Visual Analytics

Project Report

Paul M. Magos[1]

[1]Master's Degree in AI, University of Pisa

23-05-2024

Index

1 Introduction

Social media platforms have become a rich source of data for various types of analysis, particularly in understanding public sentiment on a wide range of topics. X (formerly Twitter), with its millions of daily active users and real-time nature, provides a unique opportunity for sentiment analysis due to its vast and continuously updated stream of user-generated content. This project describes the development of a data visualization dashboard designed to analyze and visualize the sentiment of posts (formerly tweets) over time. Furthermore the dashboard allows words frequency visualization for a more in depth analysis.

The primary objective of this project is to make sense of the vast amounts of data generated on X. We utilized X's API to download posts related to our research topic ("*Lampedusa*") over a specified period. Sentiment analysis, which involves classifying textual data into categories such as positive, negative, or neutral, was employed to interpret the emotions conveyed in the posts. Furthermore, we integrated image description and video transcription models to analyze the media content within the posts.

This report outlines the methodologies used for data collection, sentiment analysis, and dashboard development. It also discusses the challenges encountered during the project and the solutions implemented to address them. The final product, a data visualization dashboard, serves as a powerful tool for researchers to monitor and interpret the dynamics of sentiments in X posts related to the topic "*Lampedusa*".

1.1 State of the Art

The state of the art in data visualization dashboards for social media analysis involves the integration of advanced analytics techniques with interactive visualization tools. Many existing platforms offer features such as sentiment analysis, keyword extraction, and trend visualization. However, few provide comprehensive media analysis capabilities, including image recognition and video transcription, and many of the existing solutions are on payment only. Recent advancements in natural language processing (NLP) and computer vision have enabled more accurate sentiment analysis and media understanding. Deep learning models like BERT and RoBERTa have shown remarkable performance in sentiment classification tasks, while image recognition models excel at identifying objects and scenes in images. We took some insights from other research platform for migration analysis such as Migration data portal[A.1].

2 Data Retrieval and Preparation

In this section, we provide a detailed description of the datasets used in this project. There are five datasets in total, four of which share the same architecture. The preprocessing steps applied to each dataset are also outlined. All the information related to the data cover a period of time which goes from 06 January 2024 to today's date 23 May 2024. It is important to mention that the data are updated continuously day by day and this description may not be as relevant in the future.

2.1 Raw Dataset

The data related to the posts was obtained through X's API, we have obtained in total to the current day 17511 posts. The data downloaded contain many information such as metadata and post media inclusion, but for the purpose of this application we will describe only the raw data itself.
Fields:

- *id*: A unique identifier of the post.
- *text*: The text of the post itself, containing also hashtags and words.
- *attachments*: Dictionary containing information about the media related to the post.
- *entities*: Dictionary containing information about URLs contained in the posts, as well as geographical ids in some cases.
- *lang*: Language of the tweet, in a ISO_639-1 format (two-letter code).
- *created_at*: Post creation timestamp.
- *author_id*: A unique identifier of the post's author.
- *geo*: GeoID for the post publication site.

Following the download of the posts, we have parsed all the information contained in the posts to select the ones containing at least one media file, and then downloaded all of them.

2.2 Sentiment Dataset

The Sentiment dataset is the raw dataset parsed and reorganized to address the task of sentiment analysis. We filtered the data contained in the original dataset, by removing the ones that had missing values, or simply empty texts. In the end

we obtained data for a total of 135 days till the date of 23 May 2024.
Fields:

- *id*: A unique identifier of the post.
- *lang*: Language of the tweet, in a ISO_639-1 format (two-letter code).
- *created_at*: Timestamp for each day present in the dataset.
- *positive*: Number relative to the total positive posts of the day
- *neutral*: Number relative to the total neutral posts of the day
- *negative*: Number relative to the total negative posts of the day
- *total*: Number relative to the total posts of the day. A sum of positive, neutral and negative fields

The sentiment is computed using two approaches:

- *NRCLex*: Library for measuring emotional affect from a body of text. Affect dictionary contains approximately 27,000 words, and is based on the National Research Council Canada (NRC) affect lexicon [5] and NLTK library's WordNet synonym sets [2].
- *Twitter-roBERTa-base*: roBERTa-base model trained on around 58M tweets and finetuned for sentiment analysis with the TweetEval benchmark. [1]

2.2.1 Processing

The first step of the Sentiment analysis classification was to clean the texts by removing citations, links, punctuation, stop words and all the special characters. We then transformed all the emoticons to a description of the emoticon through the emoji library. Furthermore we lowered every word remained in the texts to prevent mismatching on words from the NRCLex library. Finally we applied the NRCLex approach and we observed a large cluster of posts classified as neutral. To address this, we integrated the results of the roBERTa model specifically for the neutral posts. Despite this effort, a significant number of tweets remained classified as neutral.

