Leopold-Franzens Universität Innsbruck
Fakultät für Mathematik, Informatik und Physik

Institut für Informatik

Semantic Technology Institute Innsbruck

Bachelorarbeit

zur Erreichung des akademischen Grades

Bachelor of Science

Analyzing User Search Behaviour in Temporal Web Repositories through Search Query Log Analysis

von

Flavie Coralie Capucine Gallois
(Matr.-Nr.: 11927931)

Submission Date: June 4, 2023
Supervisor: Prof. Dr. Adam Jatowt

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt durch meine eigenhändige Unterschrift, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Alle Stellen, die wörtlich oder inhaltlich den angegebenen Quellen entnommen wurden, sind als solche kenntlich gemacht.
Ich erkläre mich mit der Archivierung der vorliegenden Bachelorarbeit einverstanden.

Ort und Datum: _____

Unterschrift: _____

Contents

List of Figures

Abstract

This paper presents an analysis of user search behavior in a temporal web repository, based on a study of search query logs collected over a period of 3 months from June to September 2021. Through an examination of various metrics such as query length, query type, and query frequency, the purpose of the research is to gain insights into how users search for information in temporal web repositories and to identify patterns in their search behavior over time. The analysis reveals that users of the temporal web repository tend to use longer and more complex queries compared to previous studies, suggesting a higher level of precision and awareness of search techniques. We also identify patterns of multiple queries per session, indicating a need for refining search queries or exploring multiple avenues of inquiry. Overall, the study highlights the importance of understanding user behavior in web archives and provides valuable insights for researchers and developers seeking to improve the search functionality of digital archives and enhance the user experience for accessing them.

1 Introduction

With the vast amount of information available on the web, effective search tools are essential for users to find the content they need. Understanding how users search and navigate through this information is critical for improving the performance of search engines and other information retrieval systems. Search query logs, which record the queries and interactions of users with a search engine, web archives or other web repository, provide a valuable source of data for analyzing user search behavior.

The concept of a web archive, also called Internet archive, is relatively young as it exists for around only 30 years as of 2023. It has been prevalent with the democratisation of computers and so the rise of the use of Internet in the mid to late 1990s. One of the first larged scaled web archiving project was the Internet Archive, an american digital library in 1996. They stated their mission of "universal access to all knowledge" and made their archived content available to the general public in 2001 with the Wayback Machine [19] with an ascending number of saved web pages over the years.

Internet archives record various types of web content including web pages (HTML, CSS, JavaScript) but also media such as images and video. They also happen to archive diverse metadata such as access time or MIME type in order to establish authenticity and provenance of the collection.

The focus of this study is on the Portuguese web archive Arquivo.pt [7] that has been preserving web pages from 1996 on. Operating since January 2008, yet the idea came

up as soon as in 2001 with a scientific project ran by a research group at the Faculty of Sciences of the University of Lisbon [8]. Arquivo.pt provides a comprehensive crawl of the Portuguese Web, when it is made only partially by other archives. The archive also allows a search by full text or by address (URL) which can facilitate the search for users.

Such a project, is motivated by the fact that after 1 year only 20% of a set of addresses remain valid [13] and with the amount of information rising over the years, more and more information cease to be available to the public and is irrevocably lost.

One aspect of user search behavior that has received increasing attention in recent years is the temporal aspect. The behavior of users can change over time, and temporal trends in search behavior can reveal important insights into how users interact with web repositories. This research paper aims to analyze user search behavior in temporal web repositories through search query log analysis. Specifically, the paper will examine the temporal patterns in user search behavior.

While most of people use search engines almost daily, it is natural that many articles and studies of web user behaviour on search engines exist. However, as web archives are less used, but also serve a different purpose then search engines, the definition of a user's behavior on a web archive remains unclear, leaving only speculation open.

This paper will begin with a general overview of information retrieval methods and search behavior analysis techniques. We will then shift the focus to the specific web archive under study, arquivo.pt, and provide an overview of the dataset being analyzed in detail.

Next, contents of the generated search query logs will be analyzed, with a focus on understanding user search behavior in arquivo.pt. By examining the query patterns and behaviors of users over time, trends and patterns that can provide insights into user search behavior will be identified.

Finally, the paper will conclude with a discussion of the limitations of this study. Despite the value of search query logs in analyzing user behavior, it is important to acknowledge the inherent limitations of this data and to consider its potential biases.

2 Literature Review

Information retrieval is a field of study that focuses on the design and evaluation of systems for retrieving relevant information from large collections of data, such as text documents, images, and videos. The goal of information retrieval is to help users find the information they need in the most efficient and effective way possible. This can involve a range of techniques, including keyword searching, natural language processing, and machine learning.

Search behavior analysis is the study of how users search for information, including their queries, interactions with search engines. By analyzing search behavior, researchers and practitioners can gain insights into how to improve information retrieval systems and make them more effective for users. Search behavior analysis can involve various methods, such as log analysis, which is the subject of this paper, surveys, eye tracking, and user studies.

One of the key challenges in information retrieval and search behavior analysis is the semantic gap, which refers to the mismatch between the user's information need and the representation of the data in the system.

Overall, information retrieval and search behavior analysis are closely linked, as they both aim to help users find the information they need in the most efficient and effective way possible. By combining insights from both fields, researchers and practitioners can develop more advanced and user-friendly information retrieval systems.

2.1 Information retrieval techniques

Information retrieval (IR) techniques are methods and algorithms used to retrieve relevant information from large collections of unstructured or semi-structured data. We can list several models [15] :

- **Boolean Model** : based on Boolean algebra, documents and queries are represented as sets of keywords or terms. The search query is constructed using Boolean operators (AND, OR, NOT) to refine search results.

  Example of a query of a user searching articles about health and fitness : "(Health OR Fitness) NOT Weightlifting"

  This query will retrieve all articles that contain either the terms "health" or "fitness", or both, but exclude any articles that also contain the term "weightlifting".

  This method may be used by users searching for elements in an internet archive.

- **Vector Space Model** : Represents queries as vectors in a high-dimensional space, where each dimension corresponds to a term or keyword. The relevance of a document to a query is calculated using the cosine similarity between the document vector and the query vector.

  Example of three sentences :

  1. "The quick brown fox jumped over the lazy dog"
  2. "The brown fox is quick and agile"
  3. "The lazy dog is not very active"

  The vectors for each sentence may be as follows :

  1. [1, 1, 1, 1, 1, 0, 0] This vector has seven dimensions, corresponding to the terms "the", "quick", "brown", "fox", "jumped", "over", and "lazy".

The weight for "brown" and "fox" is 1, since these terms appear in the sentence, while the weight for all other terms is 0.

2. [1, 0, 1, 1, 0, 0, 0]
3. [1, 0, 0, 0, 0, 1, 1]

To determine which documents are most relevant to the query, we calculate the cosine similarity between the query vector and each sentence vector with the following formula between two vector A and B :

$$(A \cdot B)/(||A|| * ||B||)$$

The document with the highest cosine similarity is the most relevant.

In this example, the cosine similarity between the query vector and the sentence vectors are:

1. 0.67
2. 0.67
3. 0

Which means that sentence 1 and 2 are both relevant to the query while sentence 3 is not. While Vector Space Model is not used by users in this context, it is a technique later explored in this paper to try to find most similar queries and try to categorize them.

- **Probabilistic Model** : It models the probability of a document being relevant to a query using a probabilistic function that takes into account the frequency of query terms in the document and in the collection as a whole.

  Taking the collection of sentences used previously :

  1. "The quick brown fox jumped over the lazy dog"
  2. "The brown fox is quick and agile"

3. "The lazy dog is not very active"

Now suppose a user wants to find documents containing the query "lazy dog". The probabilistic model assigns a probability of relevance to each document based on the frequency of the query terms in the document.

Assuming that P(relevant) = 0.33 we can calculate the probability of relevance for each document as follows:

1. P(Sentence 1 — relevant) = $(\frac{1}{6} * \frac{1}{5}) / (\frac{1}{3}) = 0.10$
2. P(Sentence 1 — non-relevant) = $(\frac{1}{6} * \frac{2}{5}) / (\frac{2}{3}) = 0.33$
3. P(Sentence 2 — relevant) = $(0 * \frac{1}{5}) / (\frac{1}{3}) = 0$
4. P(Sentence 2 — non-relevant) = $(\frac{1}{6} * \frac{1}{5}) / (\frac{2}{3}) = 0.04$
5. P(Sentence 3 — relevant) = $(0 * \frac{1}{5}) / (\frac{1}{3}) = 0$
6. P(Sentence 3 — non-relevant) = $(0 * \frac{2}{5}) / (\frac{2}{3}) = 0$

Therefore, the most relevant document is Sentence 1 with a probability of relevance of 0.10, and the other documents have probabilities of 0 or close to 0.

- **Inference Network Model** : Probalistic approach to IR that represents both the query and the document as probability distributions. It uses Bayesian inference to estimate the probability that a given document is relevant to the query, based on their term representations. The result is a set of scores that reflect the relevance of each document to the query.

  Reusing the set of sentences used previously :

  1. "The quick brown fox jumped over the lazy dog"
  2. "The brown fox is quick and agile"
  3. "The lazy dog is not very active"

Let's assume that a user enters the query "quick brown fox". The query can be represented and each sentence above as sets of terms:

Query: {quick, brown, fox}

S1: {quick, brown, jumped, over, lazy, dog}
S2: {brown, fox, quick, agile}
S3: {lazy, dog, not, very, active}

Using the Bayesian inference, we start with the probability of relevance which we can estimate from the frequency of relevant documents in the collection. Let's estimate the probability to be 0.33 as there is one relevant sentence out of three.

Then we compute the likelihood of observing each document given that it is relevant to the query, and the likelihood of observing each document given the query itself. For example, the likelihood of observing {quick, brown, jumped, over, lazy, dog} given that it is relevant to the query would be proportional to the overlap between {quick, brown, jumped, over, lazy, dog} and the query, which is quick, brown, fox. We can compute this overlap by counting the number of times each term appears in {quick, brown, jumped, over, lazy, dog} and the query, and taking the minimum of these counts:

$$P(S1|relevant, query) = min(count(S1, quick), count(query, quick))/len(S1)$$
$$*min(count(S1, brown), count(query, brown))/len(S1)$$
$$*min(count(S1, fox), count(query, fox))/len(S1)$$

We can compute the likelihood of observing each document given the query itself in a similar way, by computing the overlap between the document and the query and taking the minimum of the counts:

$$P(S1|query) = min(count(S1, quick), count(query, quick))/len(S1)$$

$$*min(count(S1, brown), count(query, brown))/len(S1)$$

$$*min(count(S1, fox), count(query, fox))/len(S1)$$

Once this has been done for all three sentences we can use Bayes' theorem to compute the posterior probability of relevance for each sentence, and then rank them by relevance. In this case, since S1 contains all three query terms, it would likely be ranked highest, followed by S2 (which contains two of the three query terms), and then S3 (which contains none of the query terms).

2.2 Searching techniques

Searching techniques are essential for finding and retrieving relevant information from large collections of data. Following we will explore the 3 main searching techniques :

- Linear search algorithm : Basic searching technique used to find a specific value in a list or array of values.

  The algorithm starts at the beginning of the list and compares each element to the target value. If the value is found, the algorithm returns the index or position of that value in the list. If the value is not found, the algorithm continues to search through the list until the end is reached.

  With a time complexity of O(n), where n is the number of elements in the list, the time taken to search through the list increases linearly with the size of the list.

  While the linear search algorithm is simple and easy to implement, it can be inefficient for large lists or arrays.

