Report | Project Part 4

The project has been published on GitHub at the following URL and is available within this specific tag:

URL: https://github.com/ericdalmases/IRWA.git
Tag: IRWA-2023-part-4
(https://github.com/ericdalmases/IRWA/releases/tag/IRWA-2023-part-4)

For this delivery we are given a template of a web application and we are asked to implement our search engines and provide some statistics and dashboards through the user interface.

The aim of this report is to explain all the things we have implemented to the web application template, for that we are going to explain all the pages and tabs, one by one.

Search

First of all, in the search page we implemented all our 3 search algorithms (tf-idf, popularity based & word2vec) so that the user can choose the one that fits best to his needs. As in the previous labs, all the terms of the query must appear in the retrieved documents, fulfilling the AND condition.

This page is the simplest one in terms of visualization but at the same time it is pretty complex since in the back-end the three search engines we did in the previous deliverables are implemented, as commented above.

Based on the already provided SearchEngine class, we used it to implement our search methods, getting `SearchEngineTfIdf`, `SearchEngineOurScore` and `SearchEngineWord2Vec. Their implementation has been already explained in the previous labs, but all this can be found in the file `search_engine.py` . For the first two methods we decided to put the search and ranking algorithms in the file `algorithm.py` to make the code more simple and understandable.

In Figure1 we can observe our search page. We have also added for all pages a navigation bar which is very useful to go through the application in a quicker way.

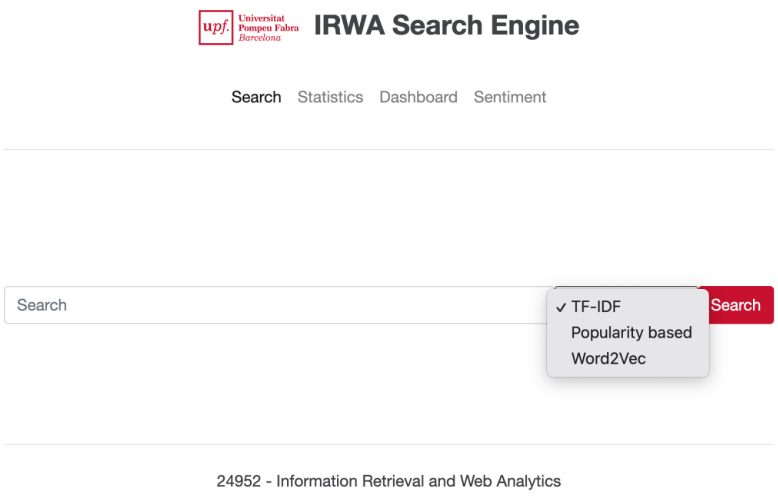

Figure 1

Then, for the ranked results page we decided to keep the same structure that was already provided, since all the relevant information was already shown. We decided to do some UI modifications but they are minor, such as setting the title to bold and modifying the date structure. See Figure 2.

