



**University of
Zurich^{UZH}**

Philosophical Faculty

Sprachtechnologische Webapplikationen

Sentiment Analysis on tweets regarding the best 30 universities

Name: Stylianos Psychias

Matrikelnummer: 20-743-605



Contents

1. General description of the project.....	3
Introduction:.....	3
Overview of the Web Application:	3
2. Description of your personal contributions.	5
Description:	5
3. Discussion and learnings.....	6



1. General description of the project

Introduction:

In this report, I will address the development of our learning web application for sentiment analysis of tweets using the Streamlit framework. Our web application allows users to quickly and easily get a grasp about the most important information of a large volume of tweets by using machine learning algorithms. The intend is to get to most relevant key words about the university, the number of students of each university such as the geolocation of each university.

Overview of the Web Application:

The web application is designed to allow users to retrieve the most recent tweets related to the best 30 Universities of the word (based on the QS University Ranking) in a database of 15'000 tweets (500 tweets for each university) and retrieve the key words regarding the tweets. The results of the analysis are presented in an easy-to-understand dashboard, with a breakdown of the words, as well as presenting the most famous tweets, which include the specific keywords.

As we can see in Figure 1, the user can choose, how the universities will be ordered by their rank or by their name. After the university is chosen, the basic information about the university is provided, meaning their rank in the top universities table as well as their scores and the link to the university.

In Figure 2, the user can choose within three features regarding the chosen university. He can choose, between the number of students at the university, the most frequent words appearing when people are tweeting about the university and their corresponding tweets.

Furthermore, in the second page of our webapp in Figure 3 we see a geological map, pointing, to the location of each university. Thus, there is a table with all the universities with their corresponding scores and links is provided.

To build the application, we used Python and several popular libraries, including nltk, pandas and NumPy. We used a large dataset of tweets, allowing it to accurately classify the sentiment of new tweets with a high degree of accuracy.

The web application is built using the Streamlit framework, which is an open-source Python framework that allows developers to build and deploy custom machine learning and data science applications as web apps. It has a simple, intuitive interface that made it easy for us to build the application without needing to have a deep understanding of web development or complex programming concepts.

Our motivation for our webapp, was to create something that all the student across the world could use, when they need information about a specific university. In our webapp they could find more useful information beyond the fees and the number of studies of each university.