  Here an example in Python

```
# Python3 code to linearly search x in arr[].
# If x is present then return its location,
# otherwise return -1

def search(arr, x):
    for i in range(len(arr):
        if (arr[i] == x):
            return i
    return -1
```

- Binary search algorithm : Technique used to find a specific value in a sorted list or array of values.

  The algorithm works by dividing the list into two halves and checking if the middle element is equal to the target value. If the middle element is not equal to the target value, the algorithm checks if the target value is greater than or less than the middle element, and discards the half of the list that does not contain the target value. The process is repeated until the target value is found, or the list is exhausted.

  With a time complexity of O(log n), where n is the number of elements in the list, the time taken to search through the list increases logarithmically with the size of the list.

  The binary search algorithm is more efficient than linear search for larger data sets, but it requires the list to be sorted. If the list is not sorted, it must be sorted first using a sorting algorithm, which adds an additional time cost.

  Here an example in Python

```
# Python 3 program for recursive binary search.
# Returns index of x in arr if present, else -1
```

```
3
4     def binary_search(arr, low, high, x):
5
6     # Check base case
7         if high >= low:
8             mid = (high + low) // 2
9
10            # If element is present at the middle itself
11            if arr[mid] == x:
12                return mid
13
14            # If element is smaller than mid, then it can only
15            # be present in left subarray
16            elif arr[mid] > x:
17                return binary_search(arr, low, mid - 1, x)
18
19            # Else the element can only be present in right subarray
20            else:
21                return binary_search(arr, mid + 1, high, x)
22
23        else:
24            # Element is not present in the array
25            return -1
```

- Hash tables : Searching technique and data structure that allow for efficient search and retrieval of values based on keys.

  A hash table uses a hash function to map each key to a unique index in an array. The value corresponding to the key is then stored in the array at the index determined by the hash function.

  When searching for a value based on a key, the hash function is used to de-

termine the index of the array where the value is stored. If the value is not found at that index, a collision has occurred, and the hash table uses a collision resolution strategy, such as chaining or linear probing, to find the correct index.

The time complexity of searching for a value in a hash table is O(1), or constant time, on average, making it a highly efficient searching technique for large data sets. However, hash tables require careful selection of the hash function and may incur additional memory costs for storing the array and handling collisions.

Here one possible implementation in Python

```
# Function to display hashtable
def display_hash(hashTable):

    for i in range(len(hashTable)):
        print(i, end = " ")
        for j in hashTable[i]:
            print("-->", end = " ")
            print(j, end = " ")
        print()

# Creating Hashtable as
# a nested list.
HashTable = [[] for _ in range(10)]

# Hashing Function to return
# key for every value.
def Hashing(keyvalue):
    return keyvalue % len(HashTable)

# Insert Function to add
```

```
21      # values to the hash table
22      def insert(Hashtable, keyvalue, value):
23
24          hash_key = Hashing(keyvalue)
25          Hashtable[hash_key].append(value)
```

2.3 Query Log analysis techniques

Temporal web repositories refer to collections of web pages or documents that have been archived over time. These archives allow researchers to study changes in the web and user behavior over time, and provide a valuable resource for studying the evolution of the web and its impact on society.

Temporal web repositories have been used in a wide range of research areas, including information retrieval, web mining, and digital humanities. Understanding temporal changes in the web and user behavior is essential for improving search engine performance and developing new search algorithms that can adapt to changing user needs and behaviors.

They are particularly useful for studying long-term trends in search behavior and web content, as they capture changes that may not be apparent in real-time web data. Additionally, temporal web repositories can be used to validate search algorithms and models by comparing their performance over time.

Overall, temporal web repositories play a significant role in search behavior analysis by providing researchers with a rich source of data that are query logs to study how users search for and interact with information on the web.

Search query log analysis techniques refer to the process of analyzing logs of search queries entered by users into search engines. These logs contain valuable data that can be used to improve search engine performance and understand user behavior.

Some common search query log analysis techniques include:

- Query suggestion: Suggesting alternative or related search terms to users based on their query history and behavior.
- Query classification: Classifying queries into categories based on their intent, such as navigational, informational, or transactional.
- Query expansion: Expanding queries to include additional related terms, which can improve the relevance of search results.
- User clustering: Clustering users into groups based on their search behavior, which can be used to personalize search results and improve ad targeting.
- Session identification: Identifying sessions of search activity, which can help to understand user behavior and search patterns.

The applications of search query log analysis techniques are numerous. For example, search query log analysis can be used to:

- Improve the relevance of search results by understanding user intent and behavior.
- Inform search engine marketing strategies by identifying popular search terms and patterns.
- Improve the design and functionality of search engines by identifying areas for improvement based on user behavior.
- Personalize search results and ad targeting based on user behavior and preferences.
- Develop new search algorithms and models based on analysis of user behavior and search patterns.

Overall, search query log analysis techniques play a critical role in improving search engine performance and understanding user behavior on the web.

The most used technique in this paper is the query classification method in order to help to understand if the needs of users has been met, what kind of category the users are searching and how are they achieving their goal comparing to a few years ago.

3 Search Environment

3.1 Arquivo.pt

Arquivo.pt is a Portuguese Internet Archive. That is to say, the goal of the team behind this project is to collect parts of the World Wide Web, in order to preserve any information for a future use for researches, historians but also the public. The archive is active since January 2008, in several languages, and provides a public search service, nevertheless files accessible on the archive are archived from the web since 1996. Recurrently it collects and stores information from all over the web, after that the data is processed to make it searchable with a search engine-like approach.

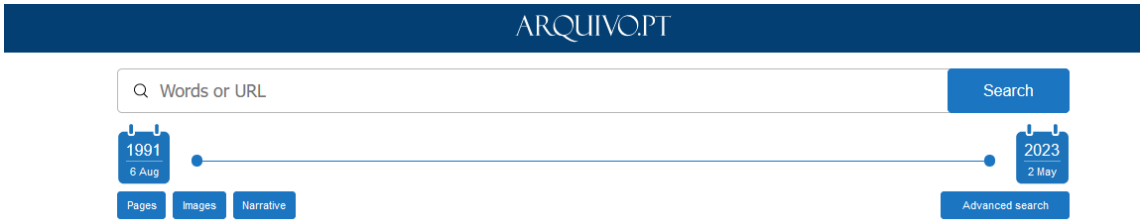

Figure 3.1: The welcome page of arquivo.pt (en)

Arquivo.pt works as follows, the user enters its query in the search bar, which can either be full-text query or an URL. The search results can be narrowed down by selecting a timeframe using the "from" and "to" date options. Additionally, there is an advanced search feature which allows users to filter their results by format (such as .jpg or .pdf) and choose the maximum number of results displayed per page.

Similar to classic search engines, Arquivo.pt offers users the option to apply a "filter"

to their full-text search, such as filtering explicit results by including ”safe:off” or ”safe:on” in the query.

If the query is a full-text query, the result will be as shown in Figure 3.2 , the archive gives back a list of websites, much like a classic web search engine, that existed between the chosen timeframe.

If it was a URL search, it will be like on Figure 3.3. As the user enters a URL, the website gives back a collapsed menu of years containing months themselves containing a specific date the searched website has been archived on.

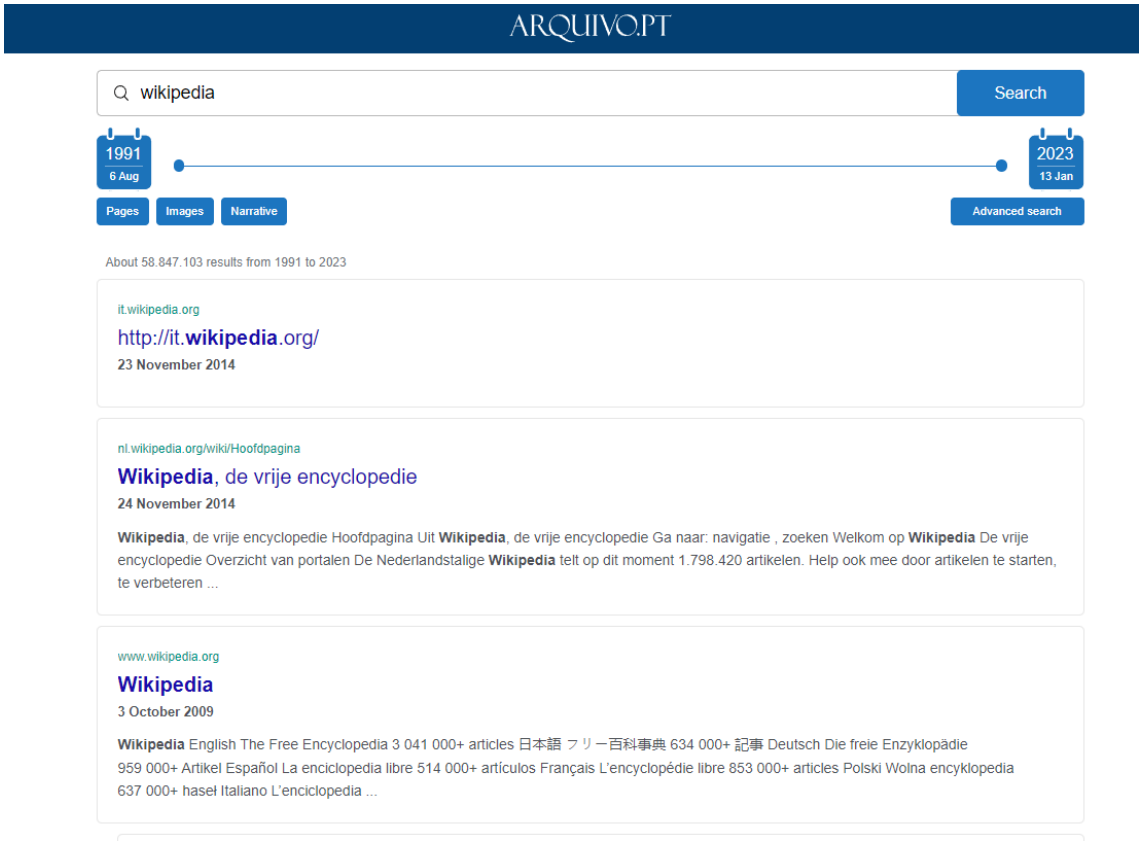

Figure 3.2: A full text search

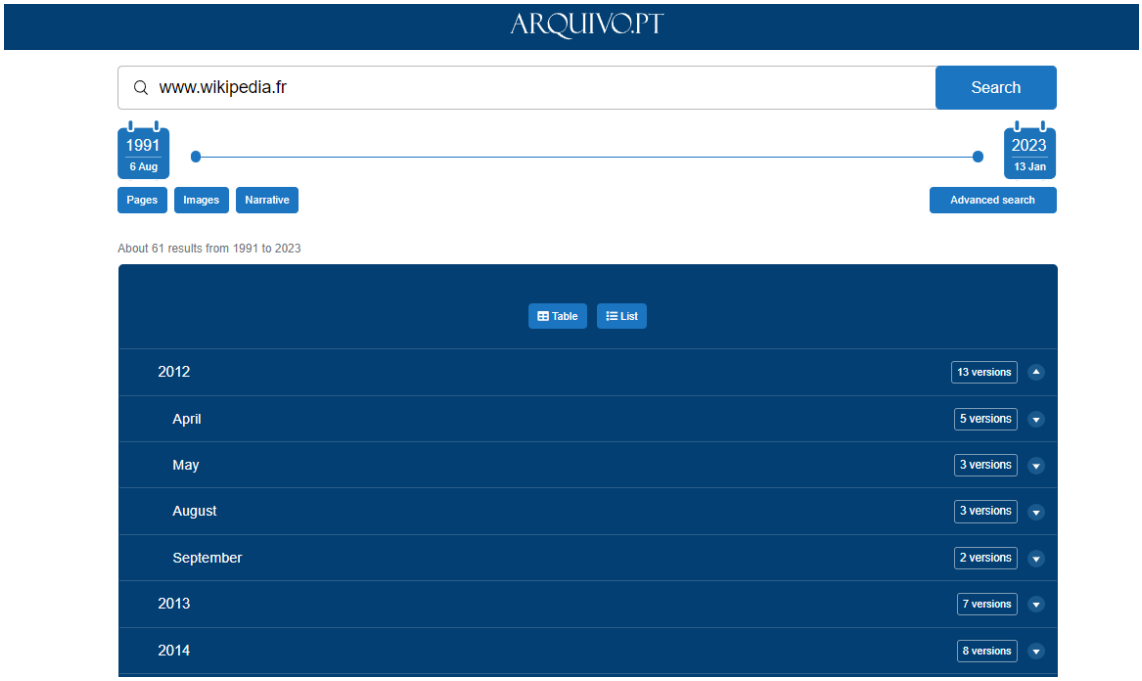

Figure 3.3: An URL search

3.2 Logs dataset

The arquivo.pt team generously provided query logs from the 1st of June 2021 until 1st of September 2021, so 3 months worth of user searches on the web archive. The data set, transferred via a .csv file, contains over 35000 logs with 36 different categories such as the IP address, a timestamp, the query but also the approximate location of the user or the device they were on.