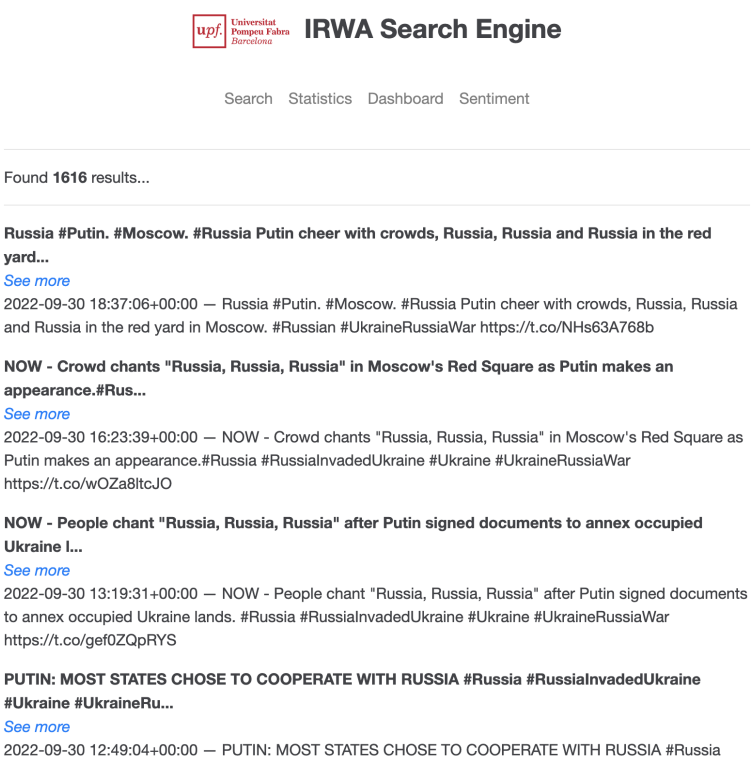

Figure 2

By clicking on the “See more” button you can access more detailed information about the specific tweet. We decided to modify all the UI of this page since it was very basic, and we added the tweet content, the hashtags it contains, the

popularity metrics and the url of the original tweet. This can be found in Figure 3.

