**Gerard Morales - 242781**
**Patricia Garay - 229260**
**Maren Clapers - 243397**
GROUP 102_6

Universitat Pompeu Fabra Barcelona

# Part 4: User Interface and Web Analytics

Information Retrieval and Web Analysis

## 1. Project introduction

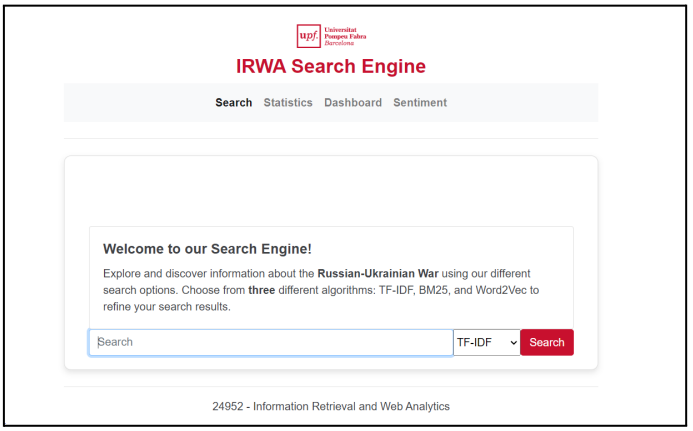
In this forth and last part, we set up the user interface (UI) of the Search Engine that we have been creating the past couple of months. This part combines all previous parts into one Web application, that let us search any query, given a specific search algorithm, to display the related tweets and see the details of each one. Moreover, we developed an Statistics tab, a Dashboard tab, and a Sentiment tab, for analyzing and tracking how users interact with the application and use our Search Engine.

## 2. User Interface

Our Search Engine Web application has a principal Search window where the user can input their queries and then we have three additional tabs (Statistics, Dashboard and Sentiment) for analyzing and tracking the behavior and sentiment of the users.
Each individual characteristic in the UI is described in the following points:

### 2.1. Search

As stated, the principal and more important tab of our Web Application. Users can input any word of phrase and the Search Engine will display all the tweets that are related with this query.




Picture 1: Search Engine Tab

One important feature in this tab is the fact the user can choose which search algorithm to use in order to obtain their desired result. Some users might find TF-IDF a more reliable algorithm while others just prefer to use the random one.

### 2.2. Algorithms used

As stated before and as we can see in Picture 1, we added a dropdown in the search bar for users to choose their desired search algorithm. We think this is a better way of proceeding than imposing the

algorithm internally and without giving the option to change. Some users might prefer the results of a specific search algorithm that provide a more accurate result for what they are looking for. The search algorithms that we implemented are the following:

- **TF-IDF:**
  This algorithm is the classic one we have seen in class and practical labs. Combines the term frequency inverse document frequency algorithm with the cosine similarity of the different terms of the query to provide decent ranked results.

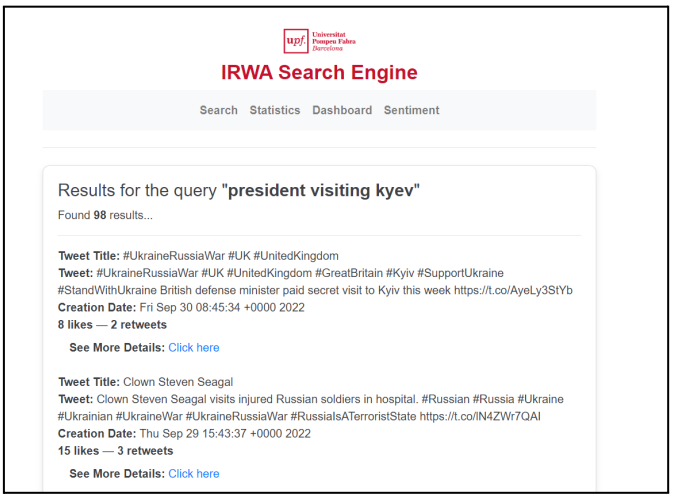
- **Random:** This "algorithm" just gives the related tweets given a query without ranking or scoring. This might be useful if you just want to see tweets related to a query and you do not mind the relevance other algorithms can determine.

