Algorithm Design and Analysis - WIA2005

Semester 2, 2018/2019

Group Project Report

Group Name: Keyboard

Lecturer Name: Dr. Ali Mohammed Mansoor Alsahag

Team Members:

Name	**Matric Number**
Hazem Hadi Hazem Allbabidi	WIF170709
Jasia Tunazzina	WIE170701
Amirul Zharfan Bin Zalid	WID170003
Muhamad Syafiq Bin Abdul Razak	WIF170091
Muhamad Aiman Bin Shamsudin	WIF170089
Ahmad Izzman Bin Mohd Sharfuddin	WIF170011

Table of Content

Introduction

The project is about multiple different tasks and questions. Each task or set of questions has a solution that uses an algorithm. Our job was to find out how to solve these questions and how to find the best algorithm that can produce the best solution with the smallest time complexity. The first few questions consisted of questions that ask for an algorithm that will calculate the shortest path from Kuala Lumpur to a specific destination by transiting 2-3 different cities. The other set of questions, asked us to use the Rabin-Karp algorithm to find and delete specific words in the articles.

Method

The Modules/Libraries Used:

- Plotly: Plotly displays the data retrieved in graph form such as line graph, bar graph, and scatter graph.
- Requests: Requests was used to retrieve the information from a specific URL to then be used to access the text in the Website.
- BeautifulSoup: BeautifulSoup was used to extract the text that was received from the “Requests” module. It collected the paragraph text from the HTML code of the online article.
- Goose3: Used to extract text from a given URL.
- Textblob : Used to identify the polarity of the text from the article
- GeoPy : Used to gain the information of latitude, longitude and distance o the cities.

The Algorithms Used:

- Rabin-Karp Algorithm: It was used to identify and delete the stop words from the text retrieved.
- Dijkstra's algorithm : To find the optimum path for the flight route recommendation
- Exponential Searching Algorithm: To compare and categorize words from newspaper of the web pages and determine it to be positive, negative, or neutral words.

Description

Task:
We have been assigned to develop a system to help the end user to find the best flight to use when there is no direct flight available. The system will recommend the end user the best flight only based on the route and the political sentiment of the country. The best flight does not necessarily be on the shortest route but also considering the political situation each country the flight transited. There is no price of ticket involved in this recommendation system.

Analysis of topic:

Analysis	**Item**
Function	● Recommend the best flight route to end user ● Analysis of the shortest path between countries ● Analysis of the political sentiment of the country ● Recommend based on the optimum cost of distance and political situation in the visited country
Rule	● Departure country are from everywhere in the listed country ● No price for ticket is involved ● End user can choose where they want to go

Work Allocations:

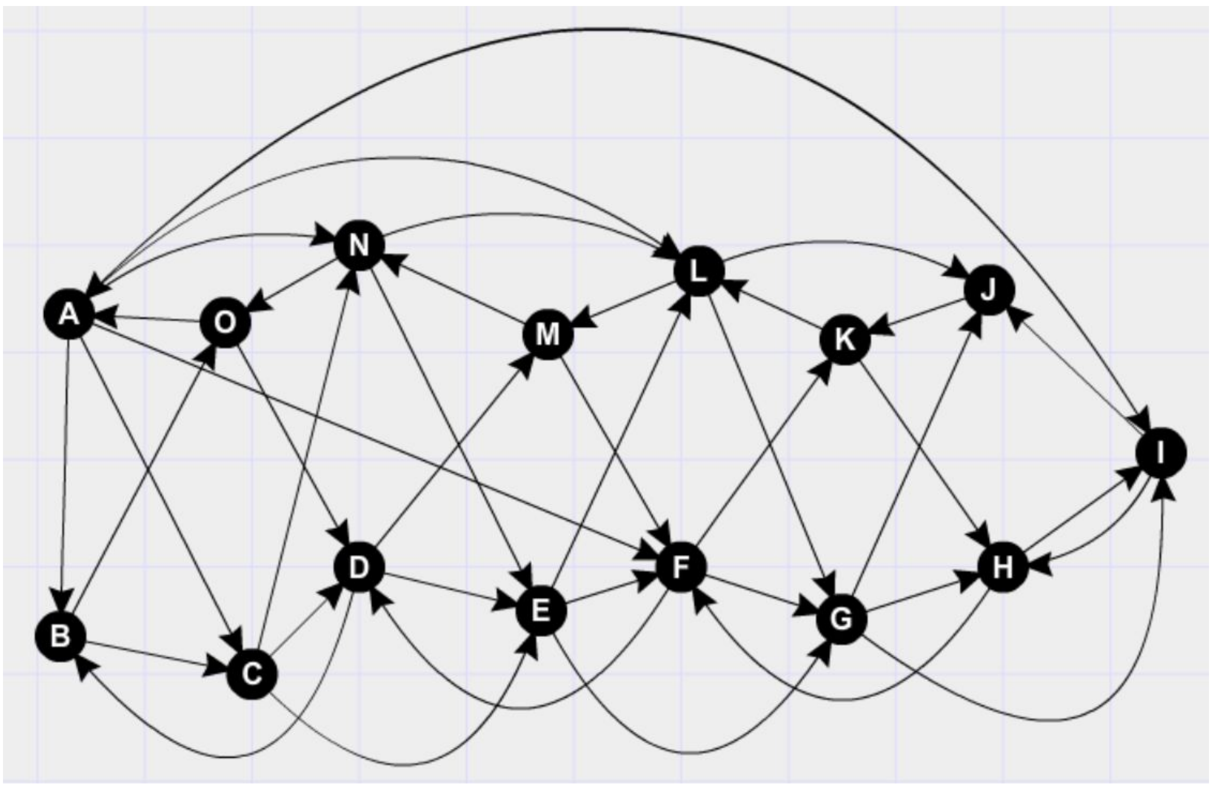
Name	Role	Task
Amirul	Leader	- Design and implementation of UI - Apply multiprocessing parallelism to optimize algorithm running time - Create Google Map view for the route - Create REST API for accessing the method and function - News article processing by getting the text for word processing
Hazem Hadi	Member	- Retrieving the URL of each article. - Extracting the data from the articles. - Calculate the frequency of each word. - Report writing.
Syafiq	Member	- Do the shortest path algorithm - report writing
Jasia Tunazzina	Member	- Plot the frequency in Plotly using line, bar, and scatter graphs.

		- Choose the article to be retrieved. - Report writing.
Muhamad Aiman	Member	- Implement Geopy library - Report writing
Ahmad Izzman	Member	-Count frequency of positive and negative words - Report Writing

Implementation

For question 1 until 5, we assume to have these airports:

1. KLIA
2. Charles de Gaulle
3. Soekarno-Hatta International Airport
4. Indira Gandhi International Airport
5. John F. Kennedy International Airport
6. Dubai International Airport
7. Incheon International Airport
8. Pyongyang Sunan International Airport
9. Heathrow Airport
10. Yangon International Airport
11. Toussaint Louverture International Airport
12. Santiago International Airport
13. Damascus International Airport
14. Baghdad International Airport
15. O.R. Tambo International Airport



Above image is the directed graph used to show the available flight path. From these cities of the airports, we collect articles about the political situation and analyze the article using the TextBlob library to get the polarity of the article. If it return as negative value, the city may have some conflicts and not recommend to fly through that area. We also use Goose3 to extract the content from the article and pass the value to TextBlob function to have the string analysis.

We have 3 python files which are “path.py”, “geocoder.py”, “pone.py”, “alternate.py” and “article.py”. The codes written in “path.py” are focused on getting data on airports around the world and finding the shortest path of the airports and . In “article.py”, the codes are focused on extracting words from websites and analysing the words to positive and negative words. The file “geocoder..py” deals with retrieving the latitude and longitude of the cities. The file “pone.py” used in counting the positive and negative words. And the “alternate.py” file used in finding the other possible route. We use multithreading in file “geocoder.py” and “article.py” to make use of the multicore and speed up the processing time. We integrate our Python3 coding with web based application using Flask library. Flask act as intermediary between the system and the user interface.

