and generates 2 outputs, 1 optimist result and 1 pessimist result. The optimist results contains more informations beyond words tagged as location, this will sometimes produce a better result since it includes more details that may have been misidentified by the NRE engine, but sometimes it will also includes unrelated date therefore compromising affect the processing on next stage. That's why a pessimist result was also introduces, the system can use it as fallbacks when the optimist result was rejected
    d. The optimist result will be first feed to the Google Location API, geolocation information is obtained using this method, if no relevant geolocation information is returned, pessimist result will be used. If the relevant geolocation information is still not returned, the program will discard the twitter post and go into standby mode, waiting for next tweet.
    e. With location information obtained, the algorithm will start to process the date information base in the tokenized text and NER results
    f. The date string extractor will evaluate the inputs and generate a sanitized text for the date regnization process
    g. Our date regnizator is provided by PrettyTimeNLP, it generates date objects based on language inputs. If no result is obtained the program will discard the twitter post and go into standby mode, waiting for next tweet.
3. The system will proceed if an event was recognized, otherwise, the post is ignored and the process is terminated
4. An event object is created with time and location based on NLP results
5. The event is associated with an vendor base on the Twitter handle of the poster
6. The created event is then stored in the database ready for query

B. Food truck vendor create event on this platform

Food truck vendor can create an event directly on this platform

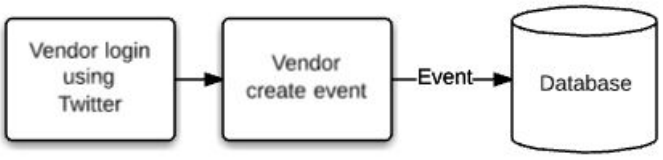

Fig. 3. Add new event by vendor

1. Vendor login to our system by using Twitter login
2. Vendor creates an event
3. The created event is then stored in the database ready for query

III. TECHNOLOGIES USED

A. Front-End:

We have primarily used Javascript, PHP and Mysql among others, to represent the front end part of the project. The basic idea of the application works is, the user enters the name of the location where he/she would like to eat, and specifies a radius, and the when the search button is clicked, the javascript part of the file, parses the name of the location, extracts the geometric coordinates and sends these as parameters as arguments to a PHP file running on a server. If the user chooses not to provide a location name, then the application automatically takes the current location of the user by using the maps' navigator API.

The PHP file utilises these values and queries the mysql database, populated by the Natural Language Processing API and uses mysqli_escape_string() functions to avoid sql injections, accordingly. In addition, it uses the Haversine formula for calculating the approximate distance between two points on a spherical structure, to further refine the results. The PHP program then uses the inherent DOM functions to output an xml containing the filtered data, using the provided parameters. This xml data is fed back to the javascript file, which then parses it and creates a marker for each event on the map, created by using the google maps javascript API. The markers when clicked, display a little information window, with the primary aspects of that particular food truck, like the name, it's twitter handle (which when clicked, redirects to the twitter profile of that food truck in a separate tab), the original tweet featured in the profile page, and link that opens up a separate webpage with google maps navigation instructions.

B. Natural Language Processing (NLP):

We're using the Stanford CoreNLP and Ocpsoft PrettyTime NLP as our major third party natural language processing engines.

Stanford CoreNLP is an advanced human language processing toolkit providing It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get the quotes people said, etc.[1]

We're using CoreNLP's named entity recognizer tokenize and recognize the type of the token, it provided a vital starting point for our project to further identify locations and dates.

PrettyTimeNLP is a human time parsing library for Java, it provides critical functions in a small footprint, it a simple social date-formatter [6] that will understand date.

We're using the PrettyTimeNLP to understand processed user inputs and generate date objects

C. Back-End Processing:

The Back-End mainly deals with storing the data arriving from the main index page as well as from the login page. The data fetched from the index page is stored in events table primarily consisting of twitter-handle, username, display-name, event-date, the location coordinates etc.

The database used to store this data is MySql and index page is run on javascript and PHP web technology. The login page is running on Node JS for server-side and AngularJS for client-side. The login process used in this project focus on oauth framework used for login from social media website such as facebook or twitter.

The project uses twitter login functionality with the help of oauth to achieve login process implementation for the food truck vendor. The login page is displaying all the events for that particular logged in food truck vendor. The food truck vendor can view his locations on the map and also view the related information in a display box shown below the map. Also, the user is allowed to create his own event apart from twitter and allow his event to be published to all the users using the website.

To implement the oauth technology, another database is used to store login information named as twitter login. THe authentication is done with the help of passport.js which will serialize the userdata and store the user information in the twitter login database.

Whenever the food truck vendor logs in again, the user data is verified with the database and allowed to login without authenticating with the help of twitter. This avoids the hazards of logging in again and again and saves the deflection of user from using the website.

IV. Potential Features

There are several potential features of this project which are beyond the scope. The first and the foremost is generalizing the idea not only limited to food truck but also to many other potential applications such as some social cause or any technical event by a company or any educational event by any school or institution.

A. Auto post to twitter after vendor created event on this platform

Once a vendor signed in to this platform via Twitter, auto posting feature can be enabled every time they posted an event. This would bring ease to the vendor since they don't have to do repeated work

B. "Open now" IoT device in vendor's truck

An IoT enabled button can be installed into vendor's truck. When vendor press this button, the device will automatically generate an event with its current location and time. With auto posting, they can even notify their followers on Twitter.

C. Engage users and crowdsourcing

Allow users to provide simple feedback about a particular food truck, whether it's open, out of stock on certain items or a photo of the menu.

V. Conclusion

In our initial stage of research, the experience took us to try various Natural Language Processing applications. We used a combination of three to four Natural Language Processing applications , preprocessed , post-processed the linking between each of the application and integrated the whole new combined application for our Natural Language Processing. As illustrated throughout the project, the food truck vendor is favoured thing we are focussing on and allowing the vendor to create his own events as well as tweet his events on twitter with the registered twitter handle.

The another most favoured thing is the user who is using the web application to find all the events he could locate within his radar from his current location and enjoy the events happening nearby his locality. The APIs and frameworks used in this web application are open source under General Public License. We believe that our code can be open sourced to add some helping hands in order to optimise the core methodology and functionality.

Acknowledgment

We would like to thank Professor Rakesh Ranjan, Dept. of Computer Engineering, San Jose State University, for getting us acquainted with the realm of Natural Language Processing

APIs, and guidance at various stages, which aided us tremendously in the completion of this project.

References

[1] "Stanford CoreNLP – Natural language software," Stanford CoreNLP. [Online]. Available: https://stanfordnlp.github.io/CoreNLP/. [Accessed: 12-Dec-2017].

[2] Google Maps API. [Online]. Available: https://developers.google.com/maps/documentation/javascript/mysql-to-maps . [Accessed: 12-Dec-2017].

[3] Sevilayha, "Easy Node Authentication: Twitter," Scotch. [Online]. Available: https://scotch.io/tutorials/easy-node-authentication-twitter . [Accessed: 12-Dec-2017].

[4] "MySQL Documentation," MySQL. [Online]. Available: https://dev.mysql.com/doc/ . [Accessed: 12-Dec-2017].

[5] "Twitter4J API," Twitter4J. [Online]. Available: http://twitter4j.org/en/index.htm. [Accessed: 12-Dec-2017].

[6] "prettytime," PrettyTime - Elapsed Timestamp Formatting and Conversion for Java (Social, JSF2) | OCPsoft. [Online]. Available: http://www.ocpsoft.org/prettytime/. [Accessed: 12-Dec-2017].