



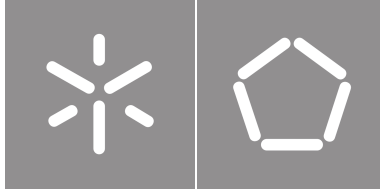
Universidade do Minho

Escola de Engenharia

Constança Machado Aires Lobo Elias

**Twitter Observatory: developing tools to
recover and classify information for the
social network Twitter**

October, 2022



Universidade do Minho

Escola de Engenharia

Constança Machado Aires Lobo Elias

**Twitter Observatory: developing tools to
recover and classify information for the
social network Twitter**

Master Thesis

Master in Informatics Engineering

Work developed under the supervision of:

Miguel Francisco Almeida Pereira Rocha

Vítor Manuel Sá Pereira

October, 2022

COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

This is academic work that can be used by third parties as long as internationally accepted rules and good practices regarding copyright and related rights are respected.

Accordingly, this work may be used under the license provided below.

If the user needs permission to make use of the work under conditions not provided for in the indicated licensing, they should contact the author through the RepositoriUM of Universidade do Minho.

License granted to the users of this work



**Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International
CC BY-NC-SA 4.0**

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en>

Acknowledgements

My first acknowledgements go to my supervisor, Dr. Miguel Rocha, and co-supervisor, Dr. Vitor Pereira. Your expertise helped me make the right decisions and focus on the purpose of this work. Thank you.

Secondly, I would like to thank OmniumAI for giving me the opportunity to do an internship this year and collaborate with them on the company's product. During the last year, I have been able to improve my work methodology and learn something new every day. I would specially like to thank Nuno Alves, Fernando Cruz and Rúben Rodrigues, who supervised me in OmniumAI. Your availability and guidance enabled me to conclude this project and produce the desired outcomes. Then, I thank Jorge Gonçalves, Tiago Silva and Miguel Barros, for sharing this year with me at OmniumAI. A special thanks to Miguel, for the availability, help and advice in the right time.

Additionally, I would also like to manifest my gratitude to all my friends for the moral and technical support. At the end of the day, all wouldn't be possible without them in the background. In the first place, I would like to thank my everyday colleagues who became friends. To a long-time friendship, Maria Araújo, for being a very supportive friend with a huge heart and a great sense of perseverance at work. To Vasco Ramos, for his knowledge and expertise, for helping me becoming more professional, for pulling me along and for being a friend who brings people together. To Carolina Marques and Renata Ribeiro, known more recently, for your happiness, empathy, support and affection. You will become friends for life. To Luís Ferreira, an amazing team leader from whom I learned a lot, professionally and personally, and I am grateful to have met this year. Last but not list, to many other friends from different areas of my life who have shared many special moments and memories with me and have helped me become the greatest version of myself. Life would not be so rich without them.

To conclude, a huge thanks to my family. To my mother, for the unconditional support, patience, and hope every day. Thank you for helping me persevere in my work, have confidence in my skills, and have a passion for what I do. To my father and sister for always being there for me, for showing me love and giving me moral support. Thank you, family.

E porque *estes anos são viagem* e a minha viagem engloba todas estas pessoas, pelas quais estou muito grata, resta-me deixar umas palavras em português. Contem sempre comigo!

Muito obrigada a todos,

Constança

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

“It always seems impossible until it’s done.” (Nelson Mandela)

Resumo

Twitter Observatory: desenvolvimento de ferramentas para recolha e classificação de informação da rede social Twitter

As redes sociais tornaram-se na nova forma de comunicar e, consequentemente, uma importante fonte de informação. Mais concretamente, o Twitter, desde a sua criação, tornou-se numa das redes sociais mais utilizadas. Esta popularidade permitiu um aumento do número de investigações na área de *Text Mining* usando o Twitter para diferentes aplicações, como saúde e política. Nesta área, a classificação de documentos tem sido aplicada a vários dados, nomeadamente *tweets*, para analisar tendências, entender o comportamento humano e prever determinados eventos. No entanto, nem sempre é possível ter os *datasets* desejados para efectuar essa classificação e análise.

