Sentiment Analysis for news publications and its evolution over time

Rafael Varela
up201706072@edu.fe.up.pt

Tiago Verdade
up201704003@edu.fe.up.pt

José Silva
up201705591@edu.fe.up.pt

May 26, 2021

Abstract

News publications inherently express an underlying sentiment regardless of how unbiased they are. Different news publications about the same topic can have different opinions which influences the readers. It is important for a reader to know whether a given publication is biased in certain topics, thus with the help of sentiment analysis Arquivo de Sentimentos aims to expose how opinions of news publications shift over time and possibly detect biases.

1 Introduction

Sentiment analysis often referred to as opinion mining, is an increasingly popular instrument for the analysis of media discourse. The platform provides a way of finding out the polarity or strength of the opinion, either positive or negative, that is expressed in written text. In the context of our project, it is the online content of news media [1].

1.1 Motivation and Goals

Nowadays the users of the Internet have access to several resources people could only dream of decades ago. In the fast age we are living in, it can be challenging to keep up with recent events and even harder to keep track of how things evolve. With so many different source, it also becomes troublesome to know which are credible and which are not.

The project proposed in this article shall provide a means to analyze the opinions, using sentiment analysis, of news media (comparing them) over the years which will help the users form a more informed opinion on their own and avoid misinformation as a result. *Arquivo de Sentimentos* shall also be capable of bias detection in online publications. By comparing the underlying sentiment and how often they appear in given publications a pattern can be established exposing the writers' biases and how they shift over time.

2 State of the Art

The application will merge two different very powerful technologies – sentiment analysis and web scraping. Although there are some tools for those two specific purposes, separately, there are only a few that combine both.

There are some capable platforms which use those technologies (such as *Aylien* [2]) for different purposes such as social media monitoring, customer support, brand monitoring, market research and so on. Not only those applications do not focus on the same goals as *Arquivo de Sentimentos* (i.e. bias detection about different news sources and how they evolve), but most of them can be very expensive (> 30$/month).

For instance, the text analysis startup *Aylien* [2], uses deep learning and NLP algorithms to parse text and extract intel from documents for its customers. This tool differs from *Arquivo de Sentimentos* by focusing more on detecting trends and only providing an API for its costumers, while our goal is to provide an easy to use web platform for everyone. *Arquivo de Sentimentos* does not only focuses on what is happening now, retrieving also information about the past. Besides, the application shall be free to use, whereas the *Aylien* product costs a minimum of $49 per month.

3 APIs

For this project we intend two make use of two different API's and integrate them with our own tools to provide the end-user a useful and robust product.

3.1 Arquivo.pt API [3]

The Arquivo.pt API allows full-text search and access preserved web content and related metadata by URL, accessing all versions of preserved web content [3].

This API will be used for searching for topics and retrieving related web content from predefined news sources (e.g. "Jornal de Notícias", "Diário de Notícias", "Público", etc), so we can then parse and analyse those news.

3.2 Google Natural Language API [4]

The Cloud Natural Language API is a part of the Cloud Machine Learning API family. It provides various language understanding tools, for example sentiment analysis, entity analysis, entity sentiment analysis, content classification, and syntax analysis [4] .

These tools are meant to analyze the sentiments of the news retrieved by the Arquivo.pt API and then display its results in a user-friendly manner.

4 System Requirements

In order to define which features the system should have, the following user stories were produced:

- As a user, I want to search for an entity so that I can visualise its sentiment analysis results over time.
- As a user, I want to pick different sources so I can compare their views on a given entity.
- As a user, I want to select different dates ranges so that I can analyse the results of the sentiment analysis on a given time window.
- As a user, I want to search for more than one entity so that I can compare the results of sentiment analysis between them.
- As a user, I want to export the results of the analysis so that I can use them in an external environment.
- As a user, I want to visualize some of the news used for the sentiment analysis.

5 Architecture

5.1 Physical architecture

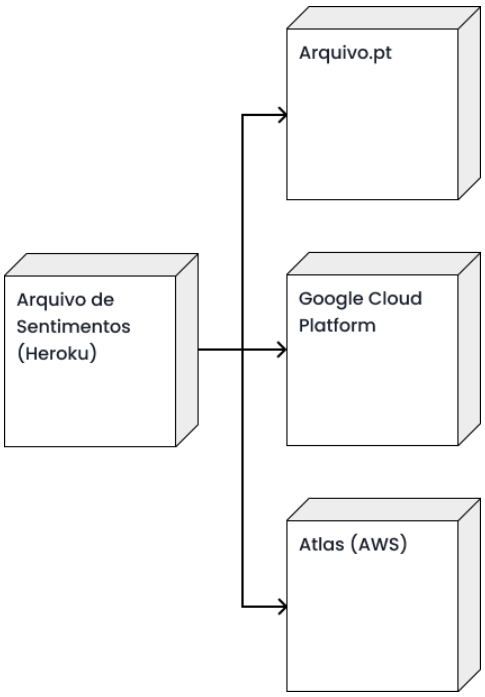

Figure 1: *Physical architecture diagram*

Arquivo de Sentimentos, which is hosted on *Heroku* [5], physically depends on 3 components - *Arquivo.pt*, *Google Cloud Platform* and *Atlas*.