For question 5, we used online articles about politics from various different countries including Japan. We took the article and used the website https://www.textise.net to convert the online page to a page in which the text can be easily extracted. After that we used the “dictionary” in python as it is an implementation of a hash table which helps decrease the time complexity for this task. We used the dictionary to add the words, then we used a loop to count for the frequency of each word. During the loop, if a word is one in the text, 1 is added to the value of dictionary with the index being the word. But before finding the frequency, we added the functionality that the program would remove the stop words from the text of the article. We used the Rabin-Karp algorithm for deleting the stop words from the text to be analyzed.

Question 6 is based on the previous question. It asks for different types of graphs that portray the data we retrieved in question 5. So it takes the dictionary as input and uses Plotly to make a graph for it. It can be placed in a line graph, a bar graph, or scatter graph. When the program runs, it uses the plotly library, as well as the username for an account and the API Key for it, it takes in the data and goes to the website of Plotly and displayed the desired graph with the given data from the article.

From question 7 onwards, we are required to compare the words in the newspaper of the web pages to determine the frequency of positive, negative and neutral words. The positive, negative and neutral words from the web page are separated accordingly. The total of those three word types are compared. We implemented the exponential searching algorithm to reduce the time complexity to O(log n). The next question, question 8, took the data obtained from the previous question and plots a graph to compare the previously mentioned criteria. We used the plotly module to create the graphs. Then, we concluded that larger gap between positive and negative words determine the credibility for the recommendation of destination and neutral words validates the accuracy of that criteria.

For question number 10, to have the probability of each city, we get the polarity values for each of the city. The positive polarity and the negative polarity is summed up. To get the probability of positive polarity city, 70% is divided with the total of positive polarity. The same process going on the get the probability of negative word. Then the distance from KLIA to each of the cities will be multiplies with the probability of each cities. Then the total is summed up to get the expected value.

Time Complexity

As in code, the worst case of time complexity will be $O(n^2)$. But for our geolocator function and article analysis function, we are using multithreading. Thus the time complexity will depend on the CPU thread. On average PC, it use 8 thread. But for our project, we are using CPU with 16 thread. Thus the time complexity in geolocator function and article analysis function, the time complexity will be O(n/16).

For question 5, there are two for loops to process the text and add it to a dictionary, they are placed separately which makes the time complexity equal to O(n). There is a function as well for deleting stop words which has two loops inside each other, so the worst case time complexity is equal to $O(n^2)$. For plotting the diagrams, the time complexity is simply O(1). for each of the three function.

Thus for the overall time complexity will be $O(n^2)$ as it is the worst case.

Conclusion

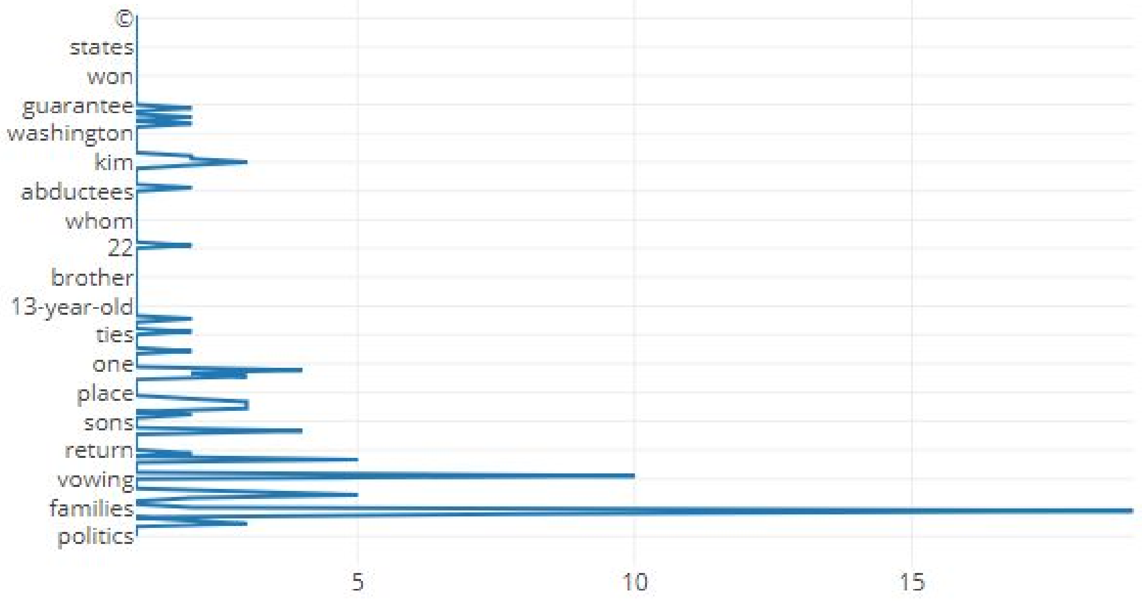
To summarise our project, the implementation of what we have learned in Algorithm Design & Analysis course was fairly applied. There are three algorithms that we have implemented in this project such as Dijkstra's algorithm, Rabin-Karp Algorithm and Exponential Search Algorithm. We installed a few libraries to further enable the project to be correctly run. We used Plotly, Requests, BeautifulSoup, Goose3, TextBlob and GeoPy. At the end of this project, we succeeded in achieving worst case time complexity of O(log n).