Para resolver o problema encontrado, esta dissertação, proposta pela OmniumAI, pretende explorar as abordagens já existentes para a extração e classificação de dados do Twitter, focando-se principalmente na língua portuguesa. Para isso, foi desenvolvida uma API capaz de extrair *tweets* de acordo com um determinado tópico de interesse, e criar *datasets* classificados automaticamente com *labels* de relevância. Foi ainda desenvolvida uma pipeline de classificação de *tweets* com base nas abordagens de *Deep Learning* encontradas no Estado de Arte para a classificação de documentos. O produto final consiste numa *framework*, *Twitter Observatory*, que permite aos utilizadores criar *datasets* de acordo com um determinado tópico de interesse e analisar esses mesmos *datasets*.

Para testar a *framework* desenvolvida, foram seleccionados dois casos de estudo: COVID-19 e a Invasão Russa da Ucrânia em 2022. Relativamente a estes dois tópicos, dois *datasets* foram extraídos e classificados de acordo com a relevância dos *tweets*, contendo, respetivamente, 2,268,575 e 219,887 *tweets* em português. Foi feita uma análise exploratória destes dados e os resultados de classificação usando modelos de *Deep Learning* foram apresentados. Para validar esses resultados, foi utilizado o dataset existente *CrisisLex*, traduzido para português.

Palavras-chave: Twitter, Classificação de Documentos, *Deep Learning*, Língua Portuguesa

Abstract

Twitter Observatory: developing tools to recover and classify information for the social network Twitter

Social media have become the new form of communication and, therefore, an important source of information. More specifically, Twitter, since its foundation, became one of the most used social media platforms. Its popularity enabled the creation of an enormous amount of content, and a lot of research has been done using Twitter in different areas, such as health and politics. In the text mining field, document classification has been applied to Twitter to analyse trends, human behaviour or predict some events. However, it is not always possible to have the desired datasets to perform the classification and analysis.

To solve the problem described, this dissertation, proposed by OmniumAI, aims to explore existing approaches to extract and classify Twitter data, in particular regarding the Portuguese Language. For that, it was developed an API capable of extracting tweets according to a given topic of interest, and creating datasets automatically classified with relevance labels. A classification pipeline of tweets was also developed based on the Deep Learning approaches found in the State of the Art for document classification. The final product consists of a framework, *Twitter Observatory*, that allows users to create datasets according to a particular topic of interest and analyse those datasets.

To test the developed *framework*, two case studies were selected: COVID-19 and the Russian Invasion of Ukraine in 2022. Regarding these two topics, two *datasets* were extracted and automatically labelled according to the relevance of the tweets, containing, respectively, 2,268,575 and 219,887 tweets in Portuguese. An exploratory analysis of this data was performed and the classification results using *Deep Learning* models were presented. To validate those results, it was used an existing dataset, the *CrisisLex* dataset, translated into Portuguese.

Keywords: Twitter, Document Classification, Deep Learning, Portuguese Language

Contents

List of Figures	xi
List of Tables	xiii
Acronyms	xiv
1 Introduction	1
1.1 Context and Motivation	1
1.2 Objectives	2
1.3 Document Structure	3
2 State of the Art	4
2.1 Social Media Text Mining	4
2.1.1 Data extraction	5
2.1.2 Preprocessing	7
2.1.3 Feature Extraction	8
2.1.4 Embeddings	8
2.1.5 Document Classification	9
2.1.6 Topic Modelling	10
2.2 Machine Learning	10
2.2.1 Naïve Bayes	11
2.2.2 Support Vector Machines	11
2.2.3 k-Nearest Neighbors	11
2.3 Deep Learning	11
2.3.1 Convolutional Neural Network	12
2.3.2 Recurrent Neural Network	12
2.3.3 Long Short-Term Memory	13
2.3.4 Attention Mechanism	13
2.3.5 Hierarchical Attention Mechanism	14
2.3.6 BERT	14