- *Arquivo.pt* hosts the news and the metadata.
- *Google Cloud Platform* is used for their SaaS [1] solution for Sentiment Analysis.
- *Atlas* hosts the *MongoDB* database, using *AWS* that is used as a cache.

[1] Software as a Service

5.2 Logic architecture

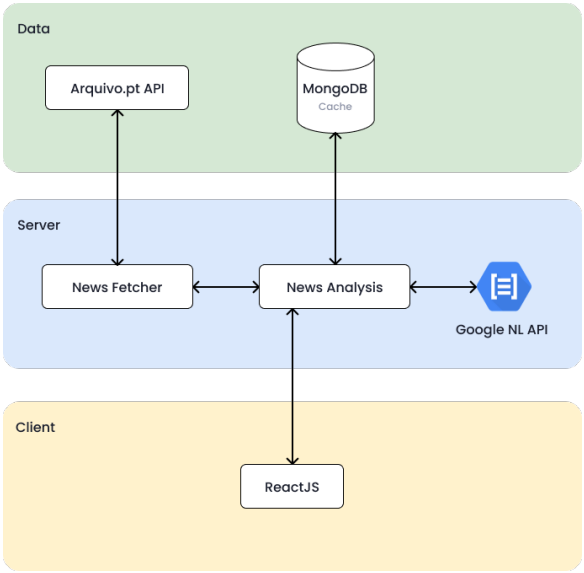

Figure 2: *Logic architecture diagram (high level)*

The system follows a layered architecture style, composed by three different tiers: **client**, **server** and **data**.

5.2.1 Data Layer

Responsible for retrieving the data necessary for the application.

The ***Arquivo.pt API*** allows the server to fetch information about the news hosted by *Arquivo.pt*.

The ***MongoDB*** database acts as a cache to store results of sentiment analysis, so that the server doesn't need to perform the analysis every time its results are requested.

5.2.2 Server Layer

Contains the necessary components to fetch and analyze news, and it is built using ***Flask***.

When the ***News Analysis*** module receives a request, it tries to serve the results from the MongoDB database. If there's a cache miss, then the request is passed to the *News Fetcher*.

The ***News Fetcher*** queries the Arquivo.PT API [3] and parses the retrieved news. Then the *News Analysis* connects to Google's Language API to perform Sentiment analysis [4]. Finally, the results are stored in the *MongoDB database*.

All this process is made **asynchronous** with the support of ***Celery***, improving the application performance and its responsiveness.

Saving and serving from the *MongoDB database* helps to **save costs** on the Google Cloud platform and **allows users to be rapidly served**.

5.2.3 Client Layer

Built on top of ***ReactJS***, it is responsible to render the user interface, uses ***Axios*** to establish a connection to the **server** layer. Every time the user performs a query, *Axios* requests the data from the *News Analysis* module.

6 Development Details

6.1 Querying Arquivo.PT API

The text search API is queried, using the endpoint `https://arquivo.pt/textsearch` with the following parameters:

```
parameters = {
      "q": entity,
      "siteSearch": source,
      "from": year+"0101000000,
      "to": year+"123100000"
    }
```

the *entity* parameter corresponds to the one being searched by the client; *siteSearch* indicates the website whose news the clients wants to query from ("Jornal de Notícias", "Diário de Notícias", "Público"); *from* corresponds to the initial date for the time span of the search and the *to* sets the end date.

The endpoint returns a JSON object with each news publication found in the following format:

```
{
    "title" : Title of the Publication,
    "originalURL" : Original URL
        of the Publication,
    "linkToArchive" : URL of
        the Publication maintained
        on the Arquivo.pt Archive,
```

```
    "tstamp" : Timestamp of
        the Publication,
    ...
}
```

6.2 Content Extraction from the News

Using the URL of each publication the python library *Newspaper3k* [6] allows our platform to **scrape the content** of those publications.

With the aid of the *Beautiful Soup* [7] library, the platform **fetches the previews of the URLs** for each news publications.

Both of these libraries make use of the HTML tags and semantics to properly parse web content.

