Trend Identification on Stack Overflow

Project's client: Christoph Treude

FINAL REPORT
COMP SCI 7098 - Master of Computing & Innovation Project

by

Po-Yi Lee

a1806207@student.adelaide.edu.au

School of Computer Science

University of Adelaide, Adelaide 5005, Australia

June 2021

Abstract

In this project, we aim to identify trends from one of the largest programming websites called Stack Overflow (SO). It is not handy for people from different industries to search up the popular words or concepts used or interested in by humans in the technology world. After extracting the SO data, the posts from SO were sent to the tool of the Natural Language Processing (NLP) to retrieve noun-noun and adjective-noun phrases. Finally, we ranked these phrases within user-specified time windows by counting their frequency and visualized these results on the webpage we created. With the searching application we created, it is easier for people interested in the technology ecosystem to learn how it is changing and where it might be going in the future.

1 Introduction

This project and its application make it convenient for the users to explore the trendy concepts discussed by people on the SO forum by simply entering a number of values and a

keyword they are interested in. Data analysis is getting more and more crucial for everyone to foreshadow technology trends. According to a survey of CS graduates, the content taught in CS courses are not aligned with employer requirements [1]. Therefore, the education institutions could design more suitable courses for their students once they are able to obtain insight from trends in the technological world. For technology industries, companies have started to invest a lot in the cybersecurity industry since it is becoming a popular topic according to the visualized data. In addition, more enterprises have been using the amount of data to understand technology trends in order to analyze their companies' prospects and make their business strategies better.

Now there is the Stack Overflow Trends searching system online on its official website which contains abundant and clear searching content for trends identification on SO. From our research, however, most of the websites only provide the single word as a tag for the search such as Windows, Android and so on. Additionally, it is difficult for humans to identify the trends by looking at all the posts on SO. Therefore, this motivates us to extract the two-words noun phrases using the tool of Natural Language Processing (NLP) algorithms as we consider that there is still a shortage of collocations, a pair of the words, seen as trend keywords and more types of the words can be more useful for the trends analysis. There are three major components of our software: data retrieval (requesting Stack Overflow posts via Google BigQuery), trend identification (pre-processing of the posts and extraction of phrases of certain patterns) and website development (building the user interface to accept user inputs and display the output). The system overview below demonstrates how the software works in detail (Figure 1).

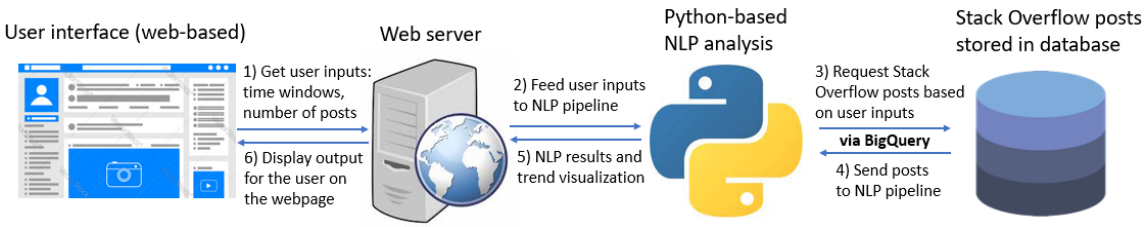

Figure 1. System overview of the product

2 Project Aims

Due to the reasons in the introduction section, our goal of the project is to develop a software with a user interface where the users are able to specify the time window which they are interested in and extract two-word noun phrases for trend identification on SO. The software was able to retrieve data corresponding to the time frame via Google BigQuery service, performed NLP algorithms on the posts of the dataset and extracted two-word noun phrases from the text of the posts. Then we can filter these phrases into only adjective-noun and noun-noun phrases. At the end, graphs on the application will be demonstrated as the results of the changes in trends over time.