2.3.7	XLNet	15
2.4	A case study: Twitter	16
2.4.1	Portuguese Twitter Data	17
2.5	Related Work	19
2.6	Discussion	21
3	Development Methodology	23
3.1	General Approach	23
3.2	Extraction Pipeline	24
3.3	Classification Pipeline	25
3.4	The Final Framework	26
4	Implementation	27
4.1	Twitter Extractor	27
4.1.1	Functional Requirements	27
4.1.2	Twitter API Limitations	28
4.1.3	Twitter Extractor Implementation	30
4.1.4	Exploratory data analysis module	34
4.1.5	Topic Modelling	35
4.2	Tweets Classification	35
4.2.1	Preprocessing	36
4.2.2	Classification Models	38
4.3	Web Application	39
4.3.1	Architecture	39
4.3.2	Components	39
5	Results and Discussion	42
5.1	Datasets	42
5.1.1	COVID-19	44
5.1.2	Russian invasion of Ukraine	47
5.1.3	Validation Dataset - CrisisLex	49
5.2	Twitter Extractor Evaluation	50
5.3	LDA results	50
5.4	Deep Learning Results	52
5.5	Discussion	53
6	Conclusion	54
6.1	Summary of the work	54
6.2	Contributions	55

6.3 Future Work	56
Bibliography	58
Appendices	
A COVID-19 Portuguese Twitter Accounts	69
B COVID-19 Dataset - Seed For Extraction	70
C Russian Invasion of Ukraine Dataset - Seed For Extraction	71

List of Figures

1	The text mining process. Adapted from [16]	5
2	Word2vec model (Continuous Bag of Words (CBOW) and Skip-Ngram). Adapted from [51]	9
3	The basic architecture of deep learning-based approach. Adapted from [21]	12
4	The Recurrent Neural Network (RNN) based model (left) and the Convolution Neural Network (CNN) based model (right). Adapted from [23]	13
5	Example of a context annotation object for a tweet classified with four different topics (fragment of a tweet payload from Twitter API v2)	18
6	General methodology approach to the identified problem	24
7	Extraction Pipeline to create a dataset according to a certain topic	25
8	Classification Pipeline for tweets according to their relevancy	26
9	Twitter Extractor architecture diagram	31
10	Twitter Extractor API - Endpoints Diagram	32
11	Collecting area of European Portuguese tweets	33
12	Example of a request body for tweets-seed endpoint	33
13	COVID-19 dataset - data extracted per day	35
14	Pipeline used for Deep Learning models	38
15	Twitter Observatory architecture diagram	39
16	Portion of the Twitter Observatory database page for the COVID-19 dataset obtained with the Twitter Extractor	40
17	Twitter Observatory extraction page	41
18	COVID-19 dataset - data extracted per day labelled as relevant	45
19	Characters frequency of COVID-19 dataset tweets	46
20	Hashtags correlation for COVID-19 dataset	46
21	Most common bigrams for COVID-19 dataset related tweets	47
22	Russian Invasion of Ukraine dataset - data extracted per day labelled as relevant	48
23	Most common bigrams in tweets of Ukraine War dataset	48
24	Characters frequency of COVID-19 dataset tweets	49
25	Latent Dirichlet Allocation (LDA) Topics for COVID-19 dataset	51

LIST OF FIGURES

26	Extraction seed for extracting COVID-19 dataset using the Twitter Extractor	70
27	Extraction seed for extracting Russian Invasion of Ukraine dataset using the Twitter Extractor	71

List of Tables

1	Twitter datasets for relevance classification	6
2	Comparison of different deep learning techniques for document classification. Adapted from [73]	16
3	Previous approaches using Deep Learning (DL) models regarding mainly Twitter data . .	20
4	Comparison of different extracting frameworks for Twitter (commercial and non-commercial)	20
5	Twitter API access levels	28
6	Python wrappers for the Twitter API	29
7	Collections of a database	34
8	Preprocessing tasks	37
9	Extracted Datasets characteristics	44
10	PyLDAviz print-screen of the visualisation of a topic from COVID-19 dataset	51
11	Train, validation and test split of COVID-19 and Russian Invasion of Ukraine datasets for BERTimbau model	52
12	Results obtained on the test set for COVID-19 dataset and for Russian Invasion of Ukraine dataset	53