6.3 Querying Google Natural Language API

The natural language API is queried using the following parameters:

```
parameters = {
    'document': {
        'content': text_content,
        'type_': Document.Type.PLAIN
                                _TEXT,
        'language': 'pt'
    },
    'encoding_type': EncodingType.UTF8
}
```

where the *text_content* contains all the contents of the news about an entity in a specific news source over a period of one year.

For each request, the API responds with a JSON object with the following format:

```
{
  'documentSentiment': {
    'magnitude': 'magnitude_score',
    'score': 'sentiment_score'
  },
  'language': 'pt',
}
```

where the *magnitude_score* represents the magnitude of the sentiment present in the news and *sentiment_score* represents the score of the sentiment.

6.4 Displaying the results

The frontend was built on top of the ReactJS framework for its high **development speed**, fueled by its relatively **low learning curve**, a considerable amount of resources and guides online and access to countless packages through *NPM* that simplify implementing certain features.

6.5 Optimizing the user experience

Profiling the web server, Google's API turned out to be the **bottleneck**, the solution was to **minimize** and **parallelize** the requests. Otherwise, a single request could hold the whole web-server for more than **6 minutes**.

MongoDB helps minimizing the amount of requests to the Google API, by serving as a cache. On a cache hit, the average response time is 85ms.

Celery [8] solves the parallelization part of the optimization. If a request must query the Google API, then it is put into Celery's task queue and runs in parallel, thus the web-server does not block, allowing more users to be served.

6.6 Deploying on Heroku

Arquivo de Sentimentos depends on lots of different services: **MongoDB**, **Google API** and a **Celery cluster**. Due to having lots of moving parts it was important to develop the application with *flexibility* in mind.

Although most of the integrations work fine by including secrets in a *.env* file, Google's API needs a path to a file which contains the credentials. Putting the credentials into the repository was not an option due to not being safe, bad actors could easily abuse our cloud credits. Passing the contents of the file as a string and writting it to disk as soon as the application started was not an option too, since Heroku Dyno's restart upon any write.

The solution was to use a buildpack [9], a Heroku worker that generates the file even before the code is deployed.

7 Application Overview

Upon opening *Arquivo de Sentimentos* the user is greeted with a search box and two graphics (Annex 3).

The **search box** allows the user to customize the sources to be queried, the time-frame and the keywords to be searched. Currently there are 3 supported publications' sources:

- Correio da Manhã
- Público
- Jornal de Notícias

After the news being fetched, the graphics are then populated (Annex 4).

The first graph shows the **score of the sentiment**, a value between -1 and 1, which corresponds to where the emotion leans to - negative values represent negative emotions while positive values represent positive emotions.

On the other hand, the second graph displays the **magnitude of the sentiment**. The magnitude is a value between 0 and $+\infty$ and indicates how strong the emotion is. The value is not normalized because every expression of emotion is accumulated, thus the longer the text the higher the magnitude can be. This factor is highly important because it can expose the propaganda aspect of articles masking as news.

To understand the results the user can use the following table as reference:

Sentiment	**Score**	**Magnitude**
Clearly Positive	0.8	>3
Clearly Negative	-0.6	>4
Neutral	0.1	0
Mixed	~0	>4

After having the results from the sentiment analysis, the user can also **visualise some of the news** that were used for the analysis, by clicking on the news icon on the right of the screen (Annex 5). With the use of the check boxes on the search box, this news can be filtered by entity and even by source.

One of the core functionalities of the platform is also the ability to **export the results** in *csv*, *png* or *pdf* formats (Annex 6), so they can be integrated in other services or used for research purposes.

Just in case the user is having trouble thinking about entities to be searched, by clicking on the "Examples" button on the navigation bar, the has access to a few **examples that will automatically perform the query** and display the results (Annex 7).

If the user doesn't want to wait for new entities to be analyzed, he can just search for some that were already analyzed, by making use of our **auto complete** feature (Annex 8).

It is also worth to note that the application supports both **English and Portuguese** (by choosing the "Language" on the navigation bar). Furthermore, the user has also access to the **about page**, which contains some information about the context and goals of the application, how it works, and links to the team members (Annex 9).

8 Results

As envisioned, *Arquivo de Sentimentos* allows its users to search for entities and visualise the results of the sentiment analysis performed on news from different sources. Not only does the platform performs analysis, which in itself is a topic not much explored in Portugal, but also provides its users with a panoramic view of the different publications about a given entity.

8.1 Social Impact

News are a very important aspect of society, they keep a person informed about worldwide events. As such, they have the power to influence how their readers think and their opinions. *Arquivo de Sentimentos* seeks to raise awareness about the underlying sentiment present in every news article, enabling the readers with more information and even motivating them to consume news from different sources.