3 Approach

The approach in this project will be described from users' inputs to graph results demonstrated. First, we needed a tool to have access to this data from the SOTorrent database in order to process the data. Given the very large amount of data on SO, however, it was difficult to download them on our computer. Therefore, we found a very useful cloud platform called Google BigQuery, used to query the SO dataset. Once we set up authentication and obtain a key from here, we are able to manage the data easily using SQL-like syntax. The query performance was still slow without a better CPU due to the large amount of data but the efficiency is not the client's requirement.

Natural Language Processing (NLP) is designed as the way humans extract meaning from natural language such as English, Spanish or Chinese, in textual or spoken form [2]. Among the various text-processing methodologies employed by text-mining, one of the most important is NLP [3]. Due to the demand of the text-mining in this project, we decided to choose NLP tools to extract the words we need and the programming language we opted for was Python since most of the team members are more familiar with this language than the others and it also can provide developers with an extensive collection of NLP tools and libraries such as Natural Language Toolkit (NLTK), spaCy and so on.

After removing some useless information such as HTML tags and code blocks in the SO posts we extracted via BigQuery, we searched some of the methods for extracting two-words noun phrases and used Noun_chunks function but the result was not good because the chunks can be of various lengths but we are more interested in trends in bigrams. Finally, we chose spaCy as the NLP pipeline and its rule-based matcher to be able to extract noun-noun and adjective-noun phrases. The other reason why we used spaCy is its excellent balance between efficiency and accuracy, and also one of the best tools to analyze SO data according to our client Christoph's studies [4].

After the extracting process, we stored all eligible phrases in a list and counted their frequency to rank their popularity. To satisfy the client's requirement, we want to visualize the result to the users. Therefore, we created a web application framework using Django, a simple Python tool for back-end development and we designed it with HTML, CSS, JQuery and JavaScript. For trend visualization, we chose Apache ECharts, a powerful charting and visualization library for browsers, to present our trend results.

4 Results

4.1 User Interface

The website interface (figure2) shows several user inputs including date period, the number of time windows, the number of posts per time window, posts type, sorting way, spelling correction, singularization, contained phrase and the number of the phrases

displayed. Since our software will show the changes in SO trends over a time period, users should enter the overall time period and the number of the sub time windows in the date period.

4.2 Authentication setting

We also provide the key uploading section (figure 3). Once users own your key (JSON file) from Google BigQuery, users can sign up and sign in with their own account and upload their key to obtain authentications for data extraction. This could be not convenient for users but we can only provide this service for users at this stage. We might use the better CPU and could store all the data in our server end instead of extracting data from a cloud platform.

4.3 NLP methods

We cleared and filtered our data using spaCy's rule-based matcher and phrases can be successfully selected out and listed with their frequency for each time window in a CSV file (figure 4) as the client's requirement.

4.4 Output result

The software will demonstrate the output result in the new webpage according to users' inputs. The output results are presented with a graph and the basic search information on the bottom (figure 5). For example, if the user chooses January 1st 2010, December 31st 2020, 2 time windows, the graph result (figure 6) will divide the date period into 2 time frames and show the phrases and their frequency in each time window.

4.5 My individual achievements

My main achievements will be divided into two parts. The first part is data retrieval and data preprocessing. Although it took long time for the team to download the dataset and affect the efficiency of the project at the initial stage of the project, I solved the problems that how we can extract the large amount of data from the online database and figured out how the cloud platform Google BigQuery works which help the team request the posts on SO more handily. In order to request the data from BigQuery, I checked if the correct data is extracted in the dataframe type. Furthermore, I was able to remove the HTML tags and code blocks from every post of a dataframe.

The second part of my achievement is to create and design the web application using HTML and CSS. Aside from being very new to writing HTML and CSS code for me, the most challenging task is to cooperate with the other team member who was responsible for the back end. Once the context of the back end was adjusted according to the client's requirements and recommendations, I should communicate with the front-end side and match my code with the back end. Besides, I helped test the code of the back end and

figured out the testing problem together in the online meeting. This project helps me learn how to communicate with other teammates to satisfy the client's requirements.