Acronyms

API	Application Programming Interface 28, 30, 40, 57
BERT	Bidirectional Encoder Representations from Transformers 14, 15, 38, 55
BoW	Bag of Words 8
CBow	Continuous Bag of Words 8
CNN	Convolution Neural Network xi, 12, 13
DL	Deep Learning xiii, 2, 3, 11, 15, 19, 20, 21, 25, 35, 36, 38, 43, 52, 55
EDA	Exploratory Data Analysis 31, 34, 40, 45
ELMo	Embeddings from Language Models 14
GTP	Generative Pre-Trained Transformer 14
HAN	Hierarchical Attention Network 14
k-NN	k-Nearest Neighbors 10, 11
LDA	Latent Dirichlet Allocation xi, 10, 35, 36, 50, 51, 56
LSTM	Long Short-Term Memory 13, 14
ML	Machine Learning 2, 3, 11, 19, 21, 25, 35, 36, 38
MNB	Multinomial Naive Bayes 11
NB	Naïve Bayes 10, 11
NER	Named Entity Recognition 15, 38, 52
NLP	Natural Language Processing 4, 14, 15, 19, 38, 52

NLTK	Natural Language Toolkit 7 , 36
RAM	Random Access Memory 50
RNN	Recurrent Neural Network xi , 12 , 13
RTE	Recognizing Textual Entailment 15 , 38 , 52
SSD	Solid State Drive 50
STS	Sentence Textual Similarity 15 , 38 , 52
SVM	Support Vector Machines 10 , 11
TF-IDF	Term Frequency-Inverse Document Frequency 8 , 11
UI	User Interface 23 , 26 , 27 , 39 , 40 , 41 , 45 , 53 , 54 , 55 , 56 , 57

Introduction

1.1 Context and Motivation

Social Networking platforms are very popular these days. Millions of users are registered on these websites and exchange their thoughts, opinions, news and personal information, in different forms such as text, photos and videos. Social media act as sources of public opinion, thus helping in knowing and understanding what the public is talking about [1].

Regarding the use of social networks, Twitter has become one of the most popular microblogging services on the Internet. As defined by Twitter itself, *"Twitter is where people go to find out what's happening in the world right now. Whether you're interested in music, sports, politics, news, celebrities, or every day moments, come to Twitter to see what people are talking about and join the conversation."* [2]. According to Torales et al. [3], there are at least four reasons that make people use Twitter instead of other social platforms on scientific research: the post are mainly text-based (in opposition to other social networks that have more disperse content like images and videos); there is a variety of web scraping tools available for meaningful tweet extraction; it is a high engagement platform; and, each post has geolocation (which is useful to extract only from specific countries or regions). The massive amount of information over the web from Twitter requires an automatic tool to determine the topics that people are discussing [4].

The scientific research done so far over Twitter data to employ various text mining tasks normally uses existing datasets or, when extracting datasets for the use case, do not have well-founded criteria to extract data. Instead of that, the extraction is based only on keywords or hashtags that *seem* to be the right input to extract the intended data and not on a well-defined algorithm. This means that there is a lack of studies in which extraction methods are better to get the most out of the available data on Twitter, regarding the use case of the study.

In which concerns to the Portuguese language, few papers in the literature have employed text mining tasks over Portuguese texts [5], particularly using Twitter data. A recent survey [6] points out that text mining tasks developed for English and applied in datasets with texts translated into English perform better than using a Portuguese dataset. This conclusion highlights the need for new proposals for the Portuguese language, as the translation process can lead to the loss of information from one language to another. In which concerns text mining tasks, particularly Document classification, the process of categorising documents in certain categories [7], DL models are widely used for better classification performance compared to Machine Learning (ML) algorithms [8, 9]. Recently, transformer-based models have been developed and reached promising results regarding this task. Currently, XLNet outperforms BERT on 20 tasks and achieves state-of-the-art results on 18 tasks including document ranking [10]. On this basis, it is urgent to apply those results in Twitter's Portuguese data to learn which messages are relevant to a given topic.

In summary, there are two detected problems in this field and they are the main motivation to develop this work: the data extraction, which is not well defined and does not allow researchers to have (enough) data satisfying the needs of the study, regarding particularly Twitter; and, document classification is not properly explored for Portuguese, specifically regarding the mentioned State of the Art results. Therefore, by exploring these two problems this project can contribute to the Portuguese-speaking community, as the Portuguese is ninth most spoken language in the world [11].