- **Word2Vec:** Implemented as in the previous part of the project. It turns the tweets into word vectors through an inverted index to then perform the ranking based on the average word vectors of the query terms, scoring each tweet based on the dot product between its word vector and the query vector.

- **BM25 + tweet popularity:**
  This is our own search algorithm we invented in the previous part of the project. It is an extended BestMatch25 (BM25), where we take into account the popularity over the social network. This decision was made because BM25 is a simple and effective algorithm that we think combined with popularity information can perform even better than TF-IDF.

## 2.3. Results

Once you write the input and select your preferred algorithm, you click "Search" and this following screen will appear:



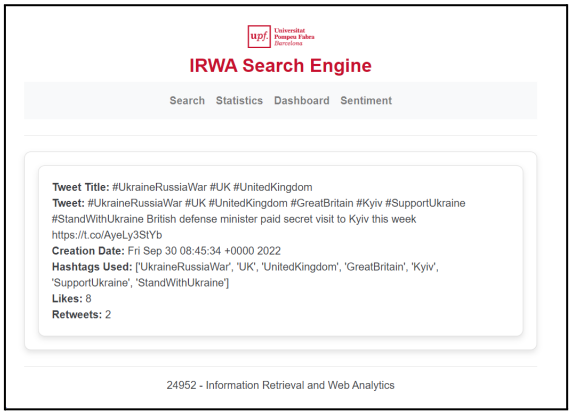

Picture 2: Search results

We firstly want to emphasize the number of matches or results with the query. This is important and might change depending on the chosen algorithm. Then we have the list of matching tweets, ordered

by the ranking and importance determined by the algorithm. Within each result we have the tweet title, the body of the tweet, its creation date, the number of likes and retweets and an additional link to see more details. We considered this information the most relevant for each tweet and reflect what most of the users might be looking for. For further information of the tweet, the user can always click on the link at the end to see more details.

### 2.4. Document details

If a user clicks on the details link, it will appear the following screen:




Picture 3: Search results

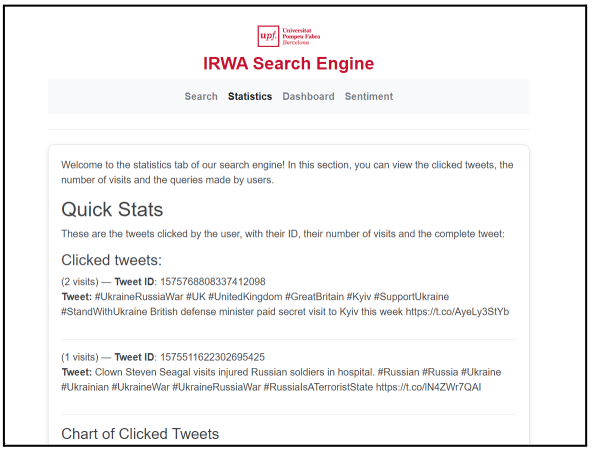
With this document details tab we can have a clearer vision of the tweet and a better understanding of it, with additional information like the hashtags used.

## 3. Web Analytics

The next tabs are meant to analyze and track the behavior and sentiment of the users.

### 3.1. Statistics Tab

The statistics tab is a useful tool to analyze and track the interactions with the Web application, seeing which are the most visited tweets and the frequency of the most searched terms and queries.