2.3 Words Dataset

The Words Dataset is a reformatted version of the original Posts dataset, designed to facilitate more detailed textual analysis. The creation process involves the cleaning of the text content using the same process described in 2.2.1. Then we extracted only the words from each text and computed their frequencies. This involved counting the occurrences of each word on a daily basis. We filtered the data on the top 5 languages and in the end we obtained a total of 136513 words till the date of 23 May 2024.
Fields:

- *word*: The word itself.
- *frequency*: The frequency of the word in the specified day.
- *lang*: Language of the word, in common language.
- *created_at*: Timestamp representing the frequency of the word in posts on the same day.

2.4 Hashtags Dataset

The Hashtags Dataset is similar to the words datases. We performed the same process described in 2.3 filtering from the texts only the hashtags and we obtained a total of 7267 hashtags till the date of 23 May 2024.
Fields:

- *hashtag*: The hashtag itself.
- *frequency*: The frequency of the hashtag in the specified day.
- *lang*: Language of the post in which the hashtag was found, in common language.
- *created_at*: Timestamp representing the frequency of the hashtag in posts on the same day.

2.5 Images Words Dataset

The Images Words dataset is a derived dataset which we created using models that describe the Image passed in input. The model used for this purpose is Recognize Anything Plus Model [3], which is an augmentation tool for image tags, created by another tool from the same authors [4]. RAM++ improves the performance of

the previous existing base model Recognize Anything [6]. The model offers two approaches: inference on known categories and on unseen categories. Both approaches were used and their results merged for comprehensive image descriptions.
Fields:

- *id*: A unique identifier of the media.
- *tags*: The list of tags created by the first approach given by the model.
- *tags_open*: The list of tags created by the second approach given by the model on unseen categories.
- *lang*: Language of the tags, at the current time is only English for all the media.
- *created_at*: Creation timestamp of the post associated with the media

2.6 Data Loss

It should be noted that some days' worth of data was lost due to external factors such as issues with X's API. This resulted into a small gap in the data.

3 Model and Design Choices

3.1 Dashboard Goals and Research Questions

The primary goal of the implemented dashboard is to provide a comprehensive visualization of the data related to posts about "Lampedusa". The dashboard aims to address the following research questions:

- **Posts Volume and Trends**: How many posts are related to "Lampedusa" over time?
- **Sentiment Analysis**: What is the sentiment of these posts? Are they predominantly positive, negative, or neutral?
- **Language Distribution**: What languages are these posts written in, and how does the language distribution change over time?
- **Word and Hashtag Frequency**: What are the most frequently used words and hashtags in the posts, especially in the most important languages?
- **Media Analysis**: Which are the information that can be expressed from the media (images, videos) associated with the posts?

- **Words, Hashtags and Media information Distribution**: How's the frequency distribution of these derived data change over time ?
- **Filters**: How can users select different time periods, languages, and filtering models to tailor the data visualization to their needs?

3.2 Technical Design Choices

This subsection outlines the technical decisions made during the development of the dashboard.

3.2.1 Backend Infrastructure

- **Dataset computation**: All the datasets are computed, after the download from X's API on a daily basis, on a server with cronjobs which execute several python scripts with pandas manipulation:
  - Media Download and Image description.
  - Text Cleaning and Sentiment Analysis.
  - Words and Hashtags Frequency computing.
  - Push data to a private GitHub Repository automatically.
  - Subsequently a GitHub Action pushes only the aggregated files without any User information, for privacy, to the API Repository.
- **API Integration**: The API at the current day, 23 May 2024, consists in 4 endpoints implemented with Fast API and pandas:
  - "*/get_tweets*": This endpoint returns the posts counting with server arguments such as: Time Filter and Time Aggregation, Lang Filter, Sentiment Filter, and a boolean flag for data Pivoting.
  - "*/get_words*": This endpoint returns the frequency for a given source, and thus has many arguments:
    * source: Words, Hashtags, Images, Videos
    * lang: Language for the source frequencies, only the top 5 Languages can be selected (Images are only in English).
    * min_frequency: Minimum frequency of the elements to be returned, by default is -1 which will retrieve only the 2% percentile of the total data after sorting by frequency. Many elements have frequency 1 so we can filter these elements to give more space to the more frequent ones.