Following is an example of an entry log :

IP_ADDRESS	REQUEST	USER_AGENT	TRACKINGID
86.20.244.236	GET /textsearch?q=etsy %20body%20chain &from=19960101000000 &to=20210601235959 &offset=0 &maxItems=10 &siteSearch=&type= &collection= &trackingId=1d6c38c4f2b 9311f726e_be 56f1c752ae7912e4b4 HTTP/1.1	Mozilla/5.0 (iPhone; CPU iPhone OS 14_6 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.1 Mobile/15E148 Safari/604.1	1d6c38c4f2b9311f 726e_be56f 1c752ae7912e4b4

TIMESTAMP	YEAR	MONTH	DAY	HOUR	MINUTE	TYPE_SEARCH
1622513748	2021	6	1	3	15	textsearch

QUERY	PAGE	MAXITEMS	PAGE_SEARCH_RESPONSE(ms)
etsy body chain	0	10	4040

PAGE_SEARCH_PARAMETERS	IMAGE_SEARCH_RESPONSE(ms)
search_parameters: {”offset”:0,”dedupValue”:2, ”from”:”19960101000000”, ”to”:”20210601235959”,”type”:[], ”collection”:[],”dedupField”:”site”, ”q”:”etsy body chain”, ”maxItems”:10,”siteSearch”:[]}	10

IMAGE_SEARCH_PARAMETERS	IMAGE_SEARCH_RESULTS	SESSION_ID

POSITION	COUNTRY	CITY	ISP	PROVINCE
	United Kingdom	England	AS5089 Virgin Media Limited	Middlesbrough

TIMEZONE	HOSTNAME	TYPE_DEVICE	BROWSER_FAMILY
Europe/London	cpc82587-midd20 -2-0-cust235.11- 1.cable.virginm.net	Mobile	Mobile Safari

BROWSER_VERSION	OS_FAMILY	OS_VERSION	DEVICE_FAMILY
14.01.2001	iOS	14.6	iPhone

DEVICE_BRAND	DEVICE_MODEL	ABSOLUTE_POSITION
Apple	iPhone	

As it can be seen, some categories can be redundant as, for example,"PAGE_SEARCH _PARAMETERS" share some common informations with "REQUEST" or "QUERY". Due to those overflowing informations, a data log may seem massive but it serves the interest of researchers analyzing the data since it allows some parameters to be extracted more easily.

1. **IP_ADDRESS** : Consists in a unique string of characters that identifies each computer using the Internet Protocol to communicate over a network.
2. **REQUEST** : Parameter "REQUEST" from apache.
3. **USER_AGENT** : Consists in a user agent is a computer program representing a person.
4. **TRACKINGID** : Unique value that identifies the user, the search, and the session.
5. **TIMESTAMP** : The instant when the user submitted a request (i.e., query

or click).

6. **YEAR, MONTH, DAY, HOUR, MINUTE** : Values generated through the timestamp column.
7. **TYPE_SEARCH** : indicates the type of search (image or text).
8. **QUERY** : Set of keywords submitted by the user.
9. **PAGE** : Number of page where the result has been clicked.
10. **MAXITEMS** : Value of maximum items per SERP page.
11. **PAGE_SEARCH_RESPONSE** : Time to return a response from Page Search API.
12. **PAGE_SEARCH_PARAMETERS, IMAGE_SEARCH_PARAMETERS** : Set of parameters concerning the query and user defined parameters.
13. **IMAGE_SEARCH_RESPONSE** : Time to return a response from Image Search API.
14. **PAGE_SEARCH_RESULTS** : List of results returned to the user for a given query on page search (only a field when there is a click).
15. **IMAGE_SEARCH_RESULTS** : List of results returned to the user for a given query on image search (only a field when there is a click).
16. **SESSION_ID** : Session id from jsp.
17. **POSITION** : Position clicked on the page by the user.
18. **COUNTRY** : Country based on IP Address [11].
19. **CITY** : City based on IP Address [11].
20. **ISP** : ISP based on IP Address [11].
21. **PROVINCE** : Province based on IP Address [11].
22. **TIMEZONE** : Timezone based on IP Address [11].

23. **HOSTNAME** : Hostname based on IP Address [11].
24. **TYPE_DEVICE** : Type of device based on User Agent [18].
25. **BROWSER_FAMILY** : Browser family based on User Agent [18].
26. **BROWSER_VERSION** : Browser version based on User Agent [18].
27. **OS_FAMILY** : OS family based on User Agent [18].
28. **OS_VERSION** : OS Version based on User Agent [18].
29. **DEVICE_FAMILY** : Device family based on User Agent [18].
30. **DEVICE_BRAND** : Device brand based on User Agent [18].
31. **DEVICE_MODEL** : Device model based on User Agent [18].
32. **ABSOLUTE_POSITION** : the absolute position in the SERPs.

4 Query Analysis

4.1 General Analysis

We first describe the results of a general analysis. With a dataset of 35,528 queries of 37 different languages, the library SpaCy [6], coupled with the language Python, was found useful to first analyse what kind of data was contained. Using the SpaCy's trained pipelines for models and languages [4] revealed itself to be particularly efficient and, as a result, resulted in figure 4.1.

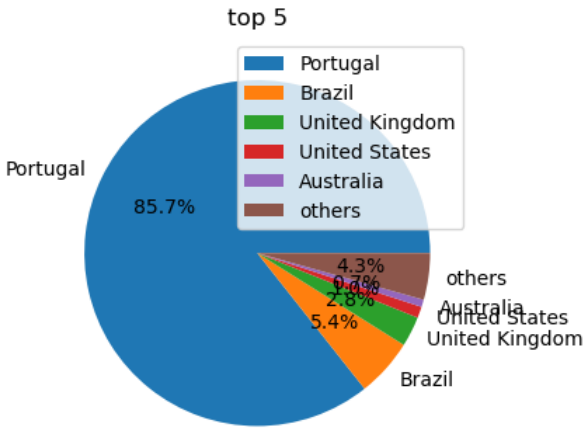

Figure 4.1: Countries the queries are coming from

The analysis reveals a significant diversity of languages in the queries. However, it is unclear what the accuracy of the model used is and so if all the predictions are correct. One might speculate whether the model has high accuracy, given that the top three languages are Latin-based, which share semantic similarities that could

Table 4.1: Distribution of languages

(a)

Portuguese	37.06%
Spanish	11.74%
Italian	7.68%
English	7.59%
Catalan	3.99%
Romanian	2.85%
Welsh	2.31%
Tagalog	2.07%
Lithuanian	2.01%
Estonian	1.88%
French	1.84%
German	1.76%
Indonesian	1.55%
Slovenian	1.49%
Afrikaans	1.36%
Dutch	1.28%
Danish	1.15%
Somali	1.10%
Kiswahili	1.04%

(b)

Norwegian	0.93%
Slovak	0.90%
Latvian	0.86%
Finnish	0.81%
Hungarian	0.78%
Croatian	0.70%
Swedish	0.65%
Unknown	0.59%
Polish	0.38%
Turkish	0.32%
Vietnamese	0.30%
Japanese	0.26%
Albanian	0.23%
Czech	0.19%
Korean	0.14%
Simplified Chinese	0.02%
Hebrew	0.02%
Taiwanese Mandarin	0.008%
Greek	0.002%

potentially challenge the model's classification.

In the study, only Portuguese or English queries, that were detected as such by SpaCy [6] using the "pt_core_news_sm" model [5], were considered, as they are the most significant. Portuguese was chosen for the Arquivo team as it is primarily a Portuguese project, and English because it is the second language available on the website. Furthermore, in Figure 4.1, it can be observed that the queries mostly orig-

inate from a top 5 countries: Portugal and Brazil, which represent the Portuguese language, and three countries that belong to the top 5 of English-speaking countries: the United Kingdom, the United States, and Australia. The normalization process was crucial in analyzing the data, especially for words, and the Cleantext [12] Python package was used to remove extra spaces, lowercase the words, remove punctuation, and eliminate English and Portuguese stopwords as much as possible.

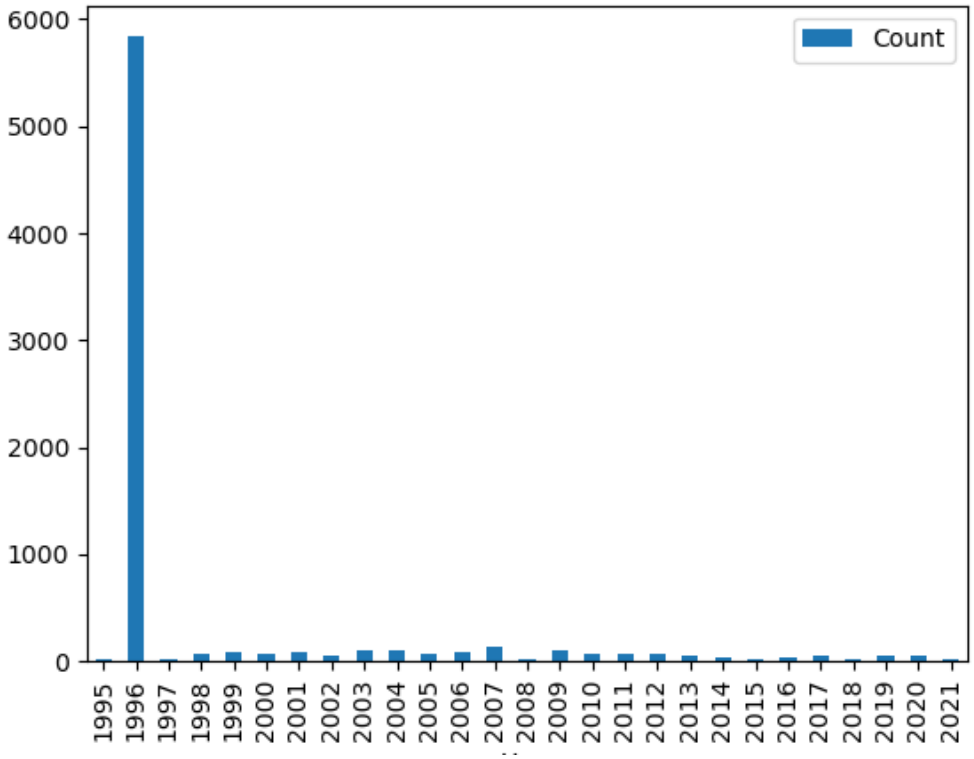

Figure 4.2: Number of "From" filter parameter in searches

After filtering the data, the "PAGE_SEARCH_PARAMETERS" specification provided access to the "from" and "to" parameters, which the user could set when executing their query. Notably, the available data was from 1996 until 2021, making these years the default "from" and "to" parameters for the given data collected in 2021.