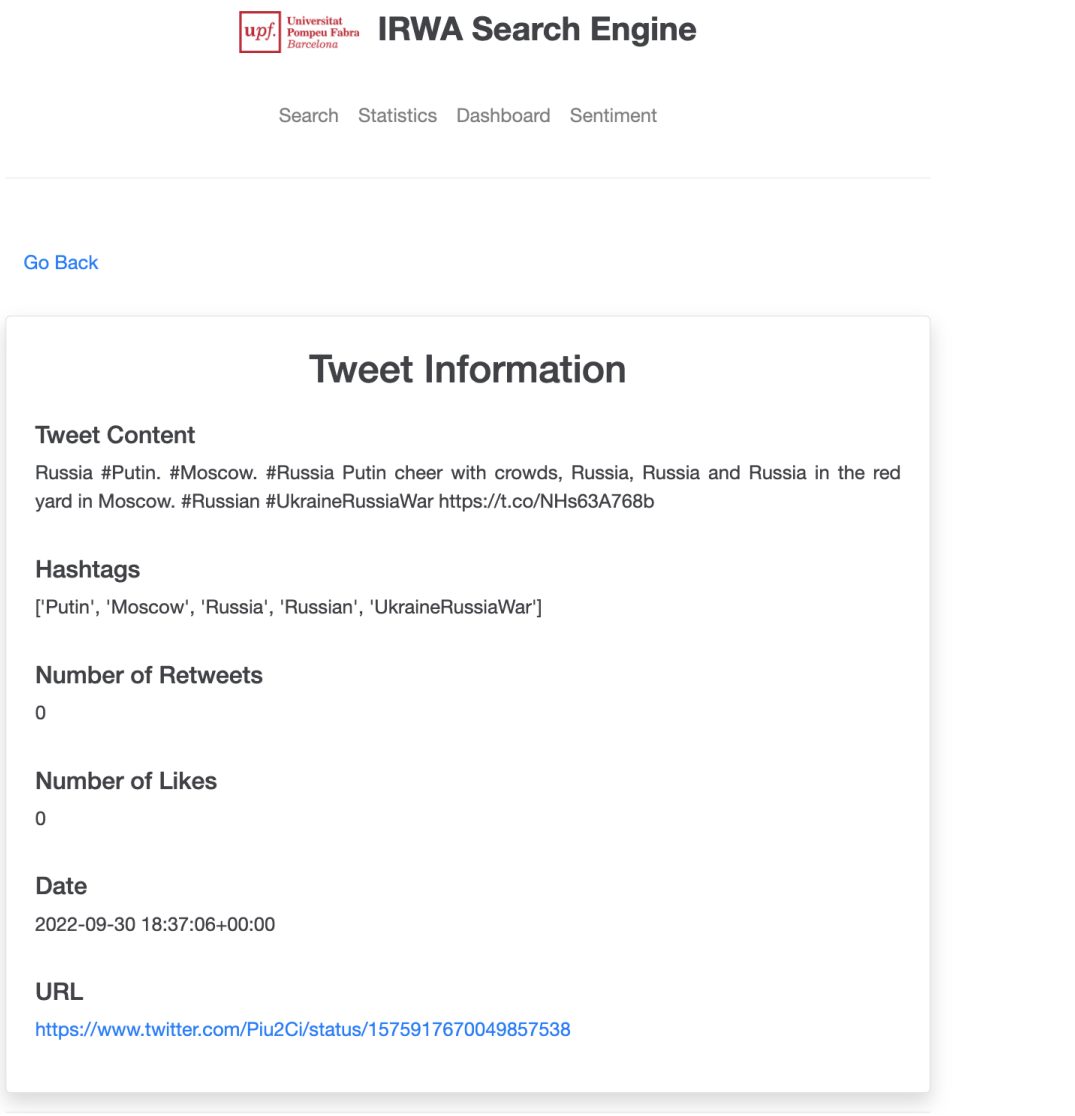

Figure 3

Statistics

In this page we mainly showed information of the current session of the user. This information is not persisted so every time the user gets out of the application or that the server is restarted, the information is lost but persisted on the Dashboard section, as we will see later.

We divided the page into three, since we computed three different statistics:

1. **Clicked docs:** The aim was to show the user how many times he/she had entered a tweet during the same session.
2. **Searched queries:** Similarly to the previous one it shows the queries the user has searched and its frequency in the current session.
3. **User information:** Retrieve and show different information of the user such from which browser the web application is running, the operating system, the date and time, the ip address. We could not get which was the location of the user due to the fact that the shown IP address is private, therefore not having access to the exact location

IRWA Search Engine

Quick Stats

Clicked docs:

(1 visits) — id: 1575917670049857538 — Russia #Putin. #Moscow. #Russia Putin cheer with crowds, Russia, Russia and Russia in the red yard in Moscow. #Russian #UkraineRussiaWar https://t.co/NHs63A768b

(1 visits) — id: 1575273715390156800 — And now 4th gas pipeline goes up in smoke - this time inside #Ukraine on the northern boundary of Kherson Oblast. What are the odds of this? #gas #RussianMobilization #WorldWar #UkraineRussiaWar #NATO #USA

(1 visits) — id: 1575825041731641346 — #russiaisateroriststate if every single one of these support the war in #Ukraine they should be charged with war crimes, they are just as guilty as #Putin #Russia #UkraineRussiaWar #UkraineWar #Russian https://t.co/6UT9QR1CTh

(1 visits) — id: 1575801868176932864 — The death toll continues to rise after the 🇷🇺artillery strike at the evacuation column of civilian cars in Zaporizhzhia (southeastern Ukraine). 24 people got killed, 36 wounded at this point. #Ukraine #UkraineRussiaWar #Russians #RussiaIsANaziState #WarCrimes #RussianArmy https://t.co/qTBwJ7l0k4

Searched queries:

Russia: 1

gas in Ukraine: 1

war crimes: 1

death people: 1

User information:

Browser: Chrome

Operating System: Macintosh

Time of the Day: 12:51:54

Date: 03/12/2023

IP Address: 151.237.210.117

Country: Unknown

City: Unknown

Dashboard

The Dashboard tab differs from the Statistics tab by presenting information from all sessions conducted on the web application. This is achieved by storing data from each session in a pickle file. To collect user information for the dashboards, functions were created in `analytics_data.py`. These functions receive data from the POST requests generated by the server during user interactions with the corresponding pages in `web_app.py`. The data is then stored both in a list of tuples and in the pickle file. Two distinct storage methods were employed to differentiate between session-specific and aggregate data.

Implemented Dashboards

In this tab, eight different dashboards were implemented in addition to the default template dashboard:

- **Ranking of visited queries:** the aim of this dashboard is to show the different queries that have been searched in our web application and the number of times. For that, we adapted the code given from the Ranking of Visited Documents.