    - from_: Filter for first day
    - to_: Filter for last day
    - filter_type: The filtering of the data based on a given methodology. Currently this can be set to *none* or *tf_idf*.
    - aggregate: Is a boolean flag which can be set to false to retrieve the words frequency for each day in the time filter, otherwise will aggregate the results for the period.
    - pivot: This will pivot the data by having a column for each element of the source and as a value the frequency of the day. We have to set aggregate to false to use pivot.
  - "*/get_langs_words*": This endpoing will receive a source as an argument, the same ones described in the previous point, and will return the available languages for that source.
  - "*/check_words_presence*": This endpoint receives the same arguments as the "*/get_words*" endpoint and will return true if any data is present for the setting. It is a redundant endpoint and will be removed in future or modified to work without parsing the same data as the original endpoint for scalability.

The *tf_idf* we compute is a modified version of the standard score, tailored by considering the frequency of elements on a daily basis as documents. We define d_w as the total number of words in a given day, w_{df} as the frequency of a specific word in that day, D as the total number of days, and D_w as the number of days in which the word w appears. We then compute the values as follows:

$$tf(w) = \frac{w_{df}}{d_w} \tag{1}$$

$$idf(w) = \log\left(\frac{D}{D_w}\right) \tag{2}$$

$$tf\_idf(w) = tf(w) \times idf(w) \tag{3}$$

It is also worth mentioning that the API is currently deployed using Vercel[A.1], which automatically updates the code, including data updates, whenever a commit is made to the main branch.

3.2.2 Frontend Technologies and Model

The front end was developed using the Vue.js 3 framework[A.1], incorporating D3.js[A.1], Vega-Lite[A.1], ApexCharts.js[A.1], and the VueWordCloud[A.1] component for various chart visualizations. Several dashboard components are part of the Vuetify.js 3 framework[A.1], which provides beautifully crafted components. For client-server communication, we utilized the Axios[A.1] plugin. TypeScript[A.1] was used for data typing.

3.3 Data Visualization Design Choices

In this subsection, we describe the design choices made for the data visualization dashboard.

Research questions addressing To summarize briefly the implemented choices to address each research question:

1. We implemented with *Vega-Lite* a bar chart for the posts frequency with a time aggregation layer defined as *Granularity* for day, week and month.

2. For addressing the sentiment and language frequency during time we implemented with *Vega-Lite* also a stacked bar chart with the same aggregation layer of the previous point.

3. For the sentiment and language distribution change we implemented with *Vega-Lite* an area chart, which shows in a visual way the flow of each distribution. The aggregation layer is also present in this chart.

4. The words and hashtags frequency, as well as the media description were addressed by implementing three different charts. The first chart is a Bubble Chart implemented in *D3.js*, which allows also splitting the bubbles by frequency and coloring the bubbles of entities that exceed the specified frequency. The second chart is a Tree Map implemented with *ApexCharts.js*. The third chart, though less informative but still widely used, is a Word Cloud created using the *VueWordCloud* component. All the charts have many filtering options: Time Period selection, Language Selection, Source selection and Filtering Type.

5. For the frequency distribution change of the aforementioned data sources, we implemented a Stream Chart in *D3.js* which has the same filtering options described in the previous points.

All of the charts in the dashboard feature an additional layer with specified tool tips for hovering. In the Vega-Lite charts, the tool tip displays the frequency of posts on the day corresponding to the hovered bar. The Bubble Chart, Tree Map, and Stream Graph each have a tool tip that shows the frequency of the hovered word. The landing chart, when a split is selected, also allows the user to interact with the legend to filter the data in the chart, setting a low opacity for the other categories. Lastly, but most important, the Bar Chart in Vega-Lite indicates to the user, upon hovering, that they can click a bar to open the word frequency component by changing the cursor to a pointer.

Colors The colors selected for each chart are all in line with the continuity of the dashboard. The only specified color that we wanted to personalize was the colors on the stream chart, which for the big volume of the data in the words frequency resulted on a limited comprehension of the data in the charts. The resulting final Stream Chart relies only on the User Interaction with the chart, hovering on the interesting areas.

Final Consideration on Charts We observed that many charts, particularly those created with Vega-Lite, lack extensive customization options. Consequently, we have begun re-implementing the Vega-Lite charts using D3.js and we plan to replace all charts with custom implementations to provide a more tailored experience.

4 Dashboard Layout

This section provides an overview of the dashboard layout and its interactive components.

4.1 Navbar

Figure 1: Navbar

The Navbar, as we can see in Figure 1, offers navigation options to various sections: the logo takes to the home page which serves as the main landing page, the about page which takes to the information about the author of the project and the Version Selection, which includes options to navigate to the v1 landing or the v2 landing (the latter being in alpha and not discussed in this report). Lastly, it allows to select between dark and light themes.

4.2 Landing

The Landing page, seen in Figure 2, includes a Bar Chart that displays a description at the top, showing the total number of posts downloaded and the period covered. At the bottom, Figure 3, there are options to select the chart type, by default Bar, choose the granularity of data aggregation (Day, Week, Month), and split the data by posts frequency, sentiment, or language. It's crucial to emphasize that users can access the word cloud analysis for a specific period only when both none split and the bar chart options are selected. The period depends on the granularity selected which by default is week.