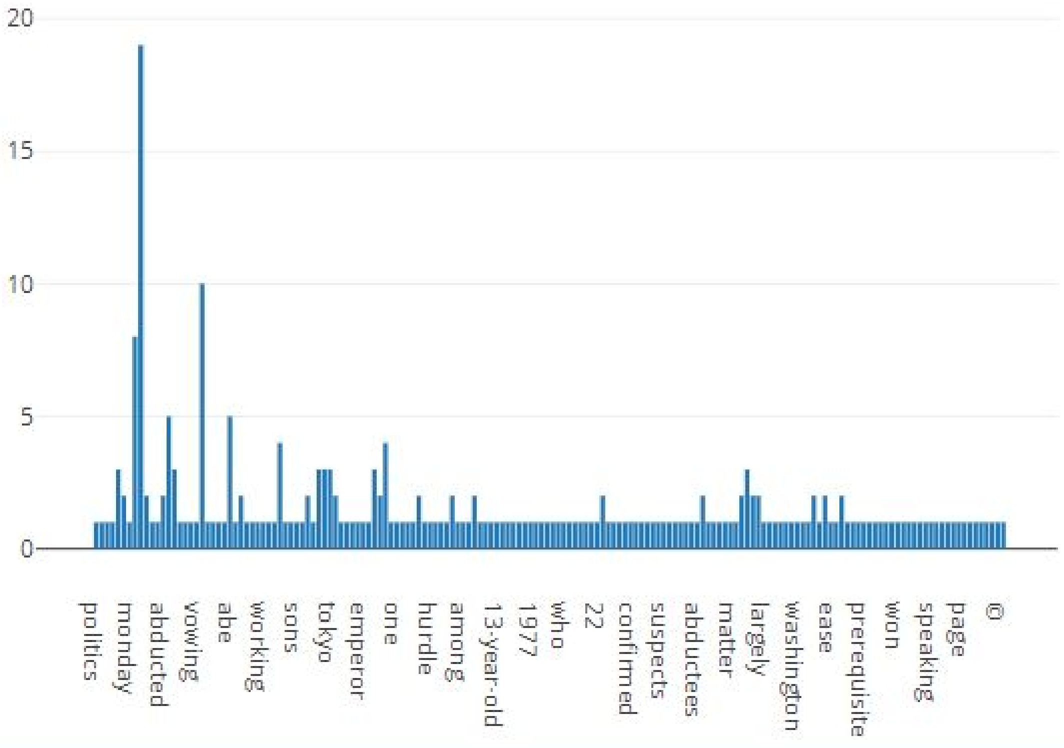
Leadership was a good normalizer for our group. The task breakdown was clear and immediate and all task was properly run through. Communication was the biggest asset of this team. Each team members had shown fair cooperation towards their commitment in this project to ensure the succession of this project. Due to the challenge ahead of the development of this project, we have acquired new knowledge of programming and further tested our capabilities on algorithm design. We are grateful for having our lecturer to be guiding us along during the development phase. During this phase, we also manage to improvise to ensure the simplicity of the codes carried by dynamic functions and methods. The project has proved to be a great success as we have reached our main goal including some extra tweeks.

