Information Search System for Versioned Portuguese News Articles about Technology

João Romão
Viseu, Portugal
up201806779@edu.fe.up.pt

Rafael Cristino
Cucujães, Portugal
up201806680@edu.fe.up.pt

Xavier Pisco
Moita do Boi, Portugal
up201806134@edu.fe.up.pt

Figure 1: Arquivo.pt

ABSTRACT

In this document we present the 1st milestone of our information search system project, which revolves around collecting the information that we will use for the development and understanding it. Our data is obtained from the portuguese web archive, more specifically from technology news articles stored by the the Arquivo.pt service. For this milestone, our goal is to collect a dataset by using this service's publicly available APIs to retrieve data and to employ data pre-processing, cleaning and refinement techniques.

After reading this you will understand how we approached this task and why did we chose to go about it in this specific way and not in any other way. We describe the data processsing and retrieval pipeline we used and the way we stored data. Finally, to provide more insight, we characterize the dataset that was collected.

KEYWORDS

websites, news, articles, technology, information, search

1 INTRODUCTION

Arquivo.pt is a service that allows anyone to visit a past version of any webpage or to visit web pages that are no longer active. This service was created by *Fundação para a Ciência e Tecnologia (FCT)* and it does so by crawling the Portuguese web and storing information about all of the webpages that it finds. This has been going on since 1996, therefore it contains a huge amount of data that can be fetched and used. All of this data is publicly available by the means of multiple API's that are described in "arquivo.pt/api".

For this project, we are going to develop an information search system centered around a dataset that is gathered from Arquivo.pt. For collecting this dataset, because we are working in an academic setting and the service's data is very extensive, we decided to use a smaller portion of what's available by defining a small scope for the information that will be retrieved. Of all the available website's archives, our work will focus on Portuguese technological news articles indexed by Arquivo.pt in 2021, from the 1st of January to the 1st of November, and that were published by 3 Portuguese news media companies: *Notícias ao Minuto*, *Jornal de Negócios* and *Exame Informática*.

2 DATA PROCESSING AND RETRIEVAL PIPELINE

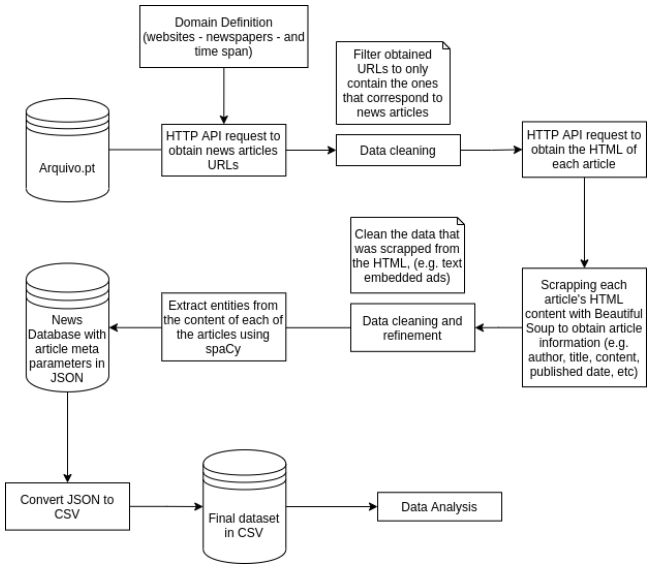

Figure 2: Pipeline diagram
Describes the data collection process.

2.1 Domain Definition

In order to get the data we need from arquivo.pt, first, we need to do API requests to get the websites Arquivo.pt has stored and select only the ones we want.

We used the CDX server API, which returns an entry for each of the indexation that exist for the query we give it. We chose to only get tech news from the three different websites already mentioned, *Noticías ao Minuto*, *Jornal de Negócios* e *Exame Informática*, and restricted them to the ones that were indexed in 2021, from the 1st of January to the 1st of November. This part of the pipeline was executed in a python script named "pull-links.py".