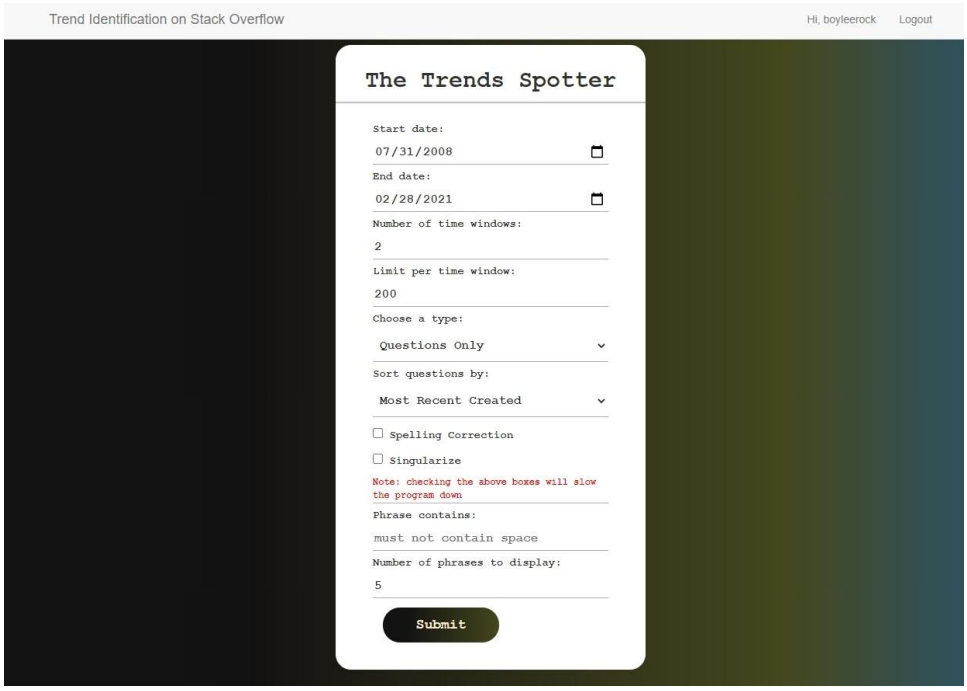

Figure 2: The user interface of the trend identification on Stack Overflow.

Figure 3: The key uploading section.

	A	B	C	D	E	F	G	H	I
1	Unnamed	word 1	word 2	frequency	2010-01-0	2015-07-02 12:00:00 - 2020-12-31 00:00:00			
2	0	web	service	8	5	3			
3	1	google	map	7	4	3			
4	2	sql	server	7	3	4			
5	3	visual	studio	5	3	2			
6	4	json	object	5	1	4			
7	5	inner	join	4	3	1			
8	6	map	api	4	3	1			
9	7	html	element	4	3	1			
10	8	unit	test	4	3	1			
11	9	entity	frameworl	4	2	2			
12	10	jquery	mobile	4	2	2			
13	11	android	studio	4	1	3			
14	12	datum	type	4	1	3			
15	13	error	message	4	1	3			
16	14	shell	script	4	1	3			
17	15	core	datum	3	3	0			
18	16	objective	c	3	3	0			
19	17	best	way	3	3	0			
20	18	radio	button	3	3	0			

Figure 4: the phrases and their frequency listed in CSV file

Basic Search Information

Download full results as csv

start date	end date	number of time windows	limit per time window	post type	sort by	phrase contains	spelling correction	singularize	number of phrases to display	total number of eligible phrases
2010-01-01	2020-12-31	2	1000	titles	random		enabled	enabled	5	2523