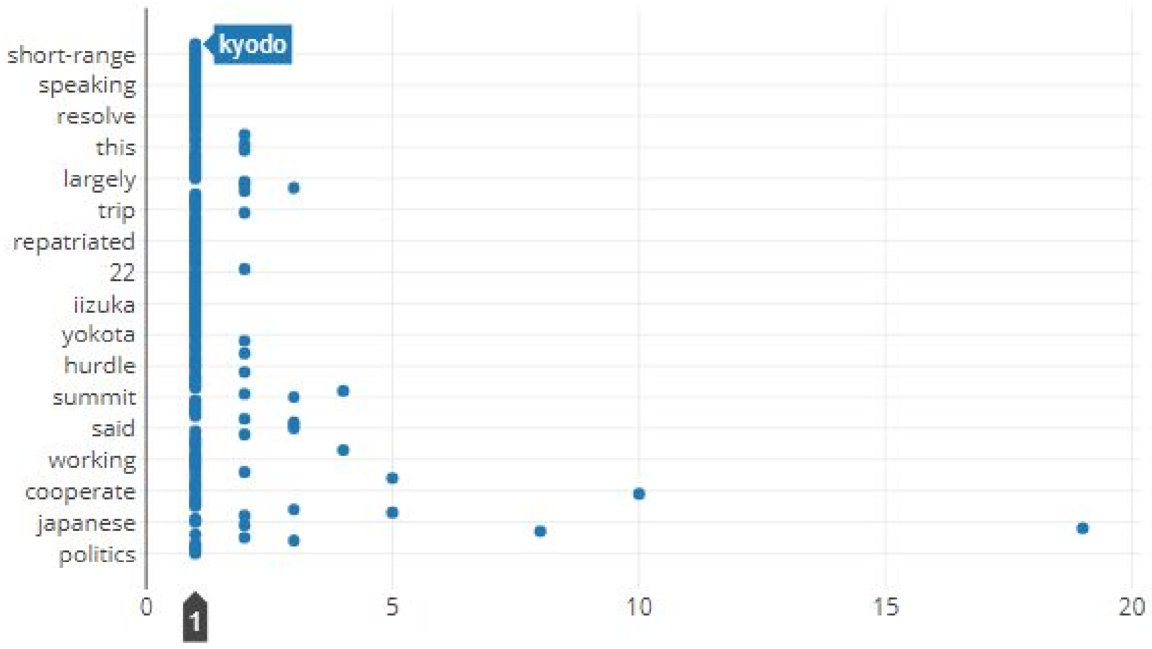
Reference

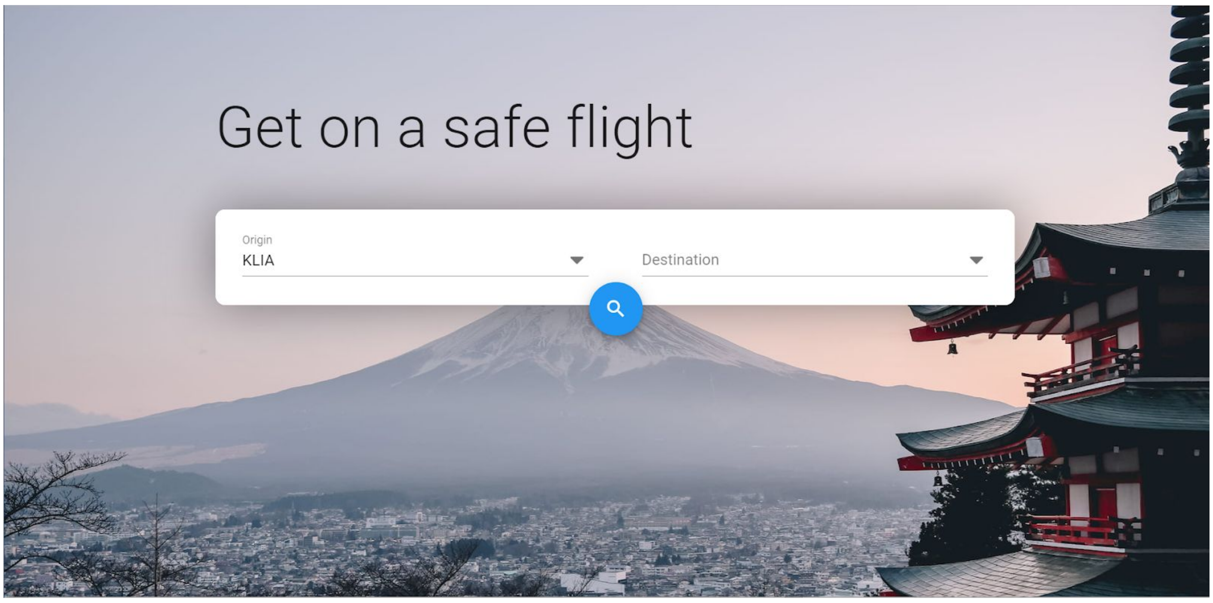
- https://www.textise.net/
- https://www.textise.net/
- https://www.ranks.nl/stopwords
- http://www.instructables.com/id/Plotly-with-Python/
- https://plot.ly/python/getting-started/
- http://positivewordsresearch.com/list-of-positive-words/
- http://positivewordsresearch.com/list-of-negative-words/
- https://www.youtube.com