2.2 Filter Domain Data

After this initial gather of data, some of it was not useful for our project. Some of the links we had were from websites that did not contain news but had a similar url to the ones that did. We decided to remove those links before downloading the websites because that would reduce the time needed to proccess the next step of our pipeline. For this we created and used a python program that we called "remove-non-news.py".

2.3 Getting the HTML

After having the links to the websites that we want to get our data from, we need to download the HTML from those websites and store it together with all the other available information about that website.

In this part, we used the NoFrame API and saved the HTML from all the websites we had previously filtered, in order to be used to get the information needed for our project. Our script "pull-html.py" is responsible for this.

2.4 Scrapping of HTML

The HTML we got was too big and had too much code that we just didn't need for our final goal. We decided that we needed to scrap that HTML and store only the important parts.

Thus, we created a python program called "parse-information.py" where we read the HTML using the Beautiful Soup library and we retrieved the parts that were important to us, for example, the title, author, text, etc.

2.5 Data Cleaning

Since some of the retrieved information that we got on the previous step had some irrelevant text and some incosistencies, we decided to clean those parts of the data. As an example, some of the news had ads in the middle of the text and sometimes the newlines were represented by more than one \n.

Most of this was made at the same time as the scrapping in order to improve the time efficiency and avoid an unnecessary extra loop on the data, except for the embedded ads, that were removed in the "clean-ads.py" script.

2.6 Extracting Entities

One thing we thought was interesting to have in our project was a list of all the entities refered in each of the articles we had. This may allow us, in the future, to get a better grasp of each article's content and possibly to define relations between multiple articles. So, for retrieving the entities from the articles' contents, we decided to use the python library spaCy in our "parse-entities.py" program to do that.

2.7 Convert JSON to CSV

In order to easily analyze and characterize the data we thought we should change its format, thus, we decided we converted it to CSV format. As a result we created 5 separate CSV files whose contents are better explained in the Data Storage section. In the makefile this is represented by the "json_to_csv.py" program.

3 CONCEPTUAL DATA MODEL AND DATA STORAGE

In order to more easily manipulate the data we decided to change its format from json to csv. For that, basing ourselves in the conceptual data model (figure 3), we splitted it into 5 different csv files:

- "domain.csv": maps to the "Website" in the conceptual data model.
- "urlkeys.csv": maps to the "News" in the conceptual data model.
- "news.csv": maps to the "Version" in the conceptual data model.
- "entities.csv": maps to the "Entity" in the conceptual data model.
- "news_entities.csv": maps to the many-to-many relation between "Version" and "Entity".

For simplicity's sake, we decided to keep the author's name in the "news.csv" file, even though that slightly increases the redundancy in our dataset.

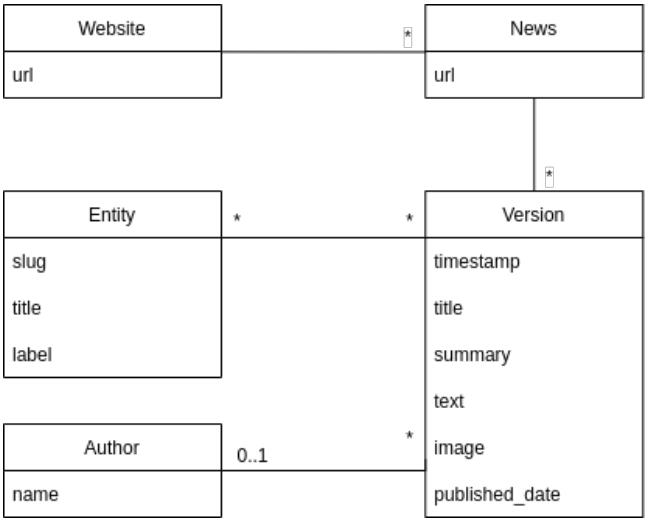

Figure 3: Conceptual Data Model
Describes the relations between our data.