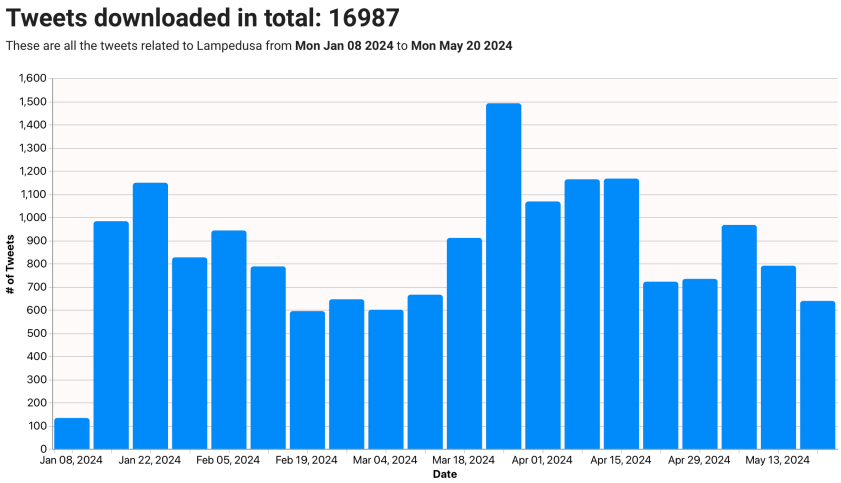


Figure 2: Landing page

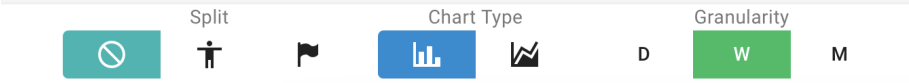
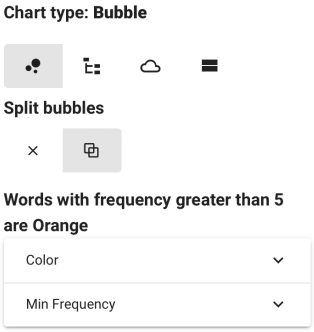


Figure 3: Landing page Chart options

4.3 Word Cloud

The Word Cloud is presented in a modal that opens when a bar on the bar chart is clicked (in default setting). The modal, Figure 4, is divided into two sides. On the left side, it displays the selected chart type, default is Bubble Chart). On the right side, it offers various options for filtering and aggregating data. The left side includes a calendar for selecting the time period for source frequency analysis, options to choose the source to view (words, hashtags, or images), a language filter, and a frequency filter type. The right side includes buttons for selecting the chart type (Bubble, Tree, Word Cloud, or Stream Chart). Only for the Bubble Chart, there are additional data transformation options, which let the user split the data into two groups based on a minimum frequency and customizing the color for bubbles that exceed this frequency, we can see it in Figure 5. Finally only the Stream Chart visualizes the