In Figure 4.2 we can see that the year 1996 was the most chosen one as a "from"

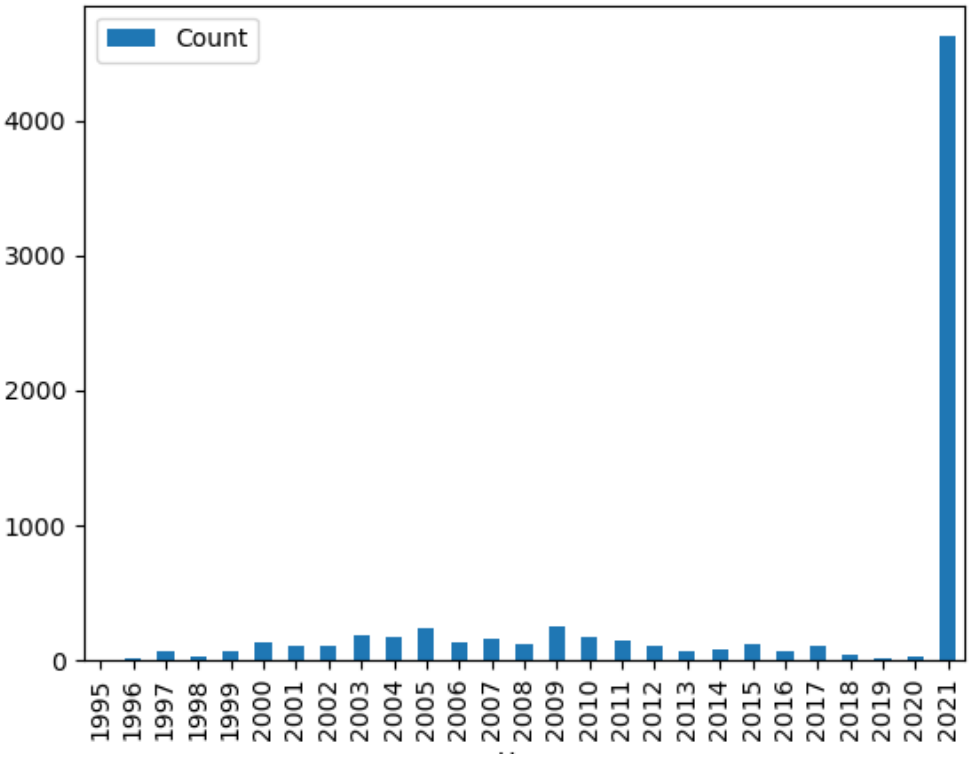

Figure 4.3: Number of "To" filter parameter in searches

parameter. In Figure 4.3 we can see that the year 2021 was the most chosen one as a "to" parameter. The are obviously expected and intuitive results since 1996 and 2021 were default values set for temporal bounders in our dataset.

Figure 4.4 includes queries that fall under the "Not default" category, which refers to those that did not have the default "from" parameter at 1996 and/or "to" parameter at 2021. This finding can lead to the conclusion that approximately 50% of the searches were intended to narrow down the timeframe of the results. Therefore, it can be inferred that at least around half of the users were aware of the feature to customize the timeframe and as they were they seemed to preferably modified the "to" parameter, inspecting the two figures 4.2 and 4.3.

The intentions and thoughts of the user at the time of the search cannot be determined with certainty. While 1996 and 2021 are the default date parameters, it is

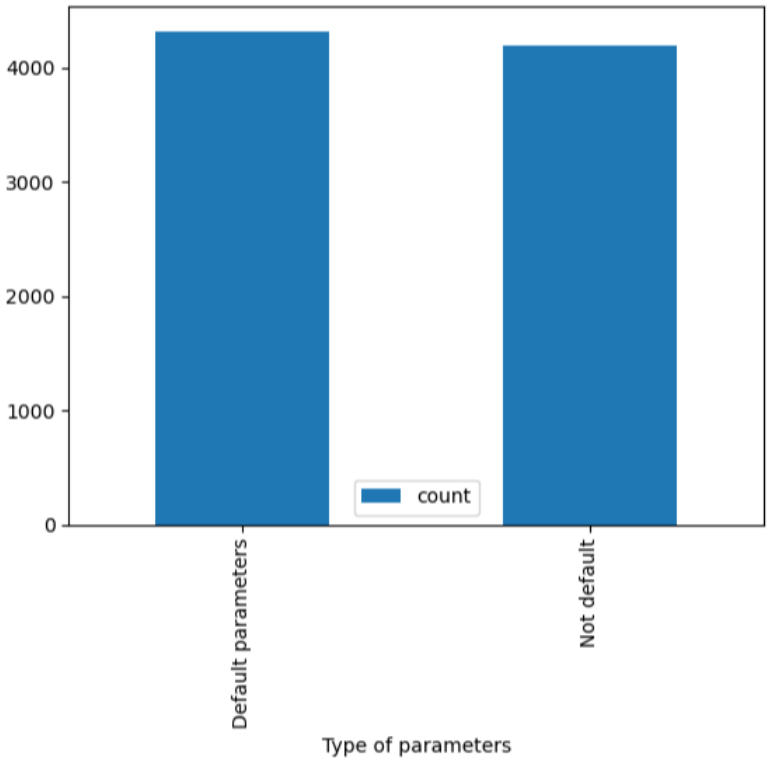

Figure 4.4: Number of queries using default time parameters

unclear if the user intended to restrict their results within this timeframe or if they were unaware of the option to refine the search by date. It is also possible that the user did not have a specific date range in mind and was looking for the most comprehensive results available.

Figure 4.5 reveals that in addition to the majority of users who prefer querying on a classic computer, more than 10% of the users opt to use other devices, such as mobile phones or tablets.

4.2 Full-text queries

Upon conducting a secondary analysis of the dataset, it was discovered that there were 8,571 queries searched by distinct users. The term "unique" here implies that if the same IP address searched for the same query multiple times, it would be

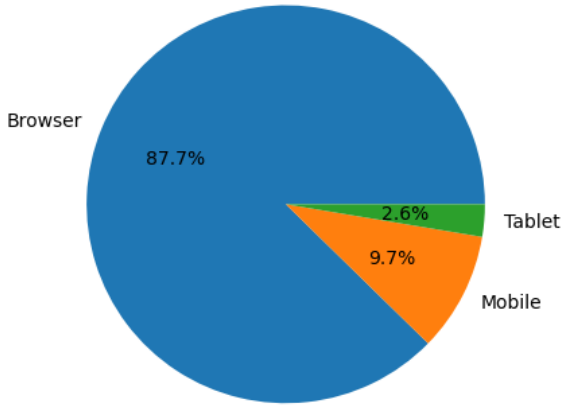

Figure 4.5: Devices the queries were made from

considered as a single instance. Out of those 8571 distinct users, 4197 of the queries were unique. As a general analysis, it can be seen in Table 4.2 that a mere 1% represents URL searches being the flagrant minority. Out of the rest of the queries that are full-text, only 15.1% of terms used are unique.

	Full-text	URL
Unique Queries	98.8%	1.18%
Unique Terms	15.1%	

Table 4.2: Unique Queries statistics

As a reminder of the topic above, some full-text queries are actually containing years. This can be interpreted as if the user was aware of date parameters but had a specific year or version of its query in mind. Out of the 8,571 distinct queries, 346 were found containing at least a year between 1900 and 2050.

Among the 346 queries that included a year, 340 of them had only one year, while two queries had two years and four queries had three years within them.

A more detailed analysis of those queries can be found in figure 4.6, revealing that users seemed to search the most results from the year 2006 and 2001 giving table 4.3 for 2001. From 2001, it could have been expected that some queries would have been about the September 11 attacks as the year was searched a lot but surprisingly most results are about football and higher education.

lista projetos financiados comissao de fomento de investigacao em cuidados de saude 2001	nomes dos estudantes da beira interior nsino superior 2001
tsf 2001	ps horta 2001
candidatos ao ensino superior 2001	rockbeach 2001 radio comercial
comissao de fomento de investigacao em cuidados de saude 2001	futebol praia 2001
listas de colocados no ensino superior 2001	listas do ensino superior 2001
beira interior ensino superior 2001	atlas das cidades 2001
liga de futebol de praia do euro 2001	lista de acesdo ao ensino superior 2001

Table 4.3: Top searched queries with "2001" in it

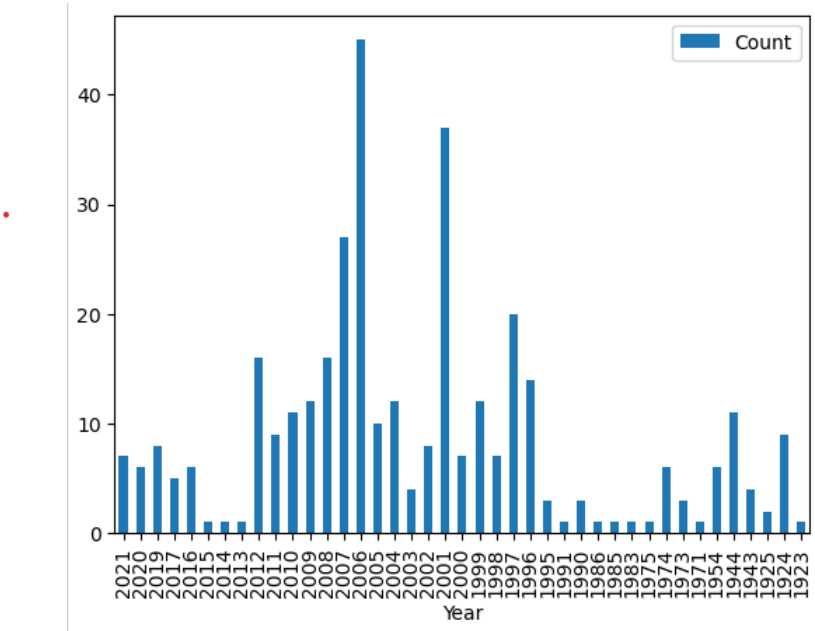

Figure 4.6: Years counted out of queries containing years

Table 4.4 and Figure 4.7 illustrate the distribution of terms per full-text query with Figure 4.8 representing the length of queries expressed as the number of characters. The data shows that the majority of queries consisted of 2 to 5 terms, with 3 terms being the most common and being between 11 and 30 characters. On average,

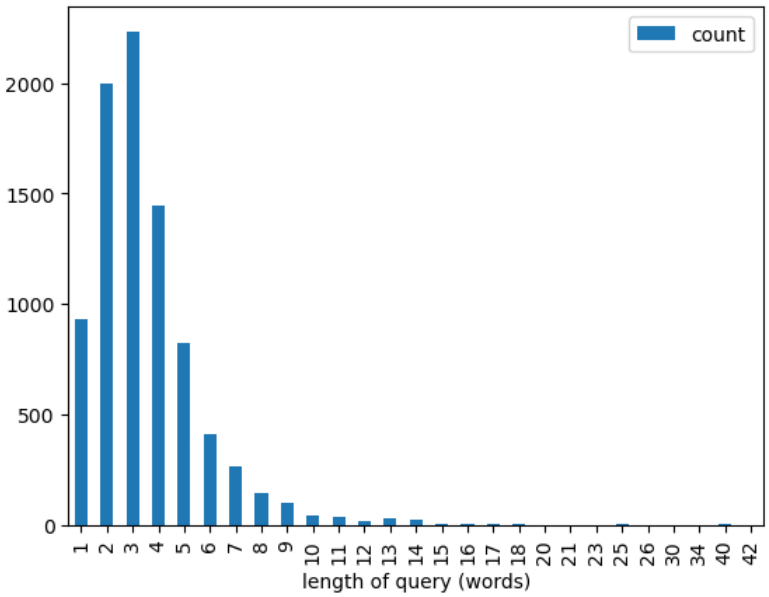

Figure 4.7: Length of queries (words)

# of words	Percentage
1	10.9%
2	23.2%
3	25.8%
4	17.0%
5	15.1%
6	9.6%
7	3.1%
8	1.7%
9	1.1%
10+	2.4%

Table 4.4: Percentage of length of queries (words)

queries consisted of 7.09 terms and 25.2 characters. Almost 92% of the queries had 5 or fewer terms, while only 2.4% had 10 or more terms. This suggests that users tend to submit short queries.