On the website table, we have the url corresponding to the news companies that we defined in our problem domain, being identified by "noticiasaominuto", "jornaldenegocios" and "exameinformatica".

On the news table, we have the urlkey for each of the news, this key is a specific name given by arquivo.pt that is unique to each of the website it has stored and based on the url that was scraped. This table is connected with the website table since each of the urlkeys belongs to one of the domains.

Each of the news has one or more versions that correspond to each of the indexations made by Arquivo.pt for that particular article. Those versions have a timestamp, that stores the date of indexation, and the article's metadata and contents, as described in the diagram.

For each version of the news article there is a specific author, and each author can correspond to more than one article versions.

At last, the versions have entities in them that are stored in a separate table, the entities table, and have a title, a slug, which is an identifier based on its title, and a label. The label can be one of PER (person), ORG (organization), LOC (localization), and MISC (miscellaneous).

4 DATA CHARACTERIZATION

4.1 Amount of news

In order to better understand the amount of data we are dealing with and to know from which domain the majority of news come from, a plot mapping each domain to the number of distinct news indexed in 2021 was built.

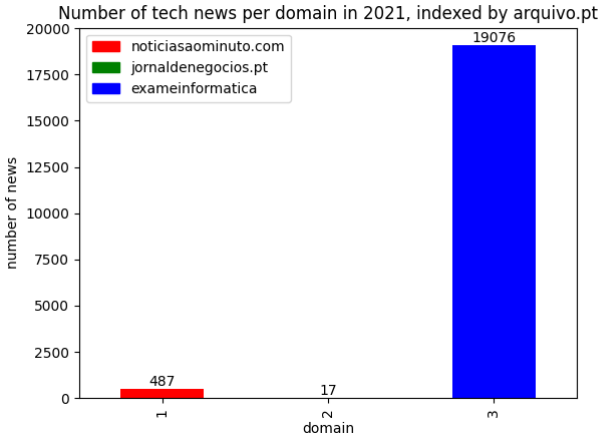

Figure 4: Amount of unique articles indexed per domain

Since each article can be indexed at Arquivo.pt more than once, the total amount of indexed articles was also plotted to each domain.

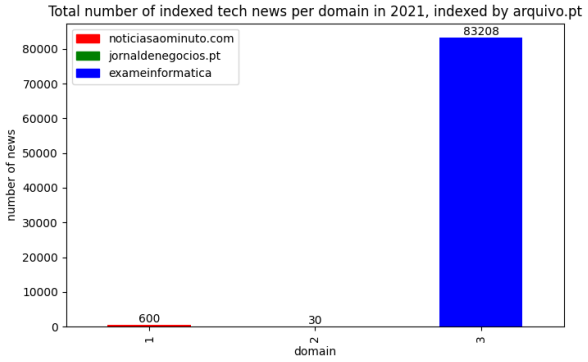

Figure 5: Total number of indexed articles, including multiple versions, per domain

4.2 Indexation count

As it was shown in the previous subsection, each article can be indexed more than once. This way, it would be interesting to know how many times each article has been indexed at the Arquivo.pt database. To visualize this aspect, a plot mapping the number of entries to the number of articles grouped by the number of times they were indexed was built (for example, in figure 6, 1665 articles were indexed only once, while 9996 articles were indexed twice).

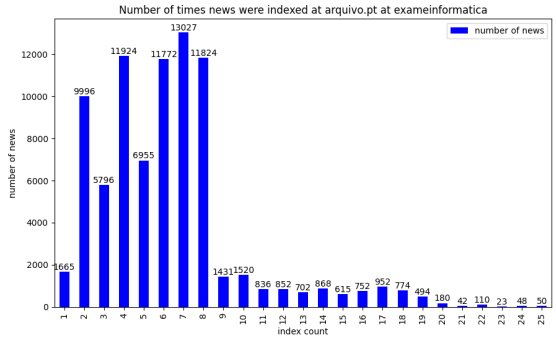

Figure 6: Count of the number of news grouped by the number of times they were indexed

4.3 Average article length

Another characterization we found interesting was to compare the length of articles from different sources. To visualize this aspect we plotted each domain to the average length of its articles taking a character as measurement unit.