Figure 5: Bubble chart additional options



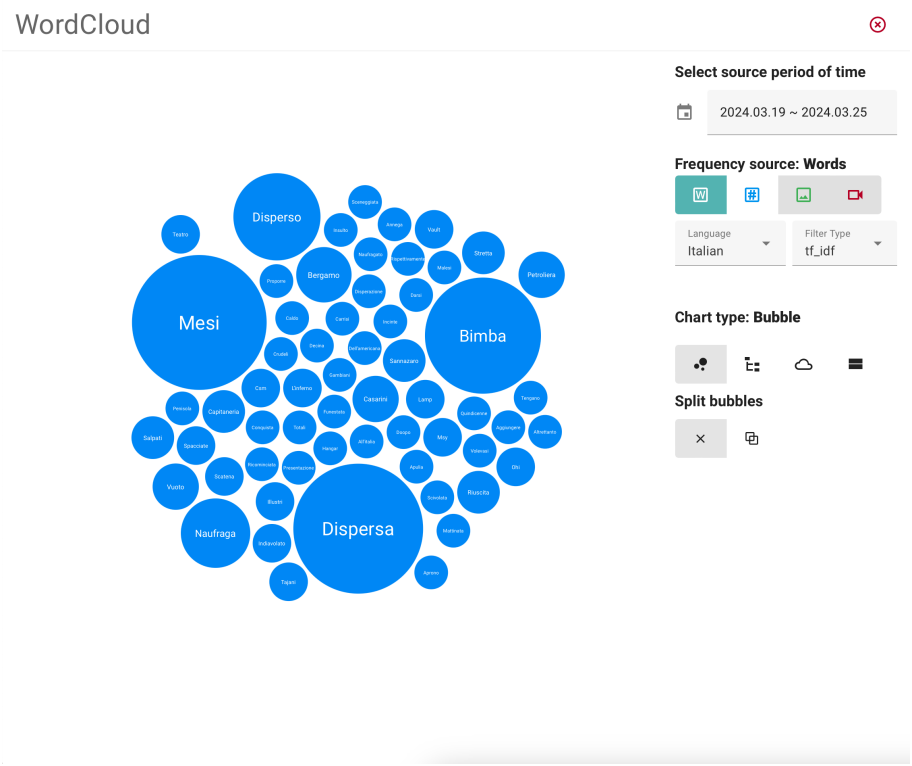


Figure 4: Word Cloud Modal

distribution changes of source frequency over time without aggregating the data, which will fail on showing data if the period is of one day only.

Final Consideration In all charts except the Word Cloud, tool tips have been incorporated to aid users in comprehending the entity and its corresponding frequency within the selected chart. Particularly in the Stream Chart, tool tips are essential as the words within the chart areas would otherwise be challenging to read and thus rendered ineffective. The splitting feature in the Bubble Chart was introduced with the vision of enabling users to filter posts in the future, distinguishing between those related to migration or tourism.

5 Use Cases

In this section, we present various use cases of our dashboard application, highlighting its functionality and potential benefits.

5.1 Use Case 1: Sentiment Analysis

Scenario A research aiming to gain insights into public sentiment and trending languages related to the keyword "Lampedusa" over time will interact with the chart present in landing page.

User Actions

1. The user navigates to the home section of the dashboard.
2. Selects the desired time aggregation period for analysis using the Granularity aggregator.
3. Chooses Sentiment or Language as the data split kind.
4. Selects the appropriate chart type (e.g., Bar Chart).
5. Interacts with the chart to explore trends and volume of posts over time, by looking at the tool tips provided for each stack of the bar chart.

Users can achieve the same interaction by opting for the Area Chart, providing valuable insights into distribution changes over time. Additionally, users can interact with the legend by clicking on various labels, thereby adjusting the opacity of other categories in the chart. For advanced interaction, users can press the shift key while clicking on multiple labels to focus on multiple categories simultaneously. To remove the filter the users will click on the legend where no labels are present. Some examples of visualization can be seen in the Appendix Figures 6-8.

Expected Outcome The user gains insights into the sentiment distribution and volume of posts related to "Lampedusa" over the selected time period, in decision-making and trend analysis.

5.2 Use Case 2: Hashtags Frequency Analysis

Scenario A researcher wants to analyze the frequency of specific hashtags used in posts about "Lampedusa" to identify key topics and discussions.

User Actions

1. The user navigates to the word frequency analysis section of the dashboard by clicking on a bar of the landing chart relative to a period of time.
2. Chooses hashtags as the data source.

3. Selects the appropriate chart type (e.g., Bubble Chart).

4. Interacts with the chart to explore word frequencies.

5. Additionally the user can:

    - Change the period of time and see how the frequency of a specified element changes.
    - Change the language and see how the frequency of a specified element changes between languages.
    - Split the data by a minimum frequency and see the top 5 elements in the specified period.

The same process can be applied for all the other sources of frequencies. Furthermore, the user in the Bubble Chart can split the data by a minimum frequency and see the top 5 elements in a specified period.

Expected Outcome The researcher obtains visual representations of hashtags frequencies, facilitating a deeper understanding of the topics discussed in posts about "Lampedusa" over time. Some examples are presented in Appendix Figures 9, 10.

5.3 Use Case 3: Word Distribution Analysis

Scenario A researcher aims to analyze the distribution of word frequencies related to the topic "Lampedusa" over different time periods.

User Actions

1. The researcher accesses the word distribution analysis section of the dashboard.

2. Selects the desired time period for analysis using the calendar feature.

3. Selects the appropriate Stream chart.

4. Interacts with the chart to explore changes in word frequency distribution over time, by hovering the interesting areas to gain information about the word.

Expected Outcome The researcher gains insights into the distribution of word frequencies associated with "Lampedusa" across various time intervals. This analysis helps in identifying trends and patterns in word usage over time. We can see some examples in Appendix Figures 11, 12.

6 Conclusion

This project has presented the development of a comprehensive data visualization dashboard tailored for analyzing social media posts related to the topic of "Lampedusa." By integrating advanced analytics techniques with interactive visualization tools, the dashboard provides researchers with valuable insights into public sentiment, language trends, word frequencies, and media content over time.

The primary objective of this project was to make sense of the vast amount of data generated on social media platforms like X (formerly Twitter). Through the utilization of X's API, we collected and analyzed posts related to "Lampedusa" over a specified period, employing sentiment analysis, word frequency computation, and media analysis techniques.