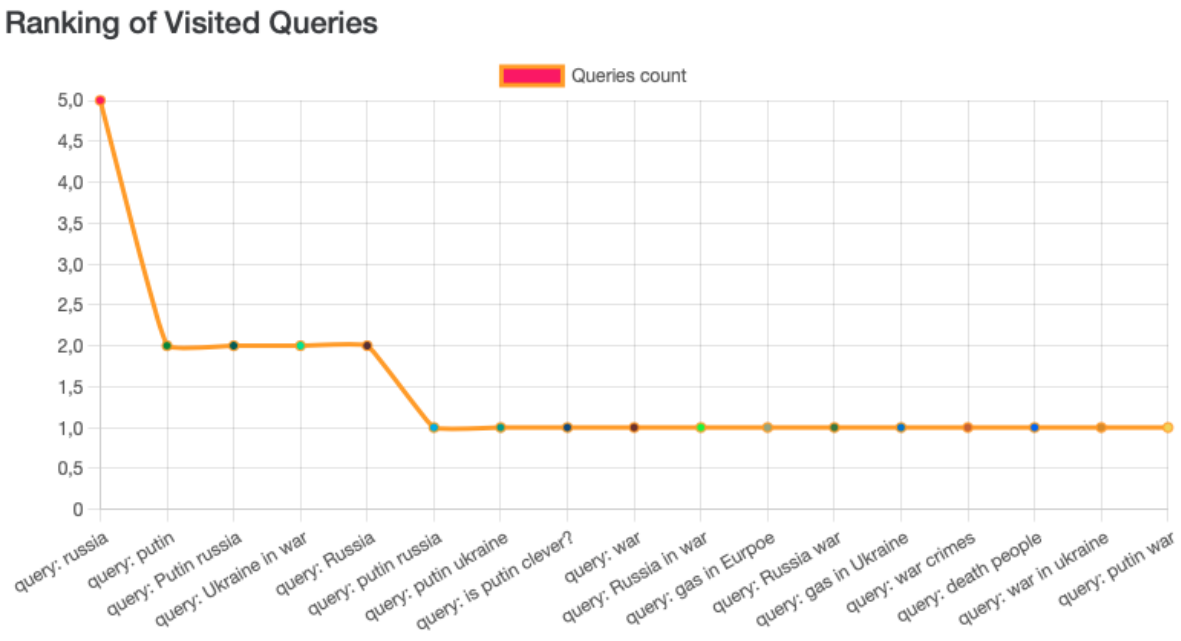

Figure 4

- **Ranking of search methods:** In this graph, we show the ranking of Search Methods. We used the same template as the graph above, but instead of being a line graph is a doughnut. See Figure 5

- **Number of Searches by day of the Week:** A bar plot which helps seeing the number of visits the web application has in every day of the week. Similar to the two graphs above, it follows the same template but changes the type of graph. See Figure 5
- **Number of users by browser:** Shows the proportion of users that have used different browsers in our web application. For that, we have also used a doughnut. See Figure 5
- **Number of users by Operating System:** Same as the graph above, but now we are interested in different operating systems and how much are used in proportion. See Figure 5