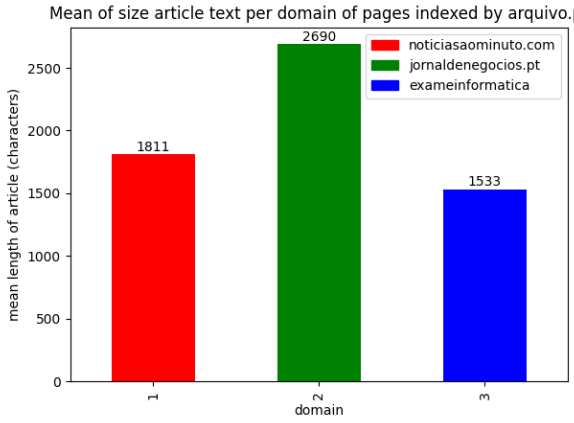

Figure 7: Average article length (characters)

4.4 Group news by publication year

Even though only news indexed at Arquivo.pt in 2021 were taken into consideration, the article's publish date may differ a lot from the date it was indexed. To understand how the articles spreaded over time, the articles were grouped by the corresponding publication year.

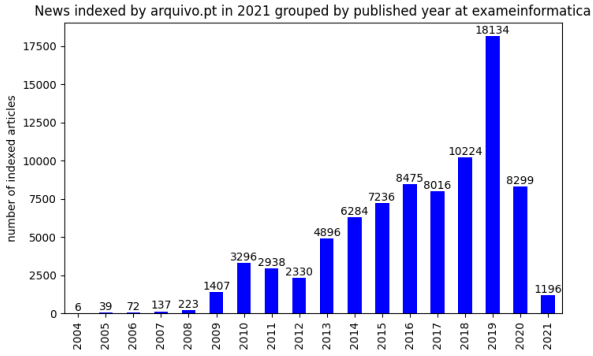

Figure 8: Count of news grouped by publishing year

4.5 Entities count

Because we have parsed all of the entities in the articles' contents, we can now get a better grasp of the topics that are most mentioned in the articles we collected. We did so by calculated the entities that are mentioned in the most articles (for example, in figure 9, "Google" was mentioned in 10756 indexed articles).

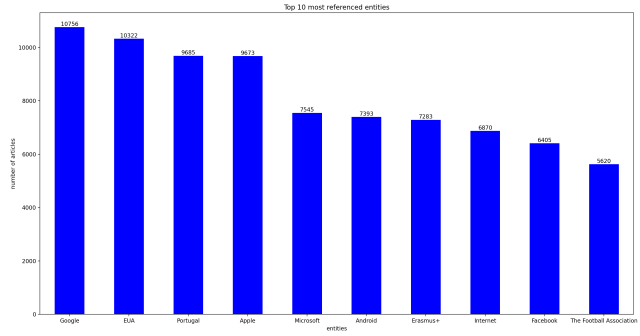

Figure 9: Top 10 referenced entities

5 1ST MILESTONE CONCLUSION

During the development of the 1st milestone of our project we got to explore Arquivo.pt and its APIs, as well as experiment with techniques such as HTML scrapping and entity extraction from natural language text.

We acquired new knowledge on how to extract useful information from webpages and on the actions that must be taken in order to clean and refine it into a format that we can use. We also got to analytically characterize this information by manipulating the resulting dataset, which left us with a better understanding of what is possible to be extracted from it.

We now have a refined dataset that will be the foundation for the next stages of our information search system and that provides us with multiple opportunities for meaningful knowledge extraction.