The final product, a data visualization dashboard, serves as a powerful tool for researchers to monitor and interpret the dynamics of sentiments, language distribution, and word frequencies in social media posts related to "Lampedusa." By providing interactive features and customized options, the dashboard enables users to tailor the visualizations to their specific research needs and gain actionable insights for decision-making and trend analysis.

In conclusion, this project demonstrates the potential in contributing to a deeper understanding of public discourse and sentiment on important topics. The project code is available on GitHub: `https://github.com/PaulMagos/frontend.git`, `https://github.com/PaulMagos/XResearchBackend.git` for the Frontend and the Backend respectively. The posts retrieval code is currently on a private repository which will be probably released during the proceeding of the research project.

6.1 Future Works

In the future we aim to implement all the charts presented in this report with *D3.js* library, as cited in 3.3, to have a more ad-hoc solution for the analytical task and also to maintain a continuity between animations. Furthermore we aim to add more filters in the word analysis such as minimum words frequency to retrieve. Moreover we want to allow the users to open the words analysis modal for a specified sentiment in the sentiment stacked bar chart.

A Appendix

A.1 Links

- Vercel: `https://www.vercel.com`
- Vue.js 3 framework: `https://vuejs.org`
- D3.js: `https://d3js.org`
- Vega-Lite: `https://vega.github.io/vega-lite/`
- ApexCharts.js: `https://apexcharts.com`
- VueWordCloud: `https://github.com/SeregPie/VueWordCloud`
- Vuetify.js 3 framework: `https://vuetifyjs.com/`
- Axios: `https://axios-http.com`
- Typescript: `https://www.typescriptlang.org`
- Migration Data Portal: `https://www.migrationdataportal.org`

A.2 Images

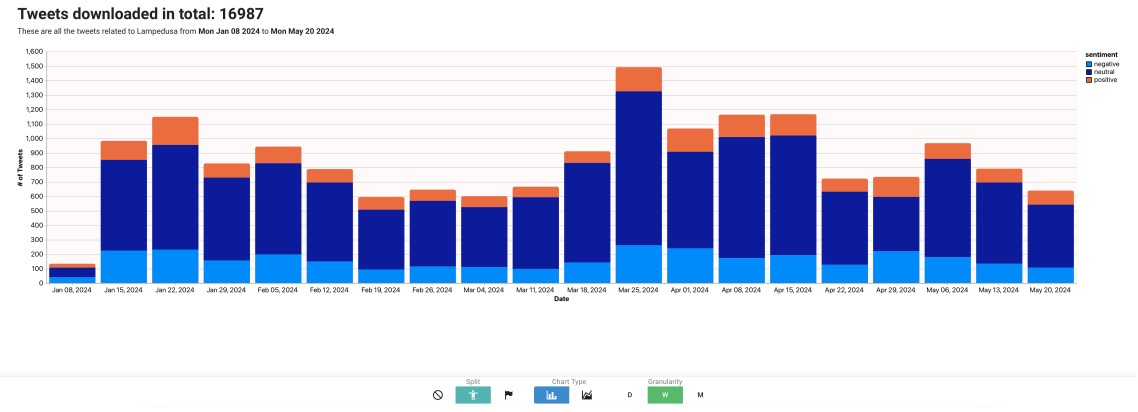


Figure 6: Sentiment split of posts frequency

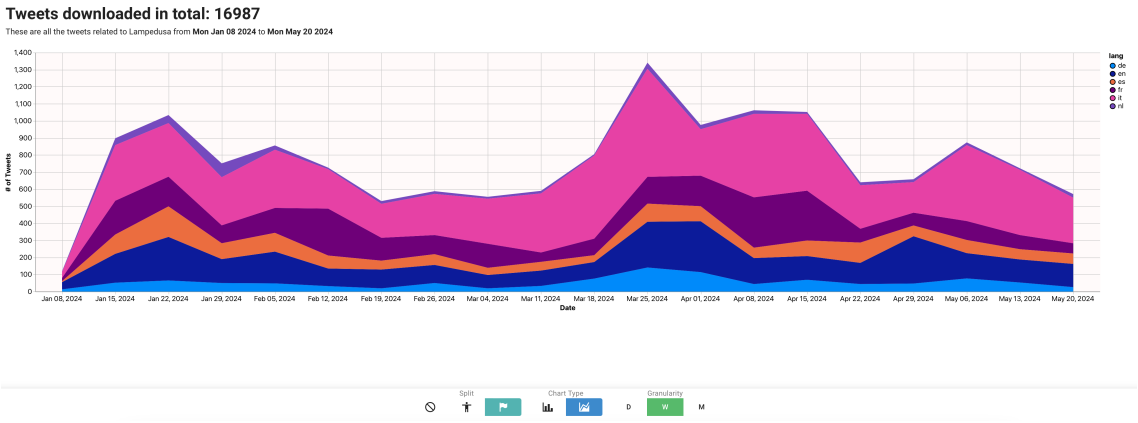


Figure 7: Language split of posts frequency on area chart for distribution analysis

