Geo-Tagging within

the

Free Text Document

Proposed by: Mathew Sheppard from FunnelBack

Tutor: Andrea Parson

Team Members:

Yuheng Wang u5553197

Wawin Sathyamoorthy u5542170

Shijie Qin u5553084

Junming Zhang u5722968

**Abstract**

Documents such as web pages frequently refer to real-world locations in a textual form, e.g. "Australia", "Canberra" or "Room N101, CSIT Building, ANU". While this form is fine for purely textual searching, it is often useful to have richer metadata in documents than the original author provided, as part of producing richer search experiences.

To address this problem, here we propose building a software system which can efficiently process text/HTML documents and add location metadata to it. In this way, the user may achieve a better experience.

The software system was built on Java platform with open sources such as IntelliJ IDEA in collaboration with Postgre SQL for working on GNAF database constituting the location coordinates of Australia.

Natural Language Processing algorithms, which were a part of Machine Learning is used to find and extract the location term from the web pages.

**Introduction**

Big data is having a huge impact on the world today socially, logistically and in the business sphere. Big data is the collection and analysis of vast sets of data that can be used to improve the way we approach the related problems.

Client

We are working for FunnelBack, which is an exciting and growing company headquartered in Canberra. They provide innovative websites and enterprise search solutions to organizations such as Westpac, ASX, ABC, University of Cambridge and Government sectors. They are recognized for leading analyst because they continually strive to produce solutions for every industry and business problem, from course finders for higher education to predictive segmentation and dynamic curation tools for the marketing team.

**Problems and Opportunities**

Why Geo-tagging?

Geotagging can help find a wide variety of location-specific information. The persisting problem is that the web pages usually tend to provide much of textual information. Considering geo-tagging a location, the locations are usually represented in the textual format apparently missing out the richer search experience

Why Choose only location term?

Since the Funnelback is a company that's based on the search engine, they are looking forward to making their search results to be more informative by adding coordinate to location term that is present in a web page, which enables a user to find location-based news, websites and other resources.

**Opportunities**

Working on the geo tagging the free text provides to explore the various search algorithms and their performances and well as gain knowledge about the open source natural language processing. Apart from learning, helps us to practically implement the code using coding languages. Working on this project not only provided technical knowledge but also managing and planning the team as well as making use of online interaction to keep up with the progress.

Helps in gaining knowledge regarding geographic information system and large-scale data processing.

**Methodology**

The system comprises of three phases such as text extraction, location finder and matching the database.

Text Extraction

An open source called JSOUP is used to extract the text from web page and save as a text file. The ultimate goal of this stage is to extract only the location terms from the whole web page containing a huge amount of textual information.

Location Finder

Natural Language processing which is a part of machine learning. NLP provides the feature of classifying the location term from the huge text document or saved file.

Matching Database

The resultant output location term from the above phase is used to match the GNAF (Geocoded National Address File) provided by FunnelBack to produce the coordinates for the location terms.

**Outcome**

With the proposed system we can help extract the location phrases and return their longitude and latitude. The results can be directly used by other software to process further analysis. The geo-tagging software system will reduce the time consuming and manual work as well, which is a really useful tool in the search engine field.

**Phase 1**

Initially, after the team was formed we planned to go through some pre-reading regarding the project as the team needs to know about the concepts and understand it properly to implement it. A regular team meeting was conducted to discuss the terminologies and coming up with a solution that was appropriate and meets the client expectations.

Very first client meeting was conducted at the client office, where the client was able to explain what we are dealing with and what is the final result that the client needs. The client shared many informative links with the team members so that we can use them wisely to get the results.

The team's very first idea was to start from scratch making each and every function. The team decided to implement this system for Canberra initially and extend the database to Australia after achieving the coordinates for the location. Made use of machine learning terminologies such as StopWords, Filters, and BagOfWords to solve the project. This method wasn't effective to extract the terms from the web pages. So changes had to be made to successfully obtain the output.