8.2 Scientific Impact

When it comes to research, *Arquivo de Sentimentos* can help correlate sudden changes in sentiment to specific events. In addition, it also allows the possibility of exposing bias of news sources in relation to certain entities.

The possibility of extracting the results also allows researchers to use those same results by integrating them with other services and / or research work.

The data presented can be exported in three formats: *pdf*, *csv* and *png* - formats easily distributed and useful for presentations and for post-processing.

8.3 Additional Comments

During the development phase of the project, the team always aimed to guarantee two main requirements regarding the user experience: ease and speed.

The ease of use of the platform was guaranteed through the simplistic design of the platform and intuitive functionality thanks to the use of tooltips and examples that can be selected by the user in order to simulate the main use case of the application.

Regarding speed, the most time consuming component is the *sentiment analysis*. It is for that reason why the project heavily relies on its cache system to serve requests. The analysis only runs once per entity, thus the first query will be the slowest.

9 Conclusions

The proposed goals of the project were achieved. Not only the initial proposed features were completed with success, but also some new ones were introduced such as the possibility to visualise news utilized on the analysis.

The platform successfully bridges sentiment analysis with popular news publications in order to see how opinions shift over time and detect inherent biases. The uniqueness of the project, its utility and the technologies' maturity enabled the team to develop a robust and ready to use product, available at. arquivodesentimentos.pt.

It is important to note the key role that both APIs (*Arquivo.pt API* and *Google Natural Language API*) played in this platform. These kind of services are extremely useful, enabling developers to build complex tools in a small time frame, almost like piecing a puzzle together.

Having said this, *Arquivo de Sentimentos* could benefit from having, in the future, an API service for interested developers and researchers.

Finally, it also became evident the importance of correctly annotating web content. Otherwise, it would not be possible to accurately parse the news' content.

References

[1] Antony Samuels e John Mcgonical. News sentiment analysis. https://arxiv.org/pdf/2007.02238.pdf. Accessed in march, 2021.

[2] Aylien. Api overview - aylien. https://aylien.com/product/news-api. Accessed in march, 2021.

[3] Arquivo.pt team. Arquivo.pt api documentation. www.github.com/arquivo/pwa-technologies/wiki/Arquivo.pt-API. Accessed in may, 2021.

[4] Google. Google natural language api. https://cloud.google.com/natural-language. Accessed in may, 2021.

[5] Heroku. https://heroku.com. Accessed in may, 2021.

[6] Newspaper3k. https://newspaper.readthedocs.io/. Accessed in may, 2021.

[7] Beautiful soup. https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Accessed in may, 2021.

[8] Celery. https://docs.celeryproject.org/en/stable/getting-started/introduction.html. Accessed in may, 2021.

[9] Heroku google application credentials buildpack. https://github.com/gerywahyunugraha/heroku-google-application-credentials-buildpack. Accessed in april, 2021.