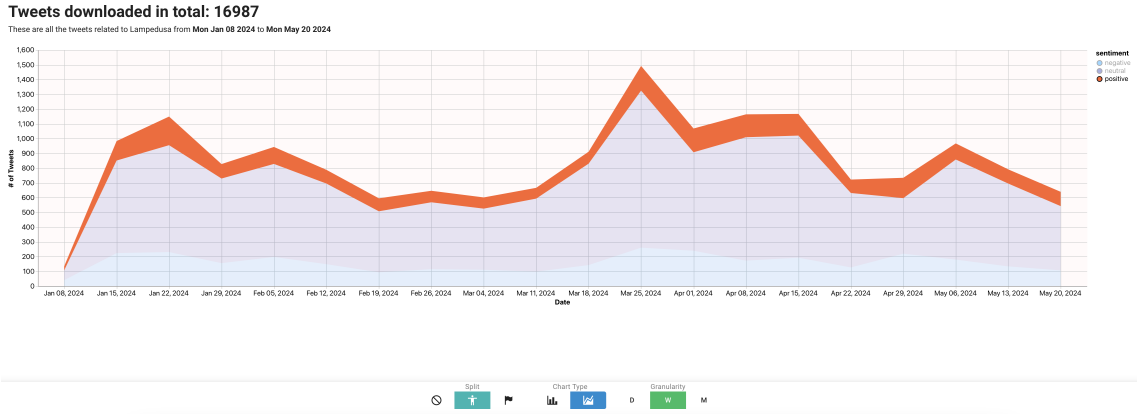


Figure 8: Sentiment split of posts frequency on area chart for distribution analysis, with focus on positive tweets