1.2 Objectives

Taking into consideration the two problems described on the previous section, the main objective of this dissertation is to develop a system capable of collecting data about a topic of interest from Twitter and classify it as being related or not to the topic. This will imply developing algorithms to select sets of hashtags, keywords and users (from an initial set) to extract data, but also to develop a framework that is able to learn which messages are relevant to the given topic, based on Machine/ Deep Learning approaches, being able to create datasets of interest.

More specifically, the work will address the following scientific/technological objectives:

1. Review the state-of-the-art for ML/DL methods and their applications in document relevance classification tasks;
2. Research relevant literature for Twitter extraction and classification, with an emphasis in the Portuguese language;
3. Develop algorithms to define adequate queries to Twitter data extraction methods that allow to obtain texts regarding given topics and apply those to selected case studies;
4. Develop and compare different data preparation methods and assess the performance impacts of each one;

5. Train and evaluate the performance of different DL models to assess the relevance of tweets in the Portuguese language and apply to selected case studies;
6. Develop a software framework to incorporate those models and the final data retrieval and classification pipelines, which may be adaptable to different case studies;

This final framework resulting from this work will be integrated in OmniumAI software. At the end of this project, results should answer the problem defined: can a structured extraction method as well as a defined classification approach lead to obtain useful datasets for achieving the state-of-the-art results for document classification tasks?

1.3 Document Structure

Apart from this chapter, this dissertation is structured in 5 chapters.

Chapter 2 (State of the Art) summarises and analyses the State of the Art regarding text mining approaches over document classification tasks and the use of Twitter data to assess those approaches. It starts with a review of the literature regarding Text Mining, followed by a detailed overview of ML and DL techniques used for two specific Text Mining Tasks: document classification and topic modelling. The last sections mention recent research on Twitter data, focusing mainly on the Portuguese language.

Chapter 3 (Development Methodology) explains, in a high level, the development methodology adopted in this project. First, the general approach to the problem is detailed, based on a three-phase development. Then the proposed pipelines for extraction and classification are explained. And finally, an overview of the main goals and strategy for the final framework is given.

Chapter 4 (Implementation) details the implementation of the framework, named *Twitter Observatory*. It starts by describing the tool developed to extract data, then the classification methods implemented and finally the incorporation of this two software components in the final framework of this project. The software architecture defined as well as the technologies used are analysed and the final user interface is presented.

Then, in Chapter 5 (Results and Discussion), the results obtained with the framework are explained and analysed regarding the initial purpose of this project. First, the selected use cases are explained and then the datasets obtained are present and discussed, regarding the existing datasets found in the literature.

Finally, Chapter 6 (Conclusion) summarises the developed work for this dissertation. The main conclusions of this project as well as the contributions that the framework can bring to future research are highlighted. To conclude the chapter, some proposals for future work are presented.

State of the Art

This chapter presents a review of the concepts and the state-of-the-art results on text mining tasks, namely document classification and topic modelling. The adopted approaches use machine learning and deep learning models and have shown significant results in this research area. Moreover, there are promising results using Twitter data. The last section explains how this dissertation can contribute to this topic.

2.1 Social Media Text Mining

In recent years, the quantities of available digital textual data have increased, generating new insights and new opportunities for research. Regarding big data analytic techniques, text mining has gained significant attention across a broad range of applications [12].

Text mining can be defined as the process of deriving high-quality information from text in order to extract useful information [13]. In other words, it is the process of transforming and substituting unstructured data into structured data to discover knowledge [14]. It uses [Natural Language Processing \(NLP\)](#), allowing machines to understand the human language and process it automatically [15].

The text mining process can be divided into four phases: data gathering, data preprocessing, content analysis, and integration of the text mining findings and results into the study. In order to analyse a simplified overview of the process underlying a typical text mining study, a diagram of this process is presented in [Figure 1](#). Data gathering consists of collecting data from databases and archives or scraping data from websites or social media. Then, preprocessing converts documents into a representation suitable for the classification task [14]. The Content Analysis phase uses algorithms to classify texts or cluster them into homogeneous groups. This dissertation will focus on the steps of data preprocessing and content analysis. These two activities might be executed in iterations to improve results by refining text preprocessing (as