Figure 5: The table of the basic search information

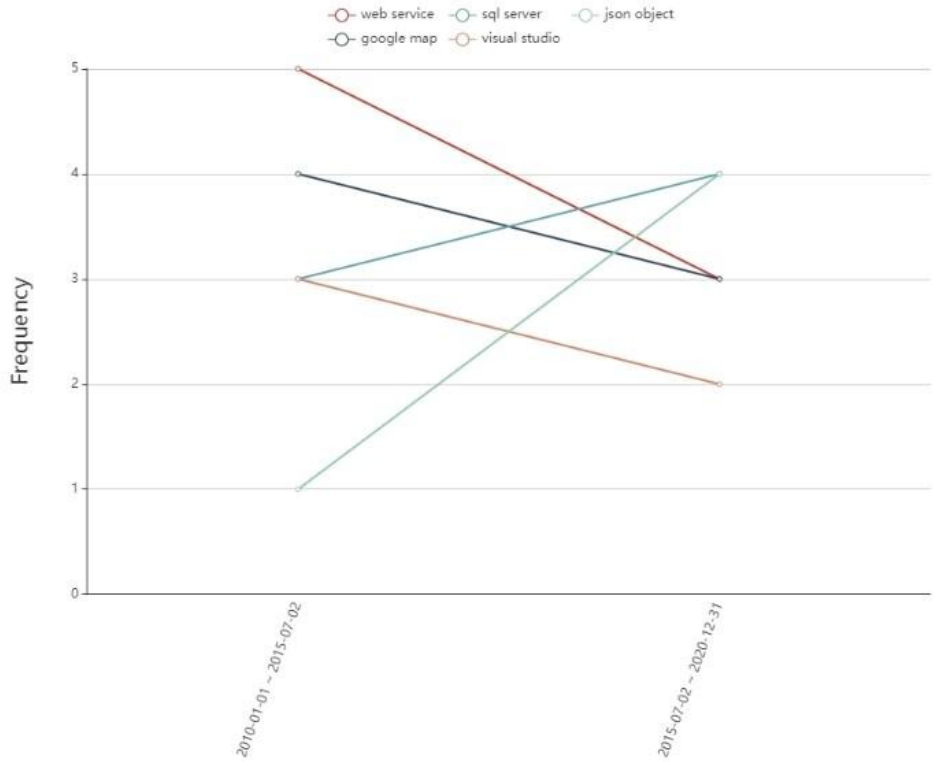

Figure 6: The graph results with x-axis: time period and y-axis: frequency.

5 Conclusion

Our main goal was to extract collocations (noun-noun and adjective-noun phrases) from Stack Overflow data, and visualize trends in this data. After retrieving and preprocessing the data, and using spaCy's pre-trained matcher, we were able to satisfy the client's basic requirement with the software which allows the users to search for trends within a particular time frame and will visualize changes in trends over the time frame. By following the client's requirements and advice, we did the testing, changed the NLP tools we used and the way we demonstrated the trends results. Finally, we were able to create an application with trends visualization in SO. With additional time, we added more options of the users' inputs to make the results more precise and valuable for the people who need to identify the most popular technology concepts in the specific time frame. In the future, the software could be improved by increasing the processing speed, expanding the range of searching algorithms and varying the way to visualize the trends results.

References

[1] Department for Business, Innovation & Skills. (2016). *Computer science graduate employability: qualitative interviews with graduates. London, UK: BIS Research Paper*

[2] Chowdhury, G.G., 2003. Natural language processing. Annual review of information science and technology, 37(1), pp.51-89

[3] Linguamatics. (2019). What is Text Mining, Text Analytics and Natural Language Processing? [online] Available at: https://www.linguamatics.com/what-text-mining-text-analytics-and-natural-language-processing#:~:text=Text%20mining%20(also%20referred%20to [Accessed 24 Nov. 2020].

[4] F. N. A. Al Omran and C. Treude, "Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments," 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), 2017, pp. 187-197, doi: 10.1109/MSR.2017.42.