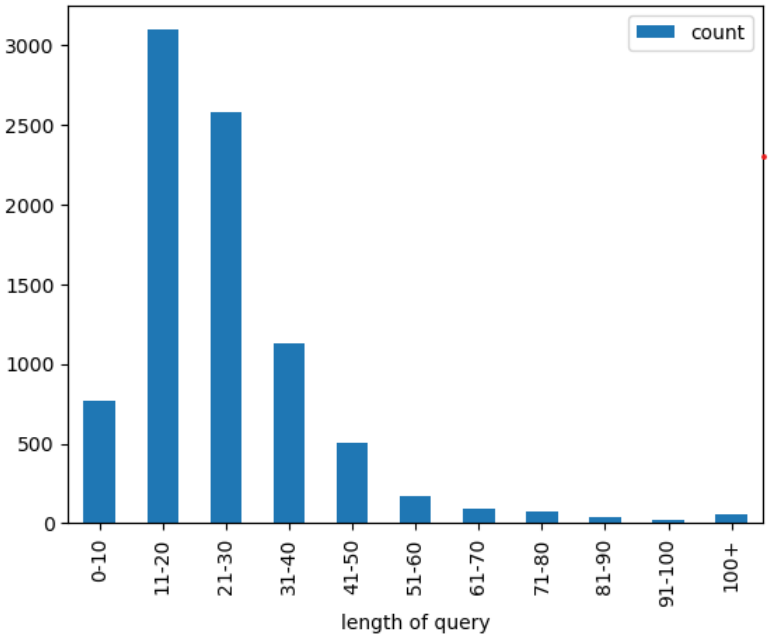

Figure 4.8: Length of queries (characters)

# of queries	% of users
1	39.40
2	17.93
3	9.97
4	5.33
5	3.80
6	3.18
7	2.83
8	2.90
9	1.31
10+	13.29

Table 4.5: Percentage of queries per user

When examining Table 4.5, where only full-text queries were taken into account, it becomes apparent that a bit over 13% of users conducted at least 10 queries. By analyzing the queries and their words more closely, it can be observed that some users performed 192 queries, others 315 and one user went up to a maximum of 325 queries per session.

With such high numbers of queries, one can only be intrigued and it has been decided to dive and have a look precisely what these users were querying about. The first user, the one with 325 queries, searched mostly about hypermarkets and more precisely about their inaugurations 4.6. The second, with 315 queries, has made similar queries about hypermarkets 4.7 and more specifically the chain "Auchan" a french multinational group of stores. The third user sought a completely different subject, they queried about the game "elifoot" 4.8, a football manager computer game where the user can manage a football club. The game was first created in 1987 by the Portuguese André Elias and resold in Brazil since 2004 where it is still one of the most successful game in Brazil.

However, with such a high volume of queries, one can not help but wonder as to whether or not they were conducted by a human.

Query	Number of time searched
"inauguracao hipermercado jumbo"	10
"continente inauguracao hipermercado"	8
"balonas arquitetos"	6
"hipermercado modelo covilha"	6
"inauguracao hipermercado modelo"	5
"hipermercado beja"	4
"centro auto jumbo"	4
"inauguracao worten"	4
"pao de acucar almada"	4
"inauguracao pao de acucar"	4

Table 4.6: Top 10 most searched queries for the user with 325 queries

With the help of the EntityRecognizer from SpaCy [2], queries have been processed to detect what were the most common terms according to the predefined labels of the service. The labels in this case were "LOC" for a location, "ORG" for an organisation, "PER" for a person and "MISC" for miscellaneous terms, results can be seen in Figures 4.10 4.11 4.12 4.13 as word clouds.

Query	Number of time searched
”ampliacao hipermercado”	7
”continente amadora”	5
”auchan pao de acucar santo tirso”	5
”jumbo alverca”	5
”expansao modelo hipermercado”	4
”ampliacao sonae”	4
”continente sao joao da madeira”	4
”almada forum sportzone”	4
”forum algarve localizacao”	4
”umbo setubal grupo auchan”	4

Table 4.7: Top 10 most searched queries for the user with 315 queries

Query	Number of time searched
”index of elifoot”	20
”elifoot ii”	8
”partidos politicos”	7
”guerra zeze camarinha”	7
”elifoot2”	6
”elifoot”	6
”guerra aberta entre zeze camarinha e ronaldo”	5
”elifoot 98”	5
”partidos portugueses”	4
”elifoot 2”	4

Table 4.8: Top 10 most searched queries for the user with 192 queries

A word cloud is a visual representation of text data where the most frequently occurring words in the data set are displayed in a larger font size and the less frequently occurring words are displayed in a smaller font size. It is a way to quickly understand and analyze the most common words or topics in a large body of text. They are used in various fields for a wide analysis. They are often used to identify themes or trends in customer or user feedback, to analyze the language used in political speeches or debates, and to visualize the key topics or concepts in

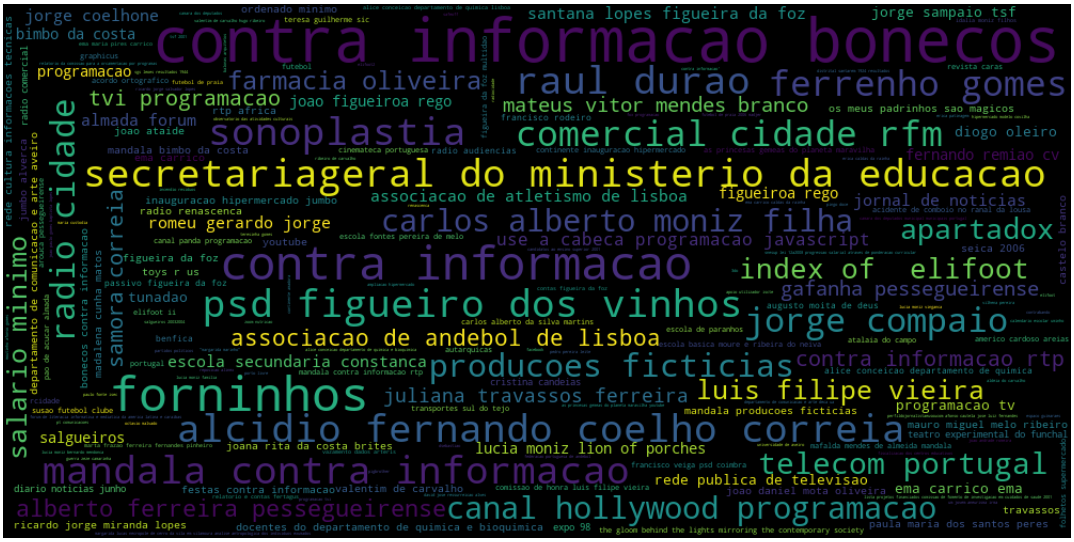

Figure 4.9: Word cloud of queries

academic literature.

In the "LOC" wordcloud depicted in Figure 4.10, the limitations of the entity recognizer can be seen, and it is apparent that the model is not entirely foolproof. However, to better understand the model's analysis, we need to consider how it operates. From the wordcloud, it is easy to deduce that the logs are most likely coming from a Portuguese-speaking country or more specifically from Portugal, given that "lisboa" and "portugal" are the most frequently searched location queries. "Hollywood" appears as well, likely coming from English queries. Some queries, such as "forum de literacia" or "radio cidade", can be logically explained. Although "forum" does not necessarily refer to a location, its primary meaning is an area in the middle of a town used for public business, so it could be interpreted as a location. Similarly, "cidade" means "city" in Portuguese, justifying its presence in the word cloud, where a person would recognize the query being about a media.

Unlike in the previous wordcloud, some queries from the ORG Figure 4.11 can not be explained logically. While some queries do fit like "onu", "partido socialista" meaning "socialist party" and "microsoft", it is not clear why "this boys", "santos" (meaning saint or holy) or "pt" was included in the graph. Generally the label ORG

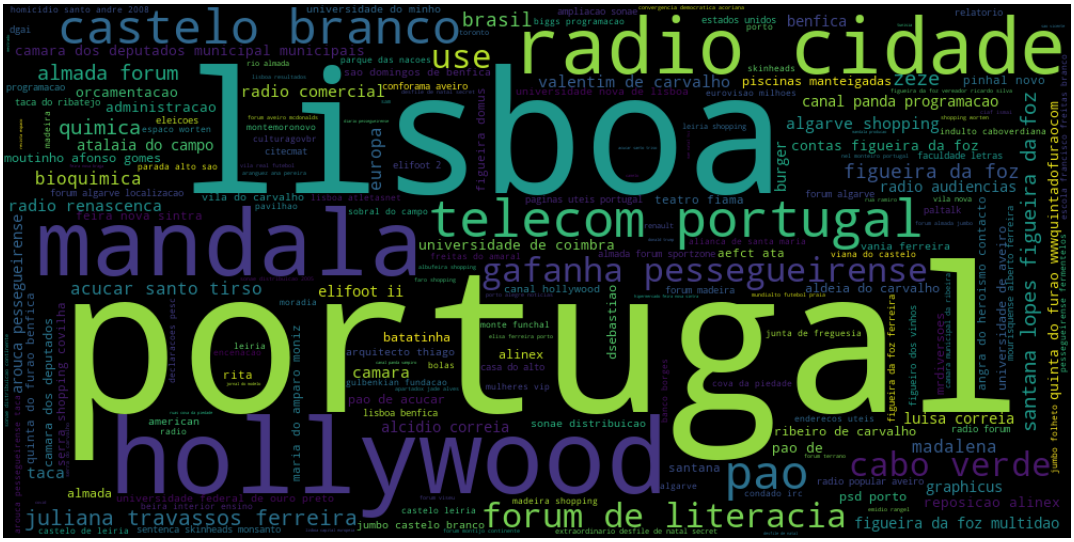

Figure 4.10: Word cloud of queries with the label LOC

can be seen as a label for organizations, companies, clubs and associations as defined by SpaCy : "ORG: "Companies, agencies, institutions, etc." [3].

For Figure 4.12, results seems to be more accurate with some misses like "programacao javascript" of course, most personalities seems to be Portuguese or Brazilian. The personalities searched come from various fields like Alberto Ferreira, a football player, Raul Durao was a journalist and television host or Carlos Alberto Moniz, portuguese musician and songwriter.