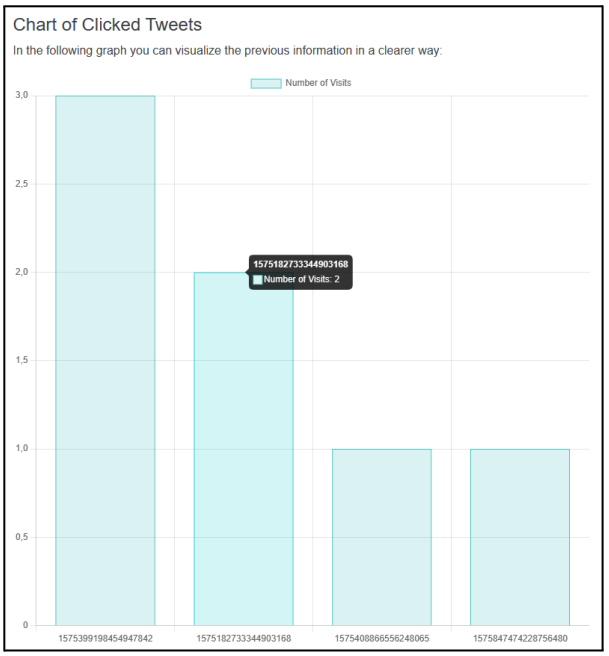



Picture 4: Statistics Tab, Quick Stats

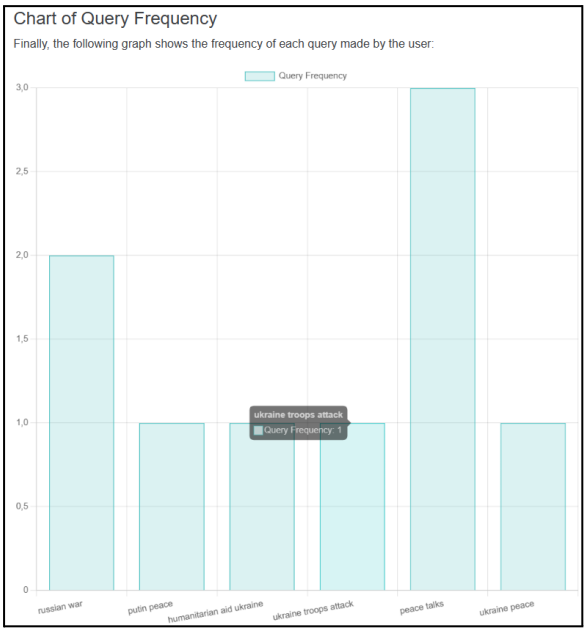
**Gerard Morales - 242781**
**Patricia Garay - 229260**
**Maren Clapers - 243397**
GROUP 102_6

The first thing we came across in this Tab is what we can see in Picture 4, which are the Quick Stats. Here we can see the most clicked tweets by the user with the actual number of clicks each tweet received. These are ordered in descending order. We have access to the tweet ID and the tweet body.

Below the Quick Stats we can find the Chart of Clicked Tweets and Chart of Query Frequency (picture 5 and 6 respectively).
In the Chart of Clicked Tweets we can see a histogram of the tweets the user has clicked the "See more details" link. This is useful to see the comparison of interest in different tweets in a more visual way.




Picture 5: Statistics Tab, Chart of Clicked Tweets




The second chart we have in the Statistics tab is the one in Picture 6. This histogram shows all the queries the user searched and the frequency of it. As well as the other histogram, it is useful to determine the different interests the user has and the most searched queries.

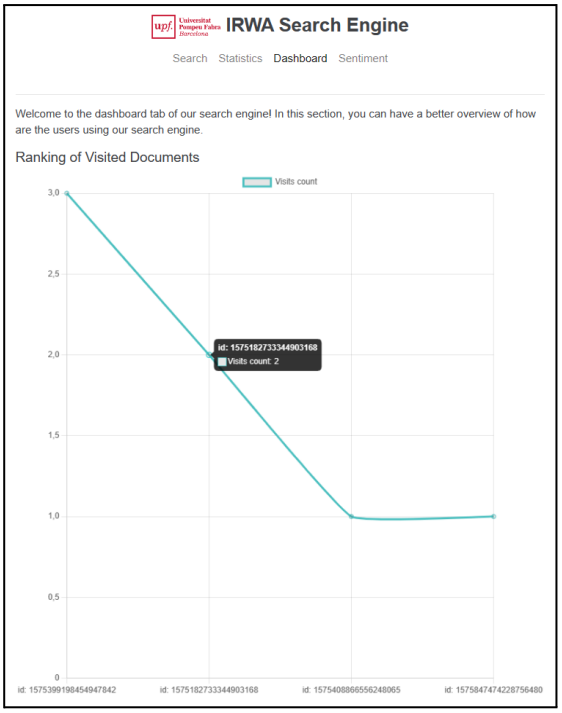
Both of these histograms are interactive, by hovering over you can see the frequency / number of visits the bar is representing.