University of
Zurich^{UZH}

Options

By Name

Choose university By Name

University of Oxford

Selected University Information

rank	name	score	link
3	4 University of Oxford	98.4000	https://www.ox.ac

Options

By Rank

Choose university By Rank

4

Selected University Information

rank	name	score	link
3	4 University of Oxford	98.4000	https://www.ox.ac

Figure 1

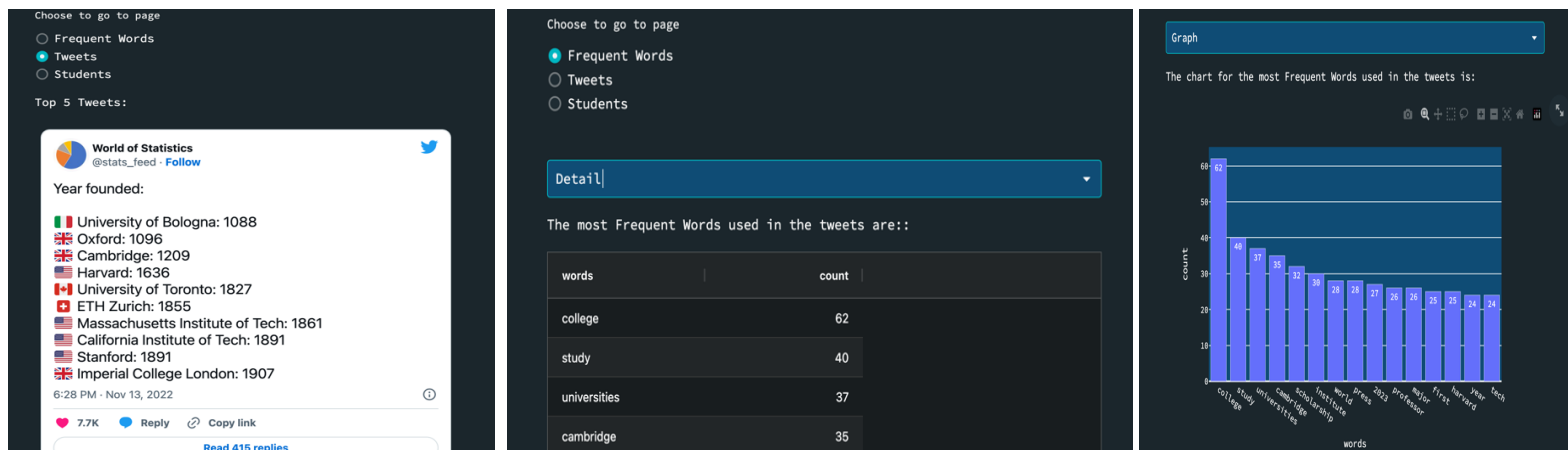


Figure 2

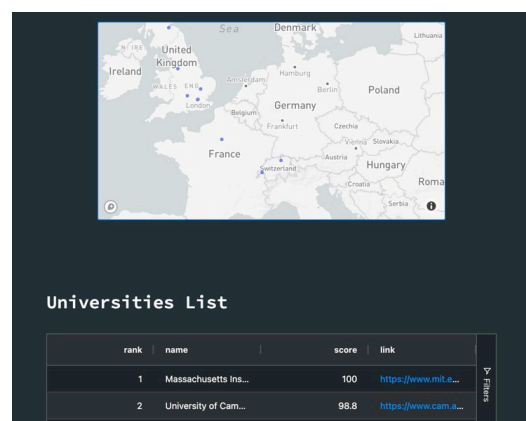


Figure 3



2. Description of your personal contributions

Description:

My contribution to the project were manifold. Initially, after our meeting with Mr. Vamvas I proposed the subject of our topic. I thought a webapp, which could analyze tweets and give important information about them could be an arduous project. Since it was an unknown topic for all of us, I had to research how we could create it. Furthermore, after I found out, that streamlit was the best option for our project, we settled on the whole idea of our project. So, I create our database (an SQLite database initially) with all the 15'000 tweets. My approach for gathering all the tweets was by using Snsrape to initially collect all the tweets. Afterwards, I applied some basic nltk methods to get the most frequent words for all the tweets.

The following steps were involved to calculate and plot the most common word frequencies: Firstly, I removed URLs or any other links, that converted all the words to lowercase letters as the words with capitalization will be different from words that are all lowercase.

Afterwards I remove stopwords with NLTK. Stopwords. These words are known as "stop words," and they contain frequently used words like who, what, you, etc. Collection words are the words that we have used to search our data from Twitter. In our case, they will be university names. I calculated and plotted the word Frequency of the cleaned tweets. To get the count of how many times each word appears in the sample, I have used the built-in Python library collections, which helps create a special type of Python dictionary. The collection counter object has a useful built-in method most common that will return the most used words and the number of times that they are used.

After having all the needed files, I assembled the initial version of our webapp.

That includes connecting the database to our script and implementing it on streamlit.

One of my contributions, was to deploy the webapp on GitLab. Since I did not have a lot of experience on it, it was quite challenging for me.

Moreover, I implemented the map on the second page of our webapp.

To create the markers on the map, I had to create a .geojson file, which contains information on the longitude and latitude of the location as well as the name of the place. Since the .geojson did not contain any location names, I had to add them manually. Then I used the map provided by Mapbox and added the layer of our own .geojson file, and plot it with the help of plotify.

In addition, I was responsible for distributing the tasks to my team members. We have split the tasks, in a way that all the people can contribute to the project based on the skill level and their motivation.



3. Discussion and learnings

Discussion:

The module web applications and language technology was quite interesting for me, since we have to be able to develop a basic web application in a short time period. That includes, learning the basics of HTML, CSS, JavaScript and databases. A quite unique module from the ones I had before, because we were called to create a skill in a specific amount of time. In the rest of the section, I will go over the challenging aspects of it.

The front-end development of the project was a challenge, as it required the creation of an intuitive and user-friendly interface for the application. We chose to use the Python library Streamlit to accomplish this task, which proved to be a powerful tool for building interactive web applications. However, its utilization also presented a steep learning curve for team members who were not familiar with it.

The database management component of the project was demanding in terms of effectively storing and retrieving the large volume of tweets that needed to be analyzed (we had a database of 15'000 tweets). The team had to ensure that the database was optimized for performance and scalability, and that the data could be easily queried and filtered as needed.

The back-end component of the project was also a matter, as it required ensuring that the application could handle a high volume of tweets without losing on effectiveness. The team had to optimize the script's time complexity and ensure that it was secure and reliable.

Finally, the linguistic component of the project was difficult due to the complexity of natural language processing and the nuances of sentiment analysis. The team had to apply various techniques such as text preprocessing, tokenization and sentiment analysis algorithms to accurately understand the context of the tweets and determine the sentiment expressed in them.

In conclusion, this was a quite exciting module, since we have learned not only how to develop our own webapps but also how to work as a team too. Organizing the team, deciding which part we should keep, distributing the tasks and be prepared on time, were only some of the difficulties that we faced as a team.