We shared our problems with the client and upon conducting a meeting with the client, he was able to direct us with different ideas that we could work to check if the result was effective to produce the desired output. To extract the text from web page he suggested us to try out JSOUP which is an open source. After the text was extracted Capital letter extraction method was used to capture only the location terms. GNAF database was provided by the client which comprised of the location details such as city, suburb, street for Canberra with a size of 1.5GB.

There were more or trail and errors that we had to do. The capital letter word extract seemed to be problematic as the probability of words that begin with a capital letter and the word being a location term was very less.

For instance, Every sentence begins with a capital letter and the chances of the word being a location were less. The Capital letter extraction had to be dropped down.

As the database was huge we had to load the database firstly. Postgre SQL was used to load the database along with PGADMIN which maintains tables containing the list of areas, suburbs and street names and address.

As the concept of Capital letter extraction was ruled out, a suggestion from the client was given to go through the Natural Language Processing which was mentioned in the project description. There are many open source NLP those are available. Apache NLP was used to extract the location term out of many available open source NLP as this was suggested by the client. Location terms were extracted by using this NLP and after combining the codes altogether the coordinates for the resultant location terms were produced. The problem with this method was that the accuracy of the location extraction by comparing with the actual web page was less and the speed of execution was slow.

So the team had to choose an alternative method to increase the accuracy of the location term extraction. Moreover, the ultimate goal of the system is the produce the result as quick as the possible as this was the main functional requirement from the client. With this system, the client will be able to incorporate this feature into their search engine.

**Phase 2**

Since accuracy and speed were our ultimate goal. Team planned for improving the accuracy first. Stanford NLP was used instead of Apache open NLP to check if there was any improvement, the Stanford NLP outperformed the Apache NLP by giving 90% accuracy in location extraction. So the code was manipulated accordingly to produce the result.

The client needed to test on a sample of 100 web pages containing location terms in order to check the performance of the system. So a test set was created to check the accuracy of the produced result. Slight issues were to be considered such as sentence definition differs from persons to person. Further improvements had to be carried out.

Currently, the team is a working on the sentence definition, since our program at this step can only extract the locations from sentences, that's to say, for the rest words and phrases, we simply match them with the database directly to get the results. For the consideration, our team member discussed and decided to create a standard rule for sentences in the web pages. In this way, we create the sentence definition that helps to extract the paragraphs from web pages.

Further, the database was extended as planned in phase 1 to Australian database. The extended database contains nine states, 16,413 suburbs, 701,290 streets and 14,175,700 addresses, and its storage cost is about 6GB. Manipulation of the code has to be carried out to cut down the amount of time needed for processing the database and matching with the database with location term.

**Management of the project**

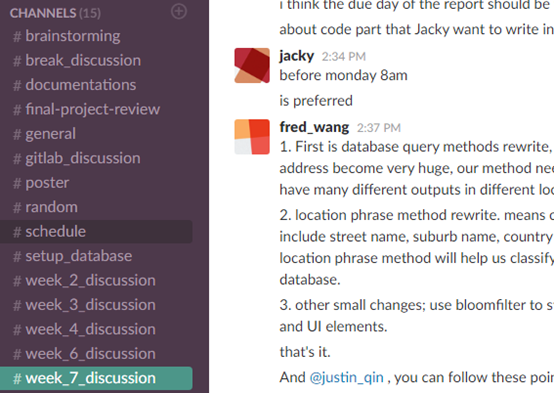
Initially, the team management was done using the social applications such as Facebook messengers to communicate with the team members.

Later in phase 2, the team moved on to a professional communication tool that allows members not only to communicate but also to share the files and channels for each week discussion. The client was added to our management tool so they can keep track of the team progress.

**Team management**:

Slack

This semester we create a new channel each week and each section of the project. So far we think It works well because it helps us to find out what has happened each week. We use it as part of the records of the documentation. It helps members to track what we have done and provide evidence for the contribution of the project.



Git

All the codes, milestones and issues were tracked and managed using the GitLab. Any modifications done to the codes had to be pushed to the Git repository named as Geo-tagging with suitable commit and issues. Repository comprises of relevant documents and materials that were useful for academics.

**Software's**

* IntelliJ IDEA 15 for java programming.
* Postgre SQL database for connectivity between java code for matching the database.
* PgAdmin for database loading.