Picture 6: Statistics Tab, Chart of Query Frequency

## 3.2. Dashboard Tab

The third tab we can find in our Search Engine Web Application is the Dashboard tab. In this section we can find a line chart (picture 7) with the ranking of visited tweets, ordered in descending order. This chart is useful to see the difference frequencies between each tweet.

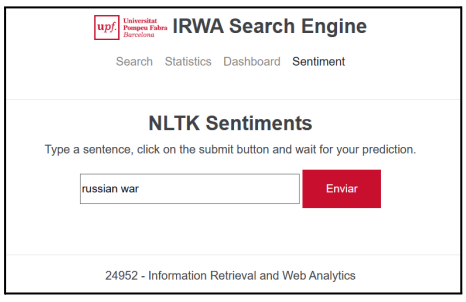
**Gerard Morales - 242781**
**Patricia Garay - 229260**
**Maren Clapers - 243397**
GROUP 102_6



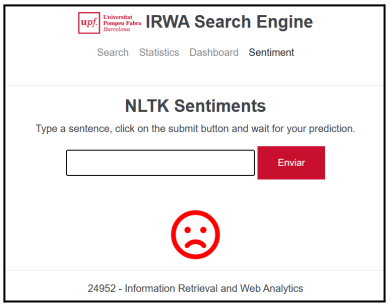

Picture 7: Dashboard Tab, Line Chart of Ranking of Visited Documents

## 3.3. Sentiment Tab

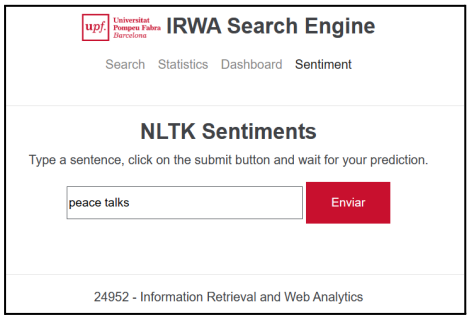
The Sentiment tab is a program that uses the Natural Language ToolKit to predict the sentiment of the input query. To test it, we input possible queries to see if the sentiment prediction is accurate



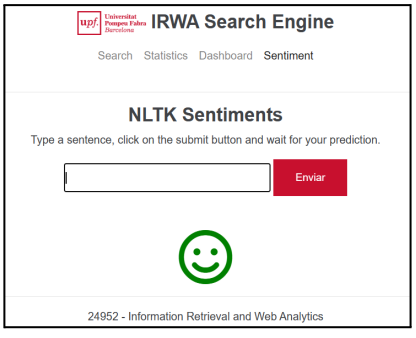

Picture 8: Sentiment Tab, “russian war” input




Picture 9: Sentiment Tab, sad output




Picture 10: Sentiment Tab, “peace talks” input




Picture 11: Sentiment Tab, happy output

As we can see in the pictures above, when we input a bad query like “russian war” which reflects the chaos and deadliness of a war, the system outputs a red sad face. But when we input a query like “peace talks” that reflect harmony and peace, the system outputs a green happy face.

## 4. Extra files used

In order to create this Search Engine Web application, we need the different files we have been developing throughout this project. We used two Jason data files, one with original data (Rus_Ukr_war_data.json) to be able to display the body of the tweets as they were uploaded originally, and the preprocessed tweets one (preprocessed_tweets.json) indispensable to apply correctly each search algorithm. We also used three pickle files (index_bm25_data.pkl, index_tfidf_data_pkl, index_word2vec_data.pkl) to store all the indices of these search algorithms.
All these files are attached in the delivery folder.

## 5. Conclusion

To sum everything up, this project has been one of the most interesting projects this year. We have been creating this Search engine from data cleaning and processing to web analytics practice after practice, seeing how different search algorithms work and applying everything into one simple but effective web application.