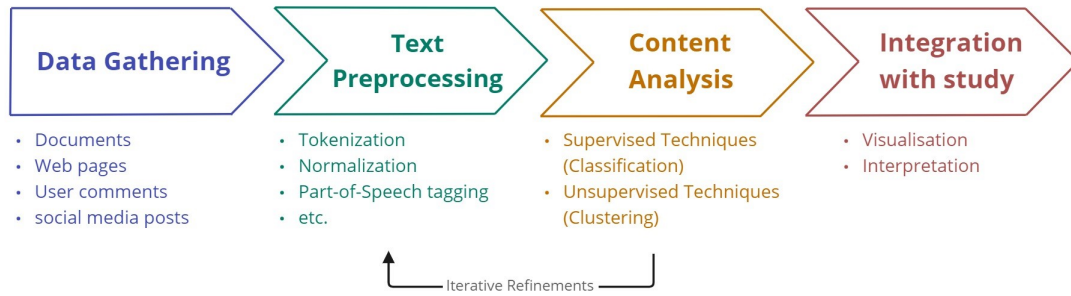


Figure 1: The text mining process. Adapted from [16]

shown in Figure 1). The following subsections will provide an overview of prevalent techniques used for each phase.

Text Mining tasks can include text categorization, text clustering, concept/entity extraction, production of granular taxonomies, sentiment analysis, document summarization, and entity relation modelling (i.e., learning relations between named entities) [17]. Regarding these techniques, the main focus of this dissertation will be Document Classification (explained in subsection 2.1.5), combined with Topic Modelling.

Regarding the use of social media for these tasks, recent research has used mainly Twitter, followed by other platforms such as Facebook [18] or WhatsApp [19]. Social media contains a massive volume of unstructured data (e.g. tweets, comments, blogs, forum discussions, user posts, and reviews) that can be used for business intelligence such as customer profiling and content analytics [20]. Most of these researches use English data, followed by Arabic, Chinese, Japanese and Persian [21].

In the next subsections, the different phases of the text mining process will be analysed, then Document Classification and Topic Modelling tasks will be defined. Previous efforts on each one of them will be explained, and to conclude, the case study of this dissertation will be analysed: Twitter Portuguese data.

2.1.1 Data extraction

The first step of the text mining process is data extraction. Regarding sources of data, there are usually two situations: open and closed domain data. Typically, closed data need to be supplemented with data obtained from public websites (including Wikipedia), textbooks, and professional literature. However, data from public networks (especially social networks) contain more noise and ill-formed expressions, taking more time to clean and preprocess [22].

Many datasets have been used for a wide variety of text mining tasks. A summary of those datasets can be analysed in [23]. Recently, many datasets for studying COVID-19 related data have been developed [24]. For example, the study in [25] contains five datasets related to Covid-19 and vaccines, three of which are publicly available. CORD-19, the COVID-19 Open Research Dataset, is the most popular open literature dataset containing 128000 papers regarding this topic [26]. For toxic speech detection, an overview of

Number of Tweets	Number of Classes	Topic	Article
10000	2 (Relevant/Irrelevant)	Disasters	[30] (2014)
34563	2 (Positive (Relevant)/Negative (Irrelevant))	Alcohol Use	[31] (2014)
10876	2 (Relevant/Irrelevant)	Disasters on social media	¹ (2016)
3785	2 (Relevant/Irrelevant)	Disasters	[29] (2017)
2311	2 (Relevant/Irrelevant)	Emergencies and Disaster	[32] (2018) (CrisisLex ²)
3200000	3 (Completely-relevant/Somewhat-relevant/Non-relevant)	News	[33] (2018)
21000	4 (Threat/Business/Irrelevant/ Don't Know)	Cibersecurity	[34] (2018) ³
10592	2 (Relevant/Irrelevant)	Influenza Prediction/Detection	[35] (2018)
3275	2 (Relevant/Not-Relevant)	European Migration Crisis 2015	[36] (2019) (MMoveT15)
3500	2 (Relevant/Irrelevant)	Symptoms of Syndrome of Choice	[37] (2019)
715894	2 (Relevant/Irrelevant)	Israel-Palestinian Conflict	[38] (2021)

Table 1: Twitter datasets for relevance classification

existing datasets is done in [27].