Screenshots





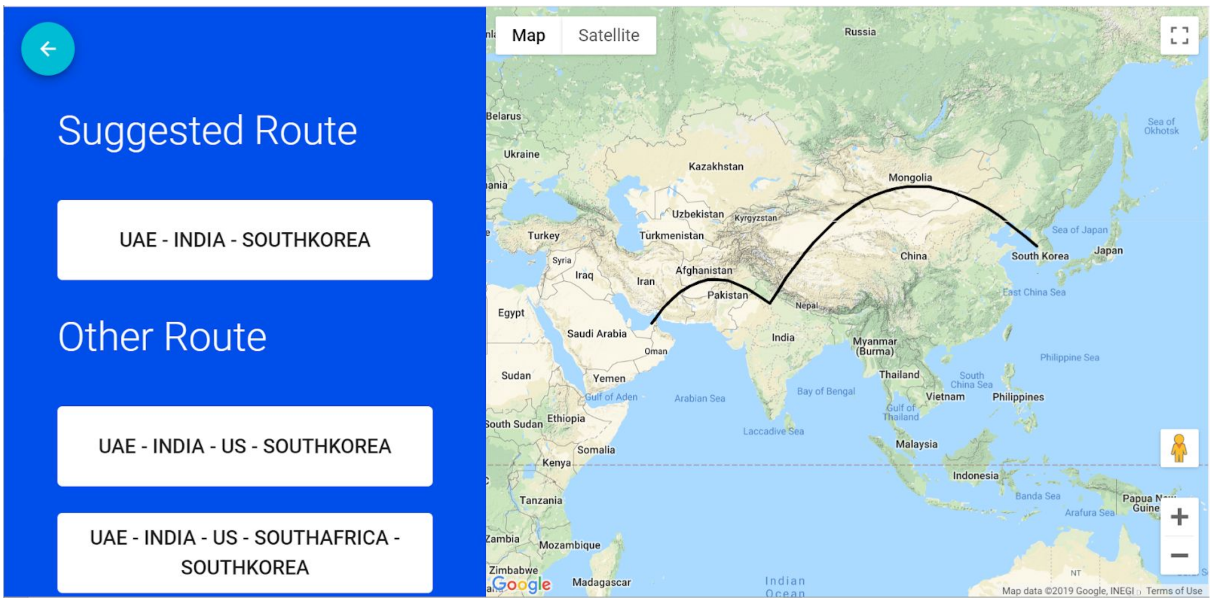

kyodo
short-range
speaking
resolve
this
largely
trip
repatriated
22
iizuka
yokota
hurdle
summit
said
working
cooperate
japanese
politics
0
5
10
15
20
1