MISC is a category from the CoNLL 2003 evaluation data which is typically used to develop NER models. SpaCy gives a definition for the label : "MISC: Miscellaneous entities, e.g., events, nationalities, products, or works of art." [3], which seems to match some of the queries. MISC could simply be defined as the category for an entity that isn't PERSON, ORG, or LOC.

An ultimate word cloud has been made generally for unique queries and see what was searched the most and can be found in Figure 4.9. It can be seen that information about the show "Contra informacao" has been searched a lot. Those results will be again shown in Table 4.11 for our last comparison with Google Trends.

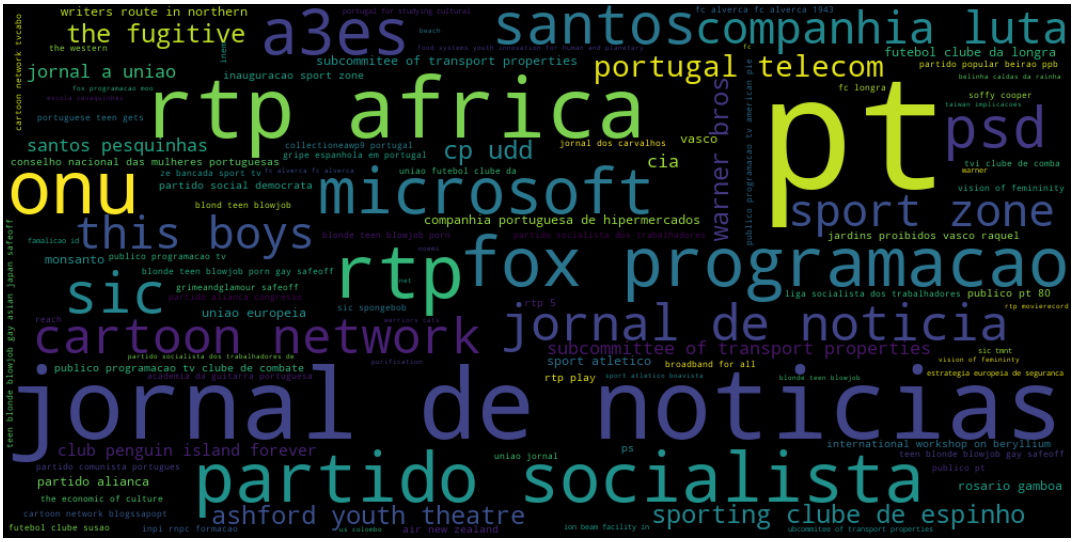

Figure 4.11: Word cloud of queries with the label ORG

Overall, as seen in 4.14, half of the queries could not be classified by SpaCy. However, with nearly 2,000 queries each, users seem to mostly search for location and person, according to the entity recognizer (which is not foolproof).

4.3 URL queries

In the overall analysis of the data, 100 queries were found that were actual working URL queries and not full text queries. The initial attempt was to classify the URLs using a website categorization API [14].

A first result can be seen in figure 4.15. The result shows itself to not be satisfactory as more than half of the analyzed URLs were either not reachable, the language of the website was not supported or there was not enough content on the page, hence they can be considered as not classified. The classified URLs were scattered across different categories, and besides "Technology and Computing", there was no discernible pattern in the users' search behavior.

The decision was then made to manually classify the items after testing multiple website categorization services, which have their own limitations. For instance, the

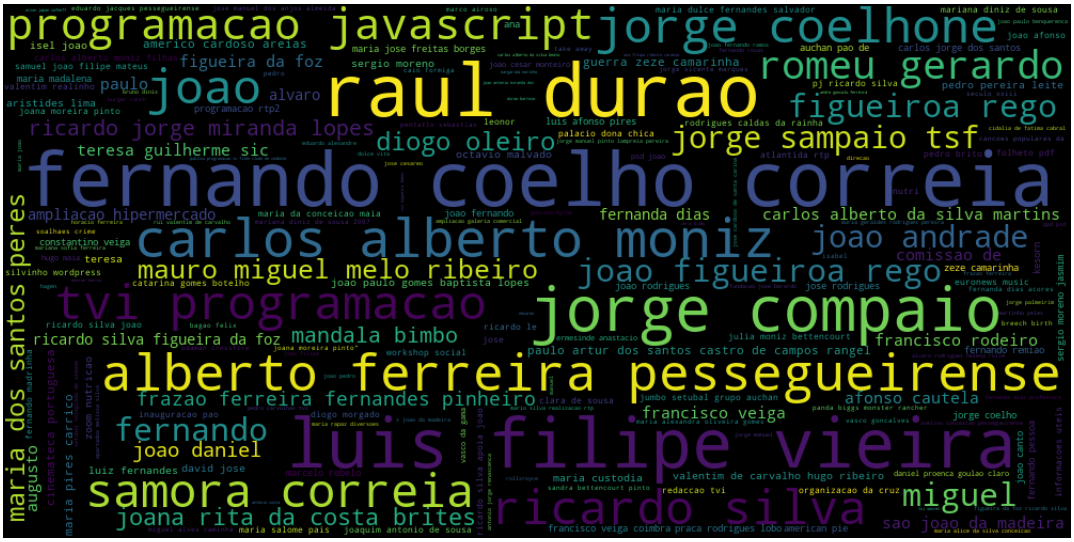

Figure 4.12: Word cloud of queries with the label PER

query must be flawless for the service to locate the item, and the classification is done in real-time, meaning the URL is fetched and categorized immediately upon request. The service uses IAB taxonomy categories [10], which were also chosen to classify full-text queries. Consequently, some websites may not be found if they no longer exist but are still present in the archive. Figure 4.16 displays the resulting graph of manually classifying the same 100 URLs.

Regarding the categories, it was decided to restrain their numbers more than in the first figure, as some categories could be merged, such as "Style and Fashion" and "Shopping" and some categories were not meaningful, like "Personal Finance". Additionally, a new category was identified during classification, which is "Adult" which is not among the last categories.

Despite the fact that 20% of the analyzed URL links were inaccessible or possibly misspelled, we can observe that the most popular search categories were "News and Politics", "Shopping", and "Technology and Computing". These categories are well-suited for web archive searches. For instance, users may want to investigate past products or locations, or track price fluctuations for shopping-related searches. Technology is also a compelling area of research, given its rapid evolution over the

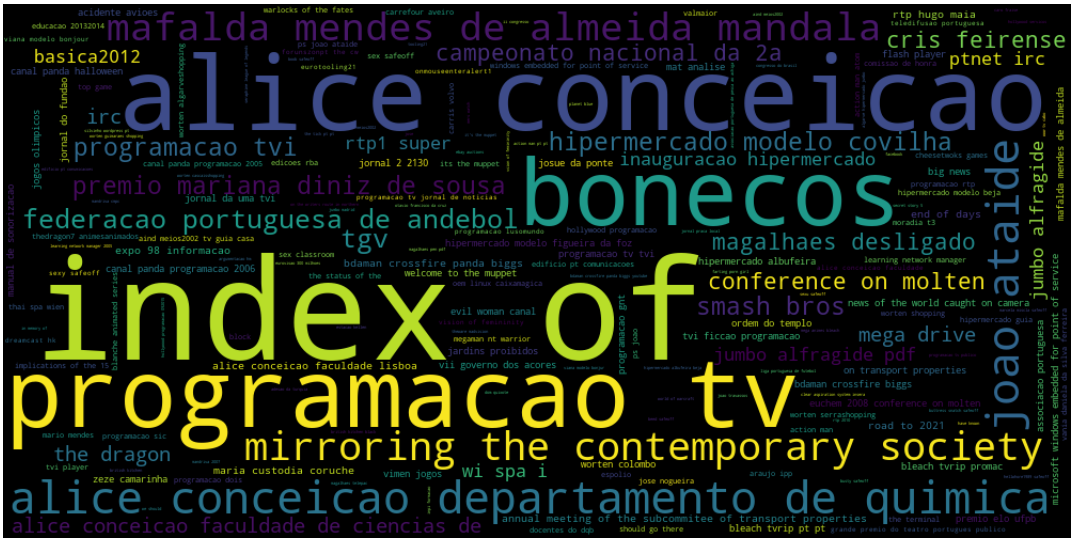

Figure 4.13: Word cloud of queries with the label MISC

past 30 years, and users may be interested in examining prominent technologies from previous years or observing their evolution over time.

4.4 Comparison with Google Trends

As the aim of this paper is to examine user search behavior on an internet archive, it is appropriate to compare the analysis with that of a conventional search engine, specifically Google in our case.

To make our own analysis of a small insight of what could a user behavior be on a web search engine, we used Google Trends. [9]

Google Trends lets you see the topics people follow, conveniently and in real time but also allows you to analyze any period of time. It analyzes the popularity of search terms on Google Search, Google Images, Google News, Google Shopping, and YouTube. Users can enter one or more keywords or phrases and see how their popularity changes over time and in different geographic locations. Journalists can use this information to explore ideas for potential articles and integrate Trends data into their articles to identify interest in, for example, a political candidate, a social

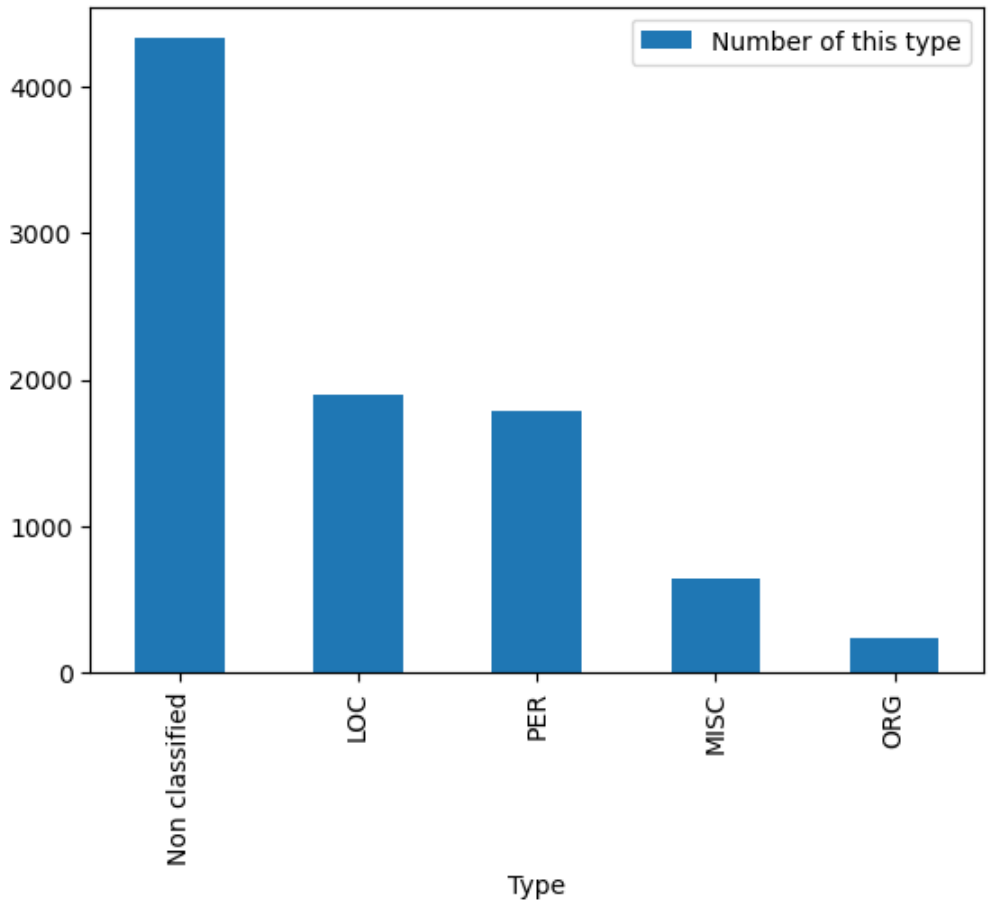

Figure 4.14: Number of queries per label

issue or an event.