10 Annexes

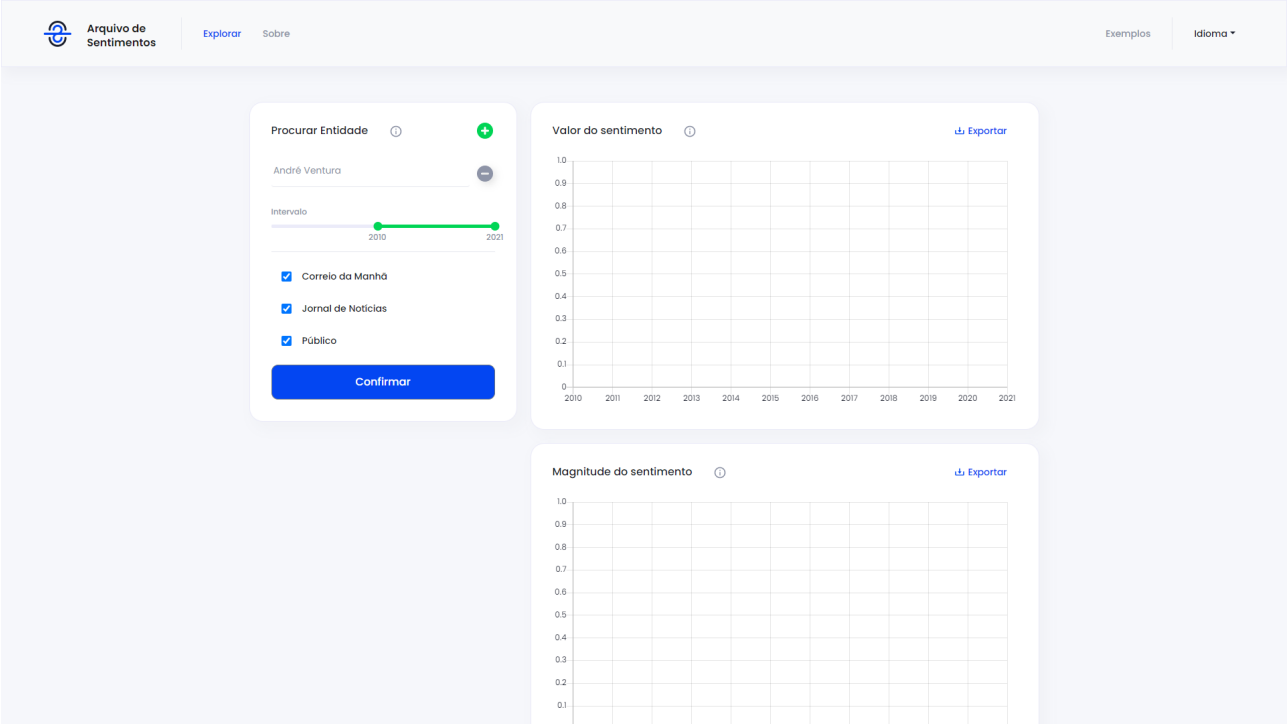

Figure 3: *Arquivo de Sentimentos - before a query*

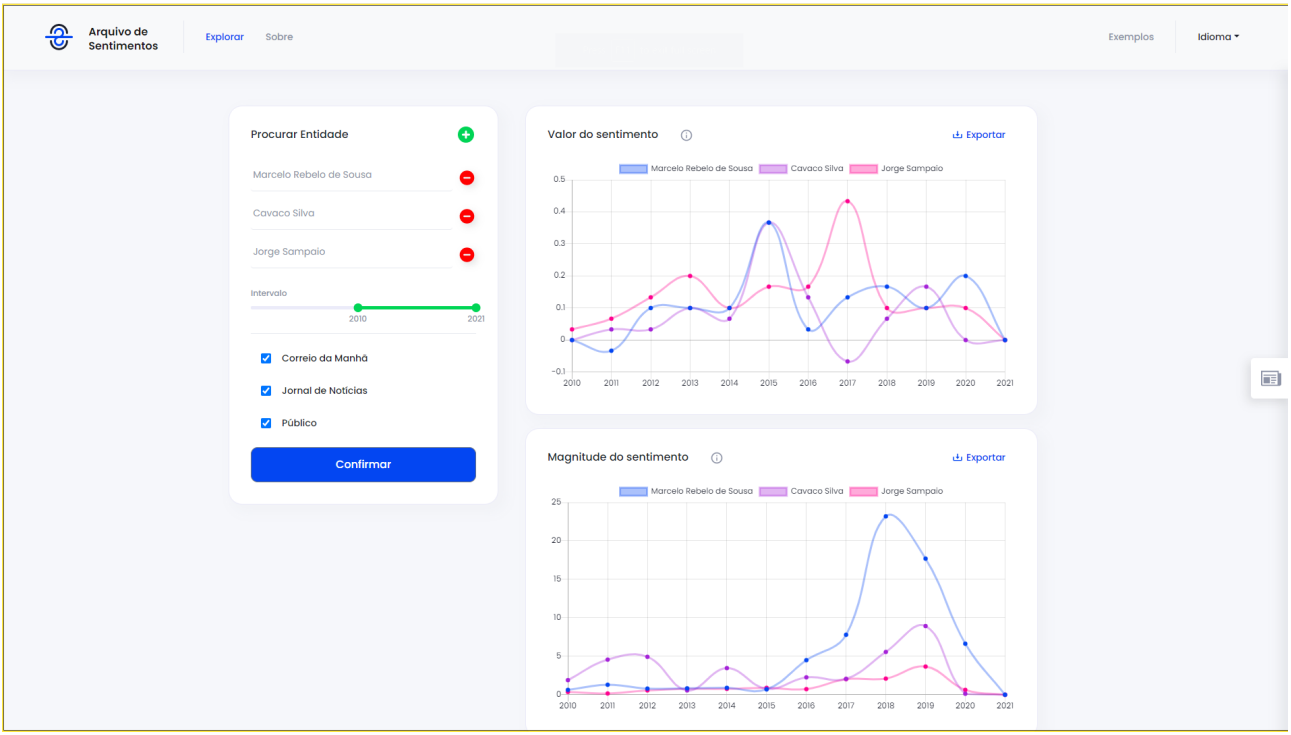