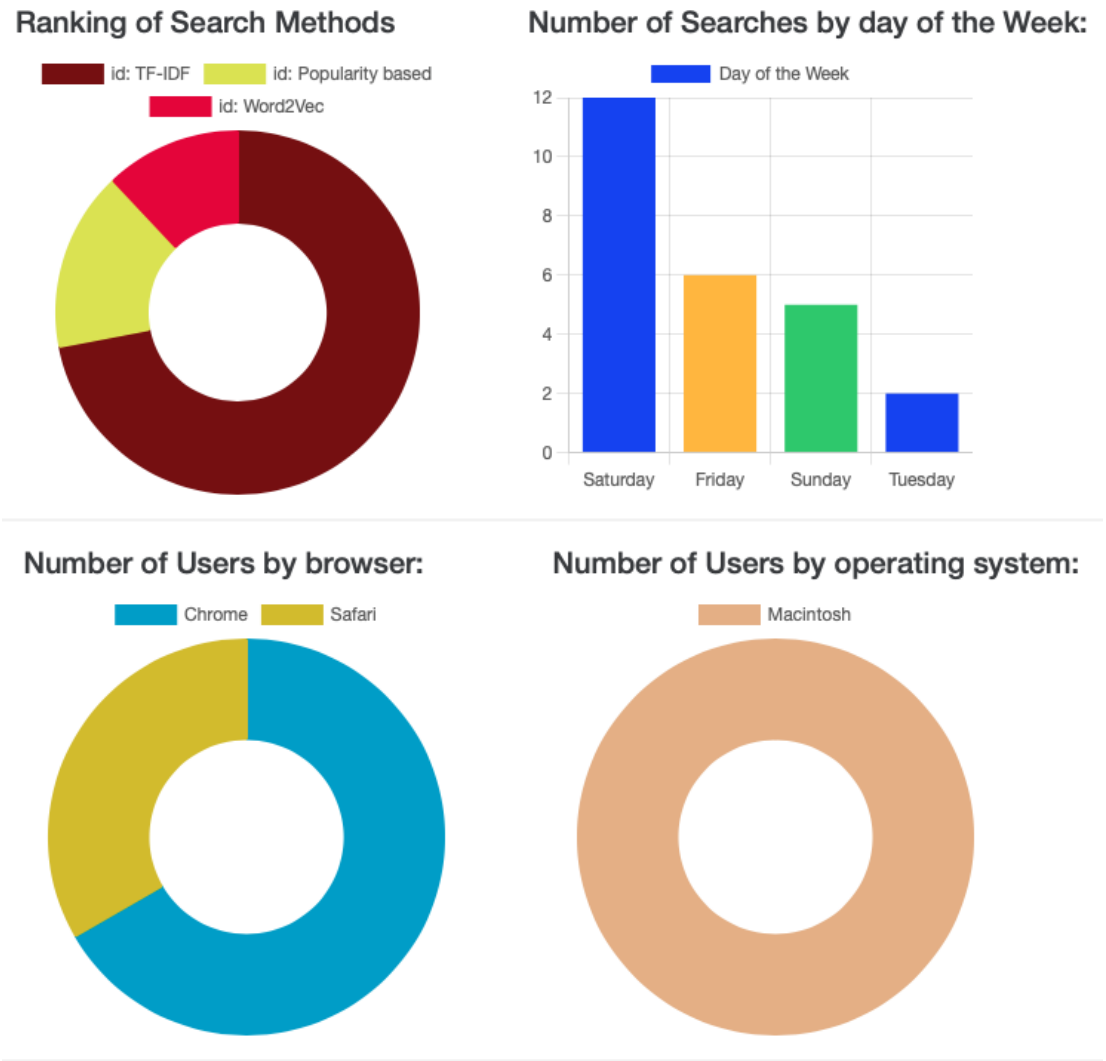

Figure 5

- **Query term WordCloud:** In this graph we show using WordCloud the frequency of the query terms of all the sessions. For that case, we used d3.js library instead of chart.js since it was easiest to implement.

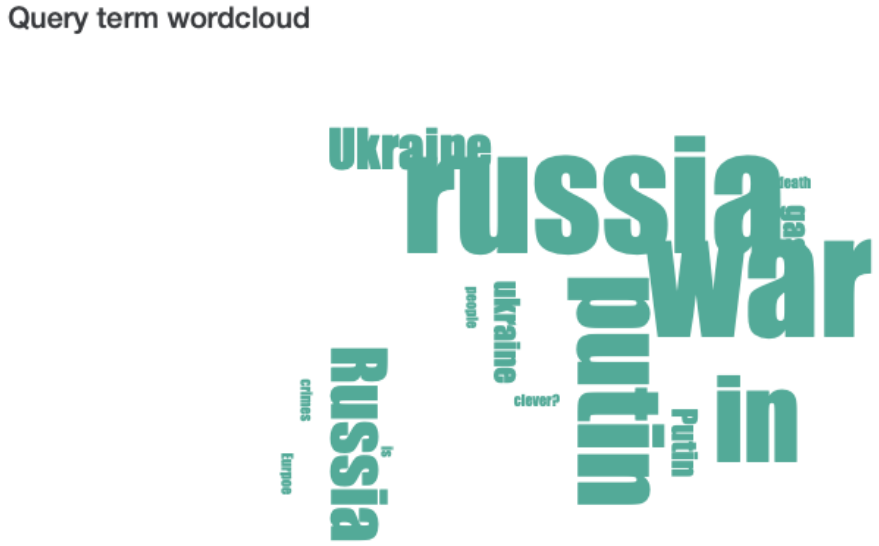

Figure 6

- **Time Spent in tweet ranking:** Lastly, using a barplot we show the time spent in the tweet detail page ranking until the user exits.