Regarding the use of social media and particularly Twitter for data extraction, many studies have collected data using the Twitter API [20], even though it is considered unreliable and incomplete [28]. Many of them have manually labelled their datasets [27], normally involving more than one annotator. For example, for obtaining relevant information during emergencies and disasters, Twitter datasets have been created [29]. Table 1 presents an overview of Twitter datasets created for different topics of research. All of these datasets are in English and were manually labelled.

In terms of extracting Tweets, many tools have been used. Besides Twitter API, Tweepy ⁴ and Twint ⁵ are popular alternatives, alongside with TweetScraper⁶ and DeepScraper [39]. All these tools were developed to overcome some limitations of the Twitter API.

⁴<https://www.tweepy.org/>

⁵<https://github.com/twintproject/twint>

⁶<https://github.com/jonbakerfish/TweetScraper>

After data acquisition, it is usually necessary to further process the data. Many techniques must be further analysed.

2.1.2 Preprocessing

The next step of a text mining pipeline is preprocessing. For many text mining tasks, data preprocessing is required. Doing this process on natural language data by computers is challenging and requires a number of sequential tasks to be implemented. The preprocessing steps implemented in many studies contain normalization, noise removal, tokenization, lowercase conversion, elimination of missing and misspelled words, stop-word removal, and lemmatization [40].

Normalization refers to a series of tasks such as converting all letters to lower or upper case, converting numbers into words or removing them and removing punctuation. In particular, stop-word removal, stemming, and lemmatization are critical processes in text normalization. Words of high frequency, such as "I", "the", "of", "my", "it", "to", and "from", which do not contain topical information, are called stop words. They are usually removed in text analysis to improve the algorithm's performance by reducing irrelevant words in vector spaces [41]. Lemmatization is the process of getting the inflexion of words, which are not just chopped off, but lexical knowledge is used to transform a word into its base form. There are many libraries available which help achieve lemmatization such as [Natural Language Toolkit \(NLTK\)](#), [gensim](#)⁷, [Stanford CoreNLP](#)⁸, [spaCy](#)⁹ and [TextBlob](#)¹⁰ [42].

A suitable preprocessing of informal texts can improve the classifier's predictive performance [43]. In 2021, Naseem *et al.* [42] studied the effect of twelve different pre-processing techniques for tweet classification using three different labelled datasets for Twitter hate speech and abusive language. They concluded that a specific combination of pre-processing techniques can lead to better classification results. This sequence consisted of using the twelve techniques in this order: remove URLs, user mentions and hashtag symbols; replace emoticons; replace abbreviations and slang; correct spelling; expand contractions; elongate character removal; remove punctuation; lower-case of words; segment words; remove numbers; remove stop-words and lemmatize words.

Before performing a feature extraction phase, researchers typically use four to five popular data preprocessing techniques for short text, like tweets [42]. When referring specifically to the extraction of tweets, the common approaches do not have any "scientific" justification for the pipeline that they are using (doing just what are common practices as resumed in Section 2.1.2). So, for this investigation, the preprocessing pipeline was defined according to the best results in the State of the Art, specifically the method proposed in [42].

⁷<https://radimrehurek.com/gensim/>

⁸<https://stanfordnlp.github.io/CoreNLP/>

⁹<https://github.com/explosion/spaCy/>

¹⁰<https://textblob.readthedocs.io/en/dev/>

2.1.3 Feature Extraction

After cleaning and normalising the initial text, it is necessary to transform its features to be used for modelling [44]. Feature Extraction aims to create new features from the initial ones and posteriorly reduces the number of features. Different feature extraction schemes are used by various researchers in text categorisation tasks. Term frequency (TF), [Term Frequency-Inverse Document Frequency \(TF-IDF\)](#), N-gram and other word embedding models [21] (which will be discussed in the subsequent sections) are some of the feature extraction approaches employed by various researchers in text categorisation problems.

[TF-IDF](#) is a feature extraction technique that measures a term's importance (weight) on a document regarding to a collection of documents. This weight increases when the term is more frequent in the text, but decreases when it is very frequent among the set of documents. For example, frequent words like "the" or "for" will have a low weight [45]. Another method is the [Bag of Words \(BoW\)](#), which tells us about the relationship between a document and the terms. However, this approach is usually worse than [TF-IDF](#) as it treats every word equally, and does not consider the importance of having words more frequent than others [44].