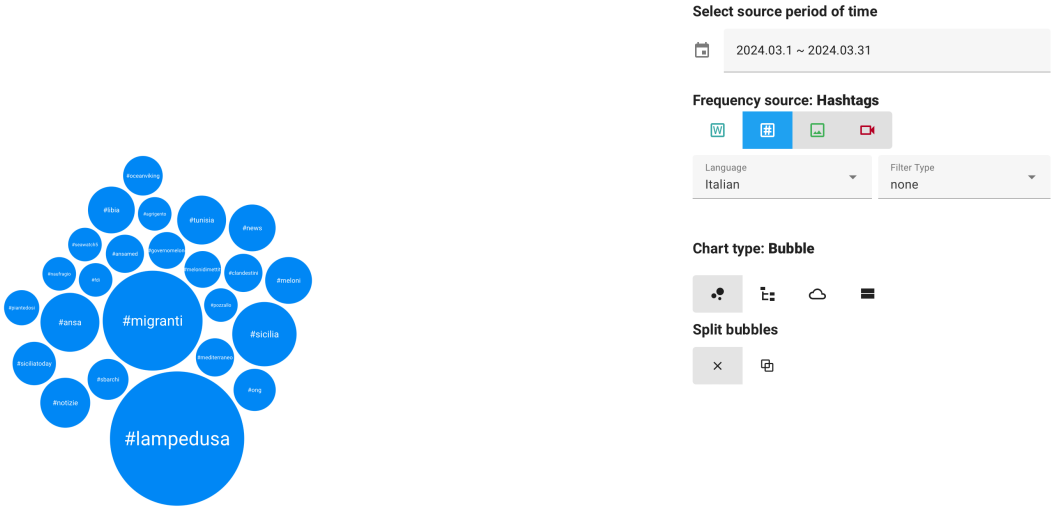


Figure 9: Frequency of hashtags in Italian in March

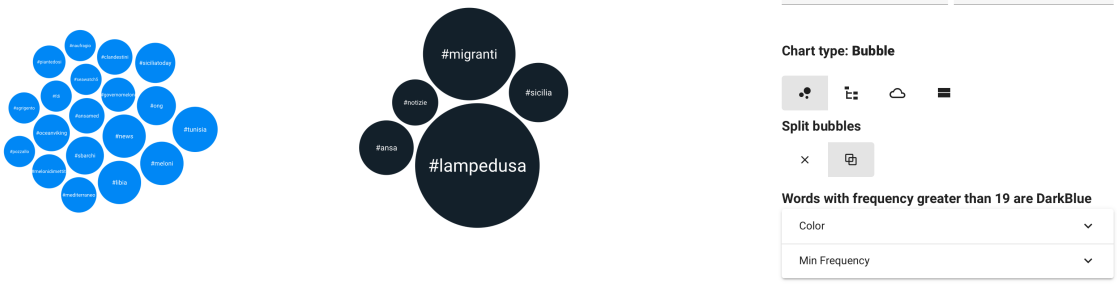


Figure 10: Top five hashtags in Italian in March

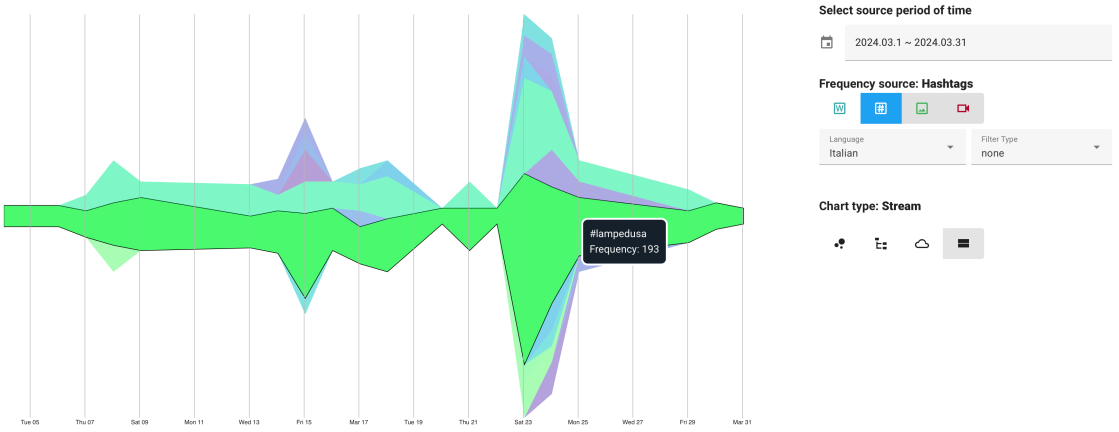


Figure 11: Hashtags frequency distribution over time with focus on most common

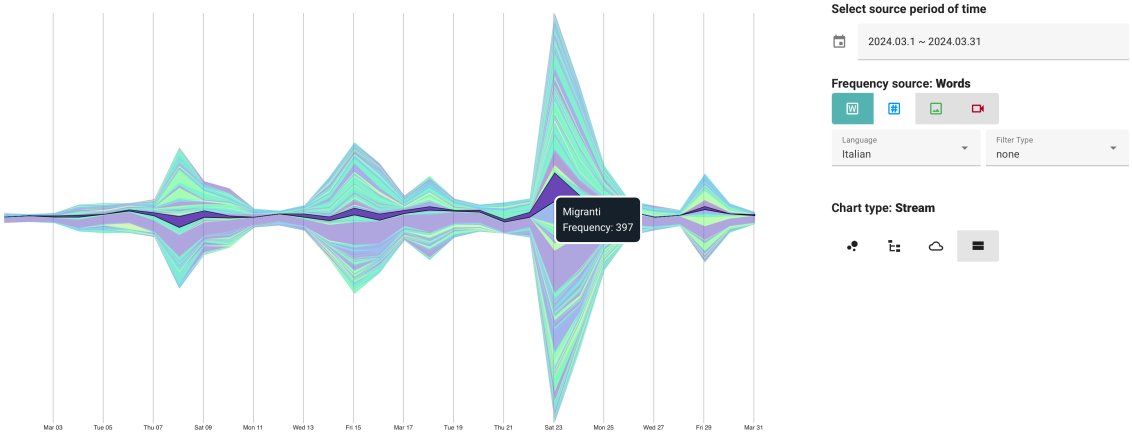


Figure 12: Words frequency distribution over time with focus on second most common

References

[1] Francesco Barbieri et al. "TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1644–1650. DOI: 10.18653/v1/2020.findings-emnlp.148. URL: https://aclanthology.org/2020.findings-emnlp.148.

[2] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.*" O'Reilly Media, Inc.", 2009.

[3] Xinyu Huang et al. *Open-Set Image Tagging with Multi-Grained Text Supervision*. 2023. arXiv: 2310.15200 [cs.CV].

[4] Xinyu Huang et al. "Tag2Text: Guiding Vision-Language Model via Image Tagging". In: *arXiv preprint arXiv:2303.05657* (2023).

[5] Saif M. Mohammad. "Word Affect Intensities". In: *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*. Miyazaki, Japan, 2018.

[6] Youcai Zhang et al. "Recognize Anything: A Strong Image Tagging Model". In: *arXiv preprint arXiv:2306.03514* (2023).