Get on a safe flight

Origin
KLIA

Destination



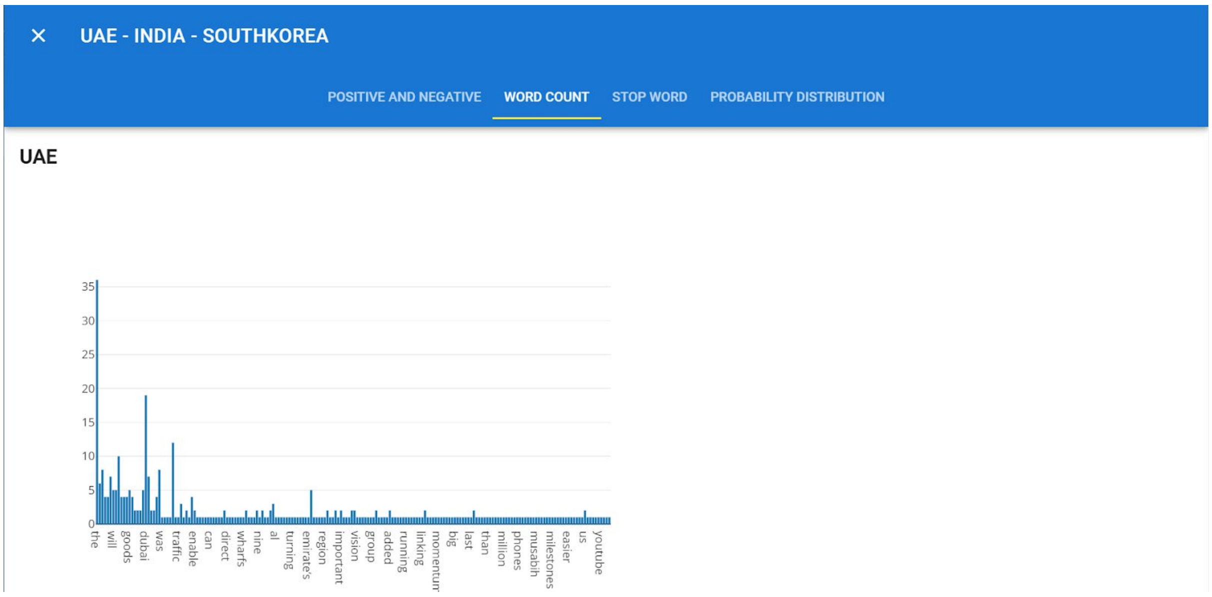
Suggested Route

UAE - INDIA - SOUTHKOREA

Other Route

UAE - INDIA - US - SOUTHKOREA

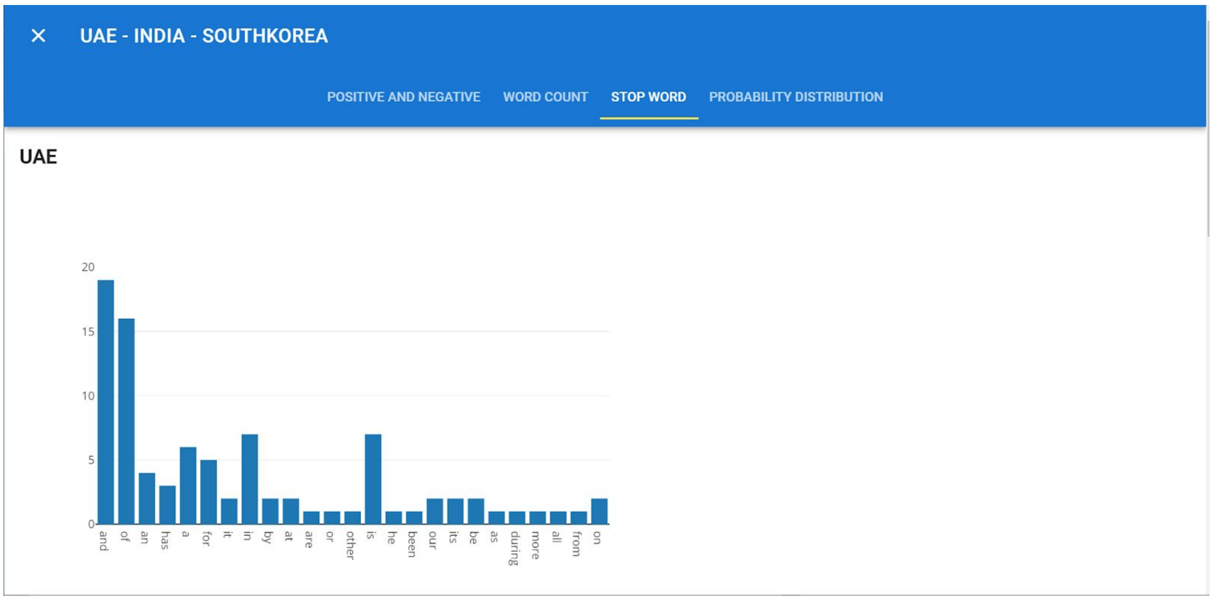
UAE - INDIA - US - SOUTHAFRICA - SOUTHKOREA