Figure 4: *Arquivo de Sentimentos - after a query*

Figure 5: *Arquivo de Sentimentos - news analysed*

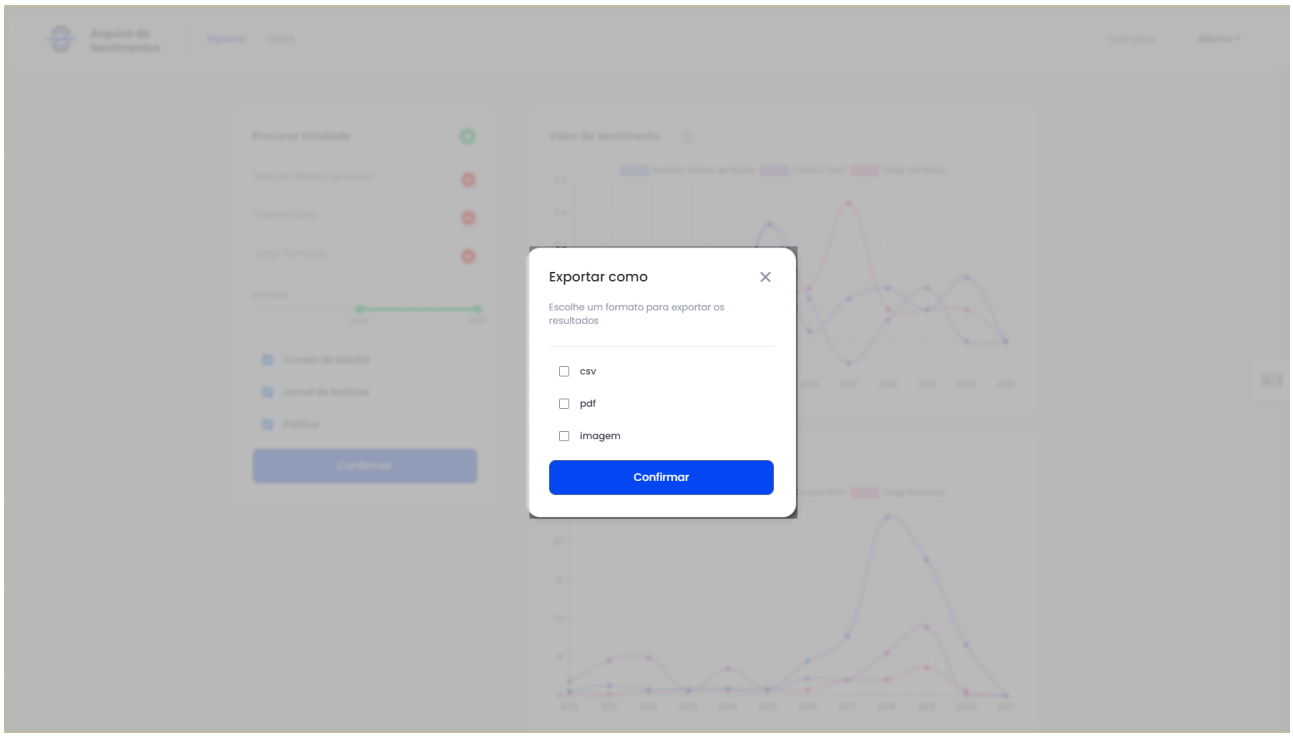