It was deemed appropriate for this study to analyze trends in Brazil and Portugal, since these are the two largest countries where Portuguese is spoken. Conveniently, Google Trends allows the user to search results for a specific country during a specific time frame. As the data set is covering period from the 1st of June to the 1st of September 2021, those dates were used as shown in Figure 4.17. On the right side of the query result, a list of the most frequently searched queries within the given time frame by the users is displayed. From these queries, topics are extracted and displayed in the list on the left side.

This list of 25 most searched queries was compared to the top 100 most searched

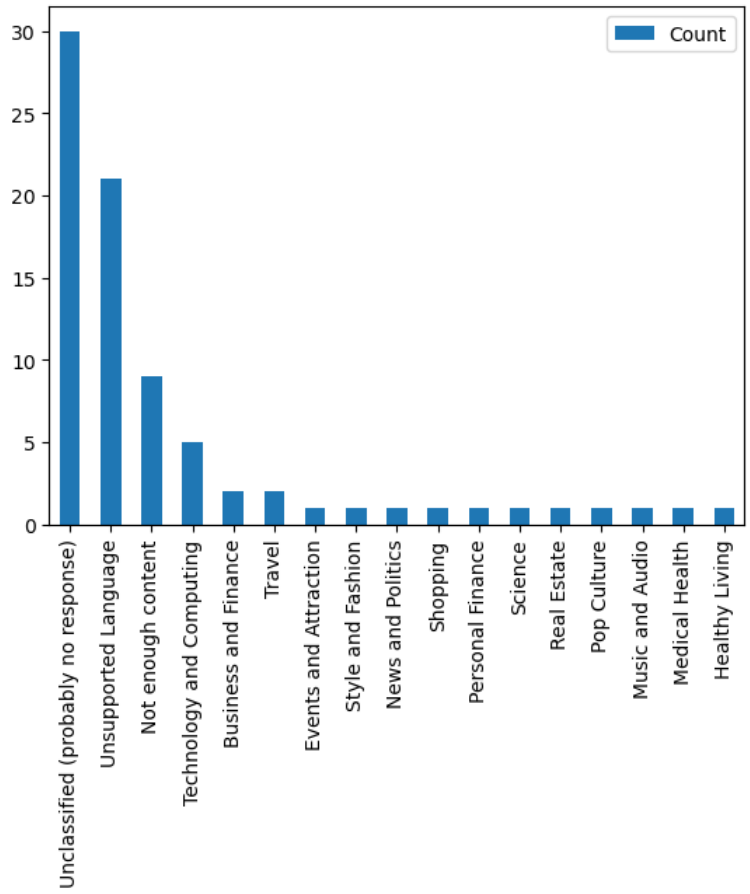

Figure 4.15: URL classification by API

queries in arquivo, tops 25 can be found in Table 4.10 and Table 4.9.

The idea was to compare the queries of arquivo and see if any match was found between the two instances. The expectation was that it would be difficult to find queries that match exactly. To check if people searched for the same topic, each query from Google Trends was normalized in the same way as the queries in arquivo and then compared the two using fuzzy string matching with the fuzzywuzzy package (now called thefuzz [16]). The differences between queries were calculated using the Levenshtein Distance. It is a metric for measuring the difference between two sequences of characters as the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one sequence into the other. The results were filtered, only accepting them when a match of at least 95% was found.

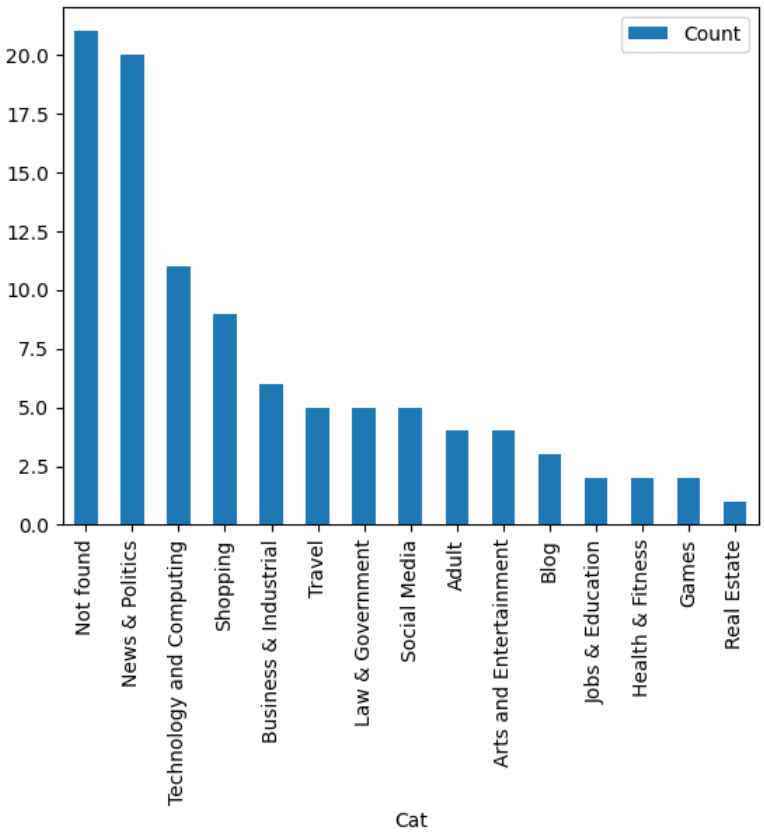

Figure 4.16: URL classification done manually

```
#Function to find the Levenshtein Distance between two strings

def levenshtein(str, tar):
if str == "":
    return len(tar)
if tar == "":
    return len(str)
if str[-1] == tar[-1]:
    cost = 0
else:
    cost = 1

res = min([levenshtein(str[:-1], tar)+1,
```

```
14                levenshtein(str, tar[:-1])+1,
15                levenshtein(str[:-1], tar[:-1]) + cost])
16    return res
```

clima	google	vacina	caixa	hotmail
tempo	flamengo	facebook	vivo	correios
brasil	whatsappweb	previsão do tempo	brasileiro	olx
whatsapp	youtube	instagram	palmeiras	uol
tradutor	globo	clima para amanhã	jogo do bicho	mercado libre

Table 4.9: Google Trends Brazil Top 25 queries

portugal	facebook	hotmail	financas	seguranca social
covid	google	restaurante	gmail	ipma
meteorologia	porto	abola	noticias	benfica
tempo	youtube	olx	record	instagram
tradutor	sapp	tempo para amanhã	jogos	nos

Table 4.10: Google Trends Portugal Top 25 queries

samora correira	salario minimo	producoes ficticias	comercial cidade rfm	raul durao
luis filipe vieira	index of elifoot	canal hollywood programação	secretariageral do ministerio da educacao	alcidio fernando coelho correia
farmacia oliveira	carlos alberto moniz filha	radio cidade	sonoplastia	contra informação
associação de andebol de lisboa	apartadox	jorge compaio	psd figueiro dos vinhos	forninhos
alberto ferreira pessegueirense	telecom portugal	ferrenho gomes	mandala contra informação	contra informação bonecos

Table 4.11: Arquivo Top 25 queries

By comparing the top 25 queries from Google Portugal 4.10, Google Brazil 4.9, and Arquivo 4.11, we can observe significant differences between the two services. Google searches are predominantly current, focusing on social media and means of communication, whereas those from Arquivo are mostly related to people or subjects

that may require research, such as "salario minimo" (minimum wage) for social studies or for civilians interested in following the evolution of their social measures over time. Furthermore, we can notice that there are three different but similar queries related to the TV-show "Contra Informação". It was a political and social satire program featuring dolls that caricature public figures from Portuguese and international society, that aired from 1996 until 2010, probably searched to watch some samples of the show again.

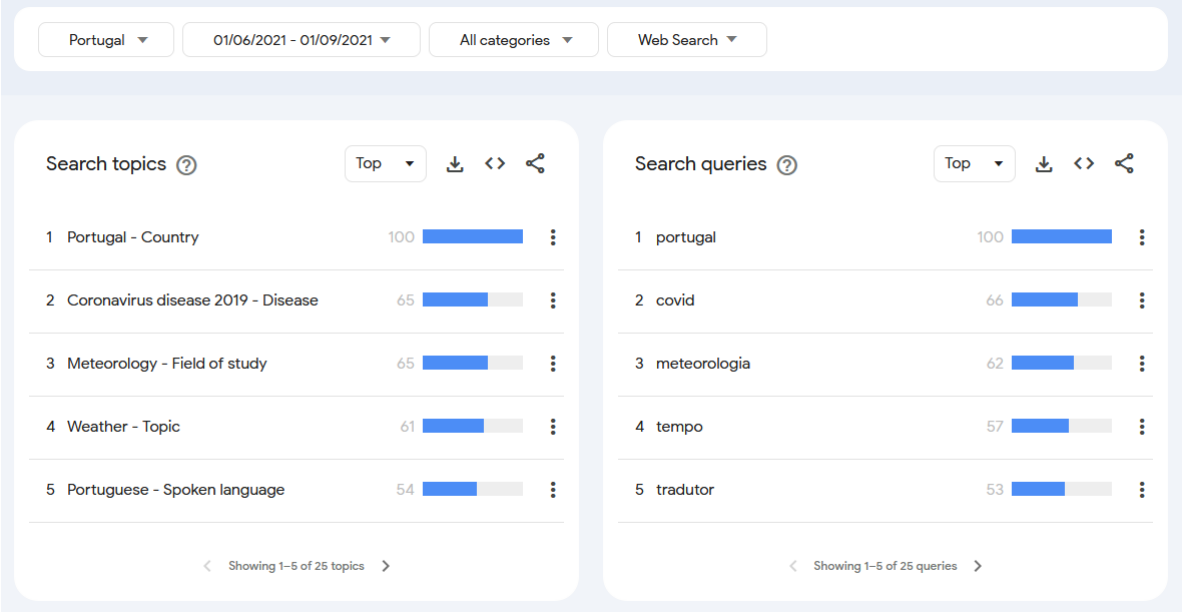

Figure 4.17: Google Trends UI

The result of this comparison can be seen in Figure 4.18 and Figure 4.19. This can be read as, for example in Figure 4.18, 152 queries were found to match the query "portugal" from Google Trends Portugal.