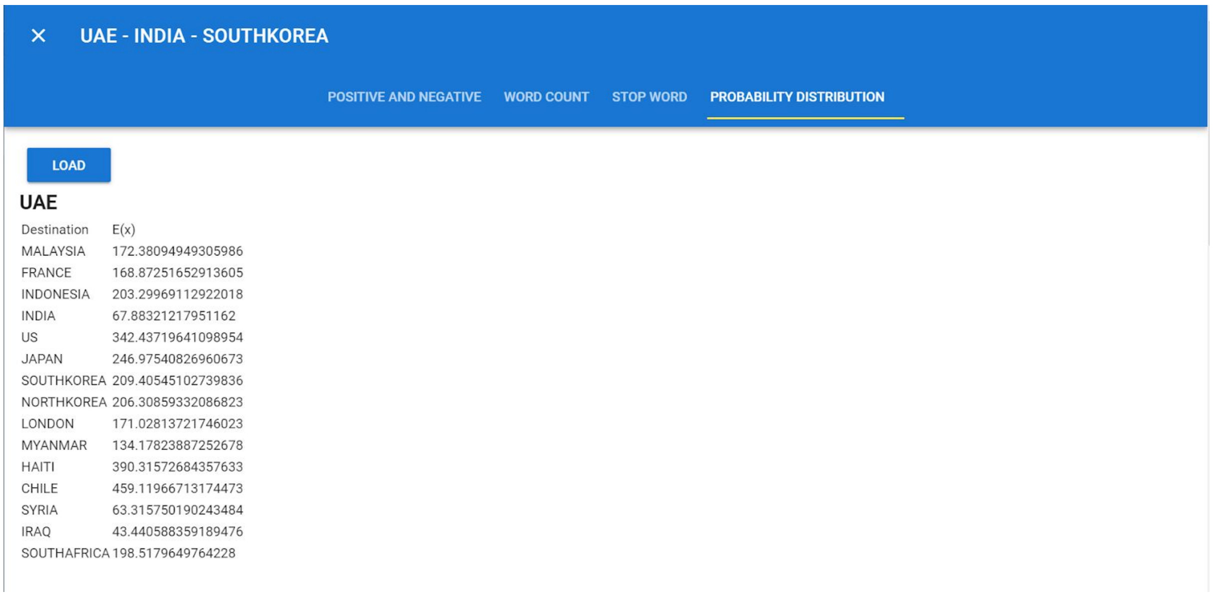


UAE - INDIA - SOUTHKOREA

POSITIVE AND NEGATIVE
WORD COUNT
STOP WORD
PROBABILITY DISTRIBUTION

UAE





UAE

Destination	E(x)
MALAYSIA	172.38094949305986
FRANCE	168.87251652913605
INDONESIA	203.29969112922018
INDIA	67.88321217951162
US	342.43719641098954
JAPAN	246.97540826960673
SOUTHKOREA	209.40545102739836
NORTHKOREA	206.30859332086823
LONDON	171.02813721746023
MYANMAR	134.17823887252678
HAITI	390.31572684357633
CHILE	459.11966713174473
SYRIA	63.315750190243484
IRAQ	43.440588359189476
SOUTHAFRICA	198.5179649764228