Figure 6: *Arquivo de Sentimentos - exporting the results*

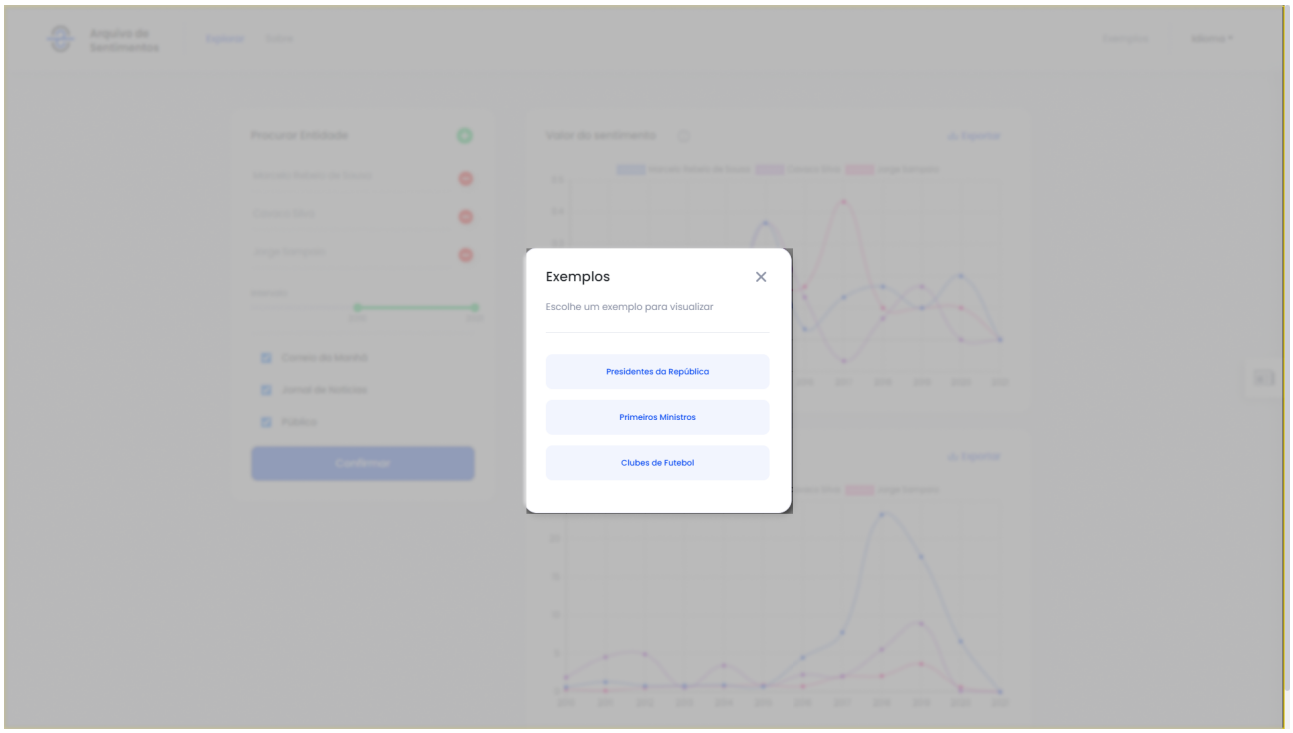

Figure 7: *Arquivo de Sentimentos - examples*

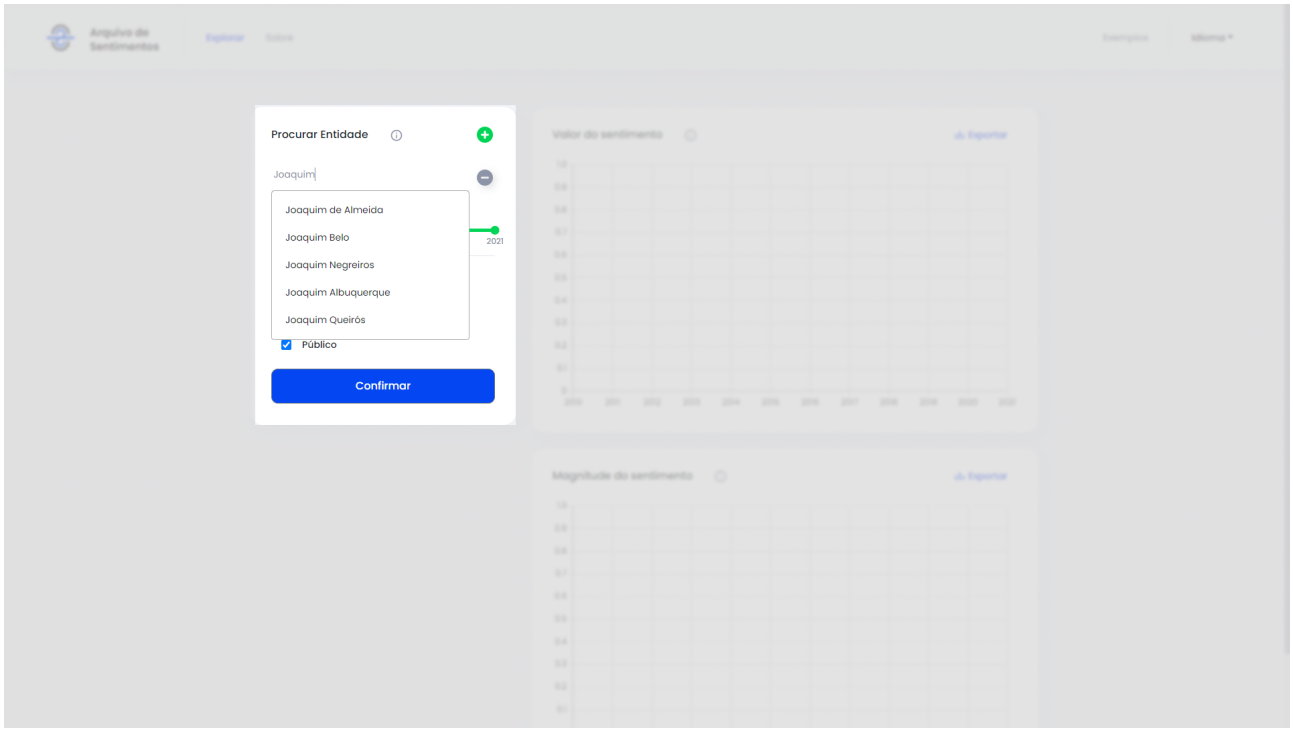