Generally there were more matches found from Google Portugal than from Google Brazil. Some queries appeared in both top 25 lists, such as "hotmail", "facebook", "youtube", and "instagram" as they are popular services and widely used. It is possible that some Brazilian queries were not found in the arquivo queries because they were very time-specific, such as "clima para amanhã" and "previsão do tempo"

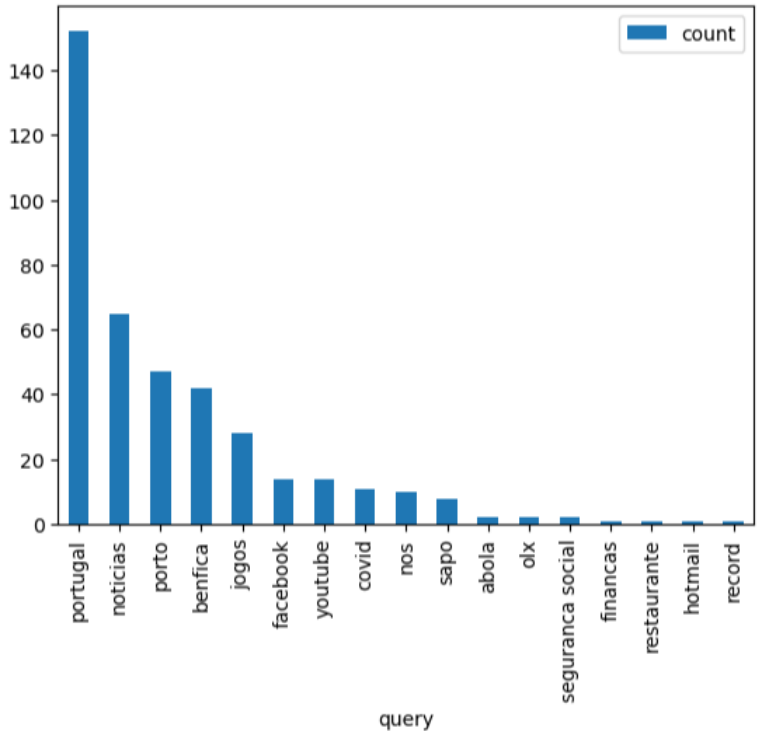

Figure 4.18: Comparing Arquivo and Google Trends Portugal

which respectively mean "weather for tomorrow" and "weather forecast" and may not be relevant for queries made on a web archive. These types of queries are more typical of classic web search engines, which provide quick access to the desired information. Two queries, "whatsapp" and "whatsappweb," are almost identical and may have matched the same queries.

One possible explanation for the high number of matches from Google Portugal is that many queries contain general terms such as "Portugal", "noticias" (meaning "news") or "porto", which are commonly used and relevant for a Portuguese web archive. This suggests that the archive contains a considerable number of queries related to Portugal, its cities or historical news events written in Portuguese.

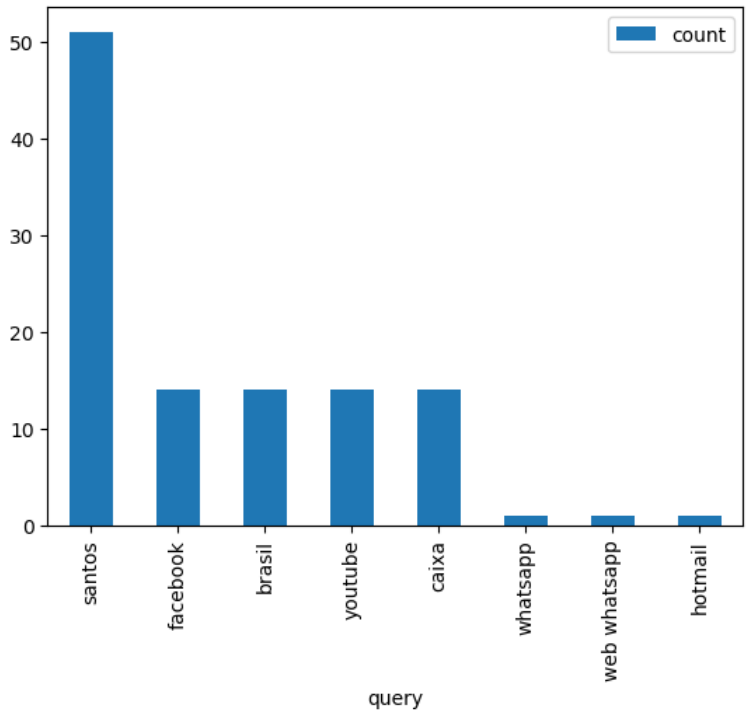

Figure 4.19: Comparing Arquivo and Google Trends Brazil

5 Comparison with previous studies

A previous study "Characterizing Search Behavior In Web Archives" [1] was made in collaboration with arquivo in 2011. In contrast, the analysis in their study was based on a longer period of 7 months, from June to December 2010, more than twice the one of this paper. One of the first noticeable thing is the **user interface that improved a lot over the years, being friendlier to the user**. Additionally, it is worth noting that users now have access to a greater amount of data since the website contains archives dating back to 1991, compared to 1996 in the past.

The first study is showing a percentage of **68.82%** full-text queries and **73.95%** URL queries among unique queries while **the results of this study are much less varied** and show **98.9%**of full-text queries and only **1.18%**of URLs. In the same table it can be found 26.66% of unique terms and 15.1% in this paper as shown in Table 4.2. Despite the current logs being half the size of the previous ones, there is still a significant difference, suggesting that users may be searching for more general information or facts rather than specific websites as **they consequently did more full-text queries than URL queries**. It may also be more difficult to recall a specific URL compared to conducting a keyword search.

An additional point of comparison could be drawn from Table 4.5. In the paper, the number of queries per session was calculated from the time the first query was submitted until the last time the user interacted with the PWA, whereas in this study the number of queries was calculated by IP address regardless of session duration. Despite this difference, the results are worth comparing. The previous study found

that **64.98% of users made only one query per session for full-text queries**, while Table 4.5 in this study reports a different percentage of **39.40% of users making only one query**, which is in **both papers, the highest result**. It is surprising however that a higher percentage of users in this study made 10 or more queries going from 0.58% to 13.29% of users. However, as noted in the initial analysis of the table, **it is impossible to determine if these were made by real humans**.

Furthermore, it can be noticed that users are **averagely using more words when doing their queries** as in this current study we noticed people using mostly between 2 and 5 terms per query, 3 being the most common number of terms, before they used to utilize 1 to 3 terms per query, 1 being the highest used number of terms.This observation suggests that users have developed a **greater level of precision and familiarity with search techniques** on search engines.

About search engines, a similarity between the study and the paper "Exploring and Exploiting User Search Behavior on Mobile and Tablet Devices to Improve Search Relevance" [17] is that both collected data for three months. The paper extracted data from Bing desktop and mobile search engine with over 3 million users, allowing for some comparison with Arquivo. While the previous study had a similar number of users on mobile, tablet, and desktop, the **majority of Arquivo's users accessed the website through a browser** 4.5. However, since Bing is available as an application on mobile and tablets and can also be the default search engine on a browser, it is worth noting that Arquivo's statistics include users accessing the website on-the-go through their mobile or tablet devices.

A difference that can be observed is that the study found a **shorter average length for queries in terms of both words and characters** . The average number of words per query was 2.88 and the average number of characters was 12.65. This is significantly lower than the average of 7.09 words and 25.2 characters observed in our study on Arquivo. This may suggest that **Arquivo users tend to be more**

precise in their searches, requiring more words or parameters in their queries because they have a clearer idea of what they are searching for and the purpose of a classic search engine is not the same as the one of a web archive.

6 Summary and Outlook

In this paper,a detailed analysis of user search behavior in a temporal web repository has been presented, based on an examination of search query logs spanning a period of 3 months from June to September 2021. The examination of various metrics such as query length, query type, and query frequency, allowed for a deeper understanding of how users interact with the archive and the nature of their search queries. This investigation revealed a number of key trends and patterns in how users search for information in this type of digital archive, and highlighted several areas where future research could further build on these findings.

Users of Arquivo.pt are seemingly tending to use longer queries with more words and characters compared to previous studies on search behavior but also compared to a classic search engine. The analysis also revealed a high percentage of full-text queries with a relatively small number of URL queries, indicating that users may be more interested in finding specific information rather than navigating to a particular website or simply by forgetting the URL which is more difficult to recall than doing a simple keyword search. Additionally, it was observed that users are more likely to make multiple queries than a single query, possibly indicating a need for refining their search queries or exploring multiple avenues of inquiry, but they could also be more curious or searching by nostalgia. This could be a side effect of a friendlier UI which make it easier to search and find results. Moreover, the average length of search queries is longer than in previous studies, suggesting that users may have a more precise idea of what they are searching for, also suggesting that users are

becoming more aware of effective search techniques.

Furthermore, about 50% of user queries analyzed in the study modified at least one side of the temporal parameter, indicating awareness of temporal sorting. However, users appeared to modify the "from" parameter more frequently than the "to" parameter, suggesting intentional behavior. Nonetheless, it is worth noting that not modifying the dates does not necessarily indicate a lack of awareness of their existence. A small percentage of users also included specific years in their queries to define sorting parameters, potentially suggesting that the existing time-restricting parameters are not specific enough. However, this behavior was observed in only a minority of users.

Overall, the study highlights the importance of understanding user behavior in web archives, both for improving the user experience and for gaining insights into how people interact with and make sense of the vast amounts of digital content that are being generated and stored online every day. These findings can be valuable to researchers and developers looking to improve the search functionality of web archives and provide a better user experience for those accessing them. Overall, this analysis provides important insights into the search behavior of users in temporal web repositories and highlights the need for continued research in this area.

Bibliography

[1] Costa, M. and Silva, M. J. [2011], Characterizing search behavior in web archives, *in* C. Bizer, T. Heath, T. Berners-Lee and M. Hausenblas, eds, 'WWW2011 Workshop on Linked Data on the Web, Hyderabad, India, March 29, 2011', Vol. 813 of *CEUR Workshop Proceedings*, CEUR-WS.org, pp. 33–40. **URL:** *https://ceur-ws.org/Vol-707/TWAW2011-paper5.pdf*

[2] Explosion [2023*a*], 'EntityRecognizer spaCy', `https://spacy.io/api/entityrecognizer`.

[3] Explosion [2023*b*], 'GitHub SpaCy', `https://github.com/explosion/spaCy/blob/master/spacy/glossary.py`.

[4] Explosion [2023*c*], 'Models and Languages', `https://spacy.io/usage/models`.

[5] Explosion [2023*d*], 'Portuguese SpaCy', `https://spacy.io/models/pt`.

[6] Explosion [2023*e*], 'spaCy', `https://spacy.io/`.

[7] Gomes, J., Gomes, D., Esteveira, F., Gomes, P., Rato, V. and Basílio, R. [2023*a*], 'Arquivo.pt', `http://arquivo.pt`.

[8] Gomes, J., Gomes, D., Esteveira, F., Gomes, P., Rato, V. and Basílio, R. [2023*b*], 'What is Arquivo.pt', `https://sobre.arquivo.pt/en/help/what-is-arquivo-pt/`.

[9] Google [2023], 'Google Trends', `https://trends.google.com/home`.

[10] Hexasoft [2023], 'What are IAB categories ?', `https://blog.ip2location.com/knowledge-base/what-are-iab-categories/`.

[11] IPinfo [2023], 'IPinfo', `https://ipinfo.io/`.

[12] MIT [2023], 'Cleantext', `https://github.com/prasanthg3/cleantext`.

[13] Ntoulas, A., Cho, J. and Olston, C. [2004], What's new on the web?: the evolution of the web from a search engine perspective, *in* 'WWW '04: Proceedings of the 13th international conference on World Wide Web', Association for Computing Machinery, New York, NY, United States, pp. 1–12.
URL: *https://dl.acm.org/doi/10.1145/988672.988674*

[14] Quantum, A. [2023], 'Website Categorization', `https://www.websitecategorizationapi.com/demo_dashboard_iab/index_url.php`.

[15] Roshdi, A. and Roohparvar, A. [2015], 'Review: Information retrieval techniques and applications', *International Journal of Computer Networks and Communications Security* **3**(9), 373–377.

[16] Several [2023], 'TheFuzz', `https://github.com/seatgeek/thefuzz`.

[17] Song, Y., Ma, H., Wang, H. and Wang, K. [2013], 'Exploring and Exploiting User Search Behavior on Mobile and Tablet Devices to Improve Search Relevance', `http://sonyis.me/paperpdf/fp016-songPS-www2013.pdf`.

[18] Stamps [2023], 'Python User Agents', `https://pypi.org/project/user-agents/`.

[19] Team, I. A. [2023], 'Internet Archive', `https://archive.org/`.