Figure 8: *Arquivo de Sentimentos - auto complete*

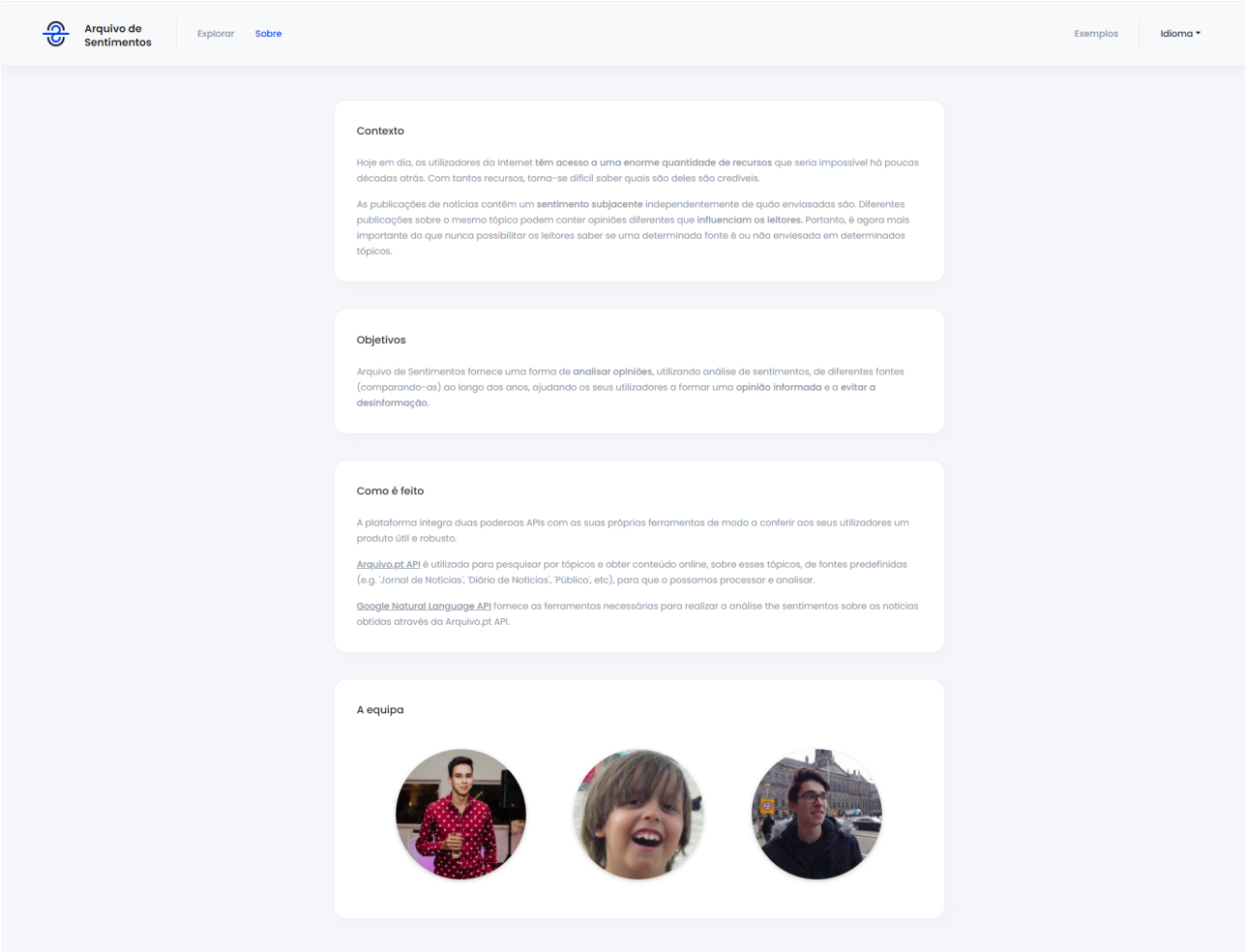

Figure 9: *Arquivo de Sentimentos - about page*