2.1.4 Embeddings

Word Embeddings are word representation vectors that attribute similar representation to words with similar meaning. This method is not restricted to words and has been applied to sentence and document level [46]. These text representations have been widely adopted, leveraging approaches like [GloVe](#) [24] or [fastText](#) [47]. Results prove the importance of word embeddings as a default feature extractor compared to the [BoW](#) [48]. The most used word embeddings are [word2vec](#), [fasttext](#) and [GLoVe](#) [49].

2.1.4.1 GloVe

The [GloVe](#) model (Global Vectors for Word Representation) is an unsupervised learning algorithm to obtain vector representations of words. Word embeddings using this method use only contextual information of the words (i.e. by computing the co-occurrence matrix) and ignore its morphology [50]. The co-occurrence matrix consists of entries of co-occurrence weight for each pair of words. The more times two words appear together, the higher the weight.

2.1.4.2 Word2Vec

[Word2vec](#) is a popular sequence embedding method that transforms natural language into distributed vector representations. It can capture contextual word-to-word relationships in a multidimensional space and has been widely used as a preliminary step for predictive models in semantic and information retrieval tasks. [Figure 2](#) describes the [Word2vec](#) process, which involves two distinct components: [Continuous Bag](#)

of Words (CBOW) and skip-gram. The CBOW component infers the target word when given the context words, while the skip-gram component infers the context words when given an input word [51].

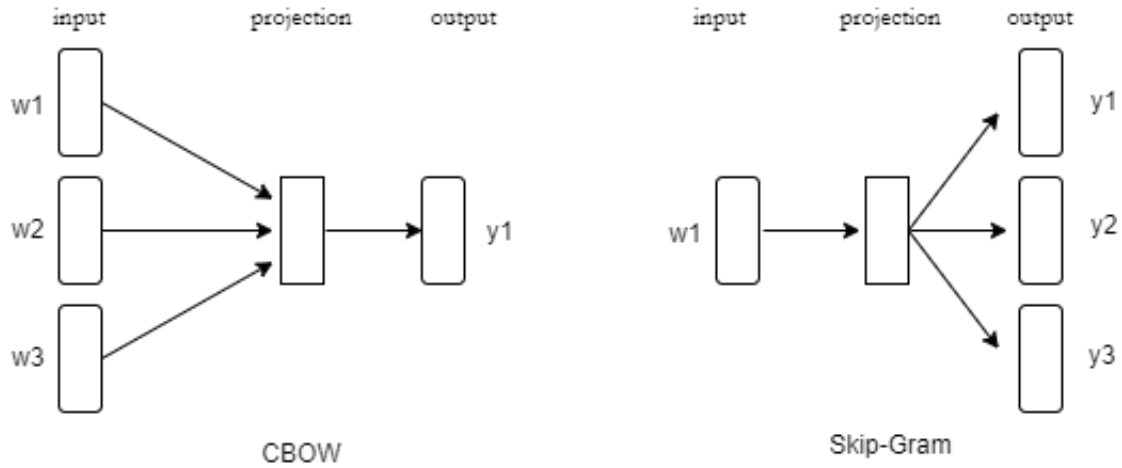


Figure 2: Word2vec model (Continuous Bag of Words (CBOW) and Skip-Ngram). Adapted from [51]

2.1.5 Document Classification

Document classification (also referred as Text Classification or Text Categorization) is the process of categorising documents in certain categories [7]. Document classification can be used for predicting binary or nominal class labels [20]. It is important to refer that this task is different from Document Clustering. Classification is a supervised learning technique used to assign pre-defined tags to a text, on the basis of features, thus requiring training data. Alternatively, clustering is an unsupervised technique that does not require training data and is used to group similar instances based on its features [14].

A typical document classification system can be deconstructed into four phases: Feature extraction, dimension reduction, classifier selection, and evaluation [52]. Many document classification methods are proposed by utilizing pre-trained word embeddings [53] (explained in section 2.1.4).

Dhar et. al (2021) [21] explored a wide variety of algorithms used for categorising text documents into three basic fields: conventional methods (1), fuzzy logic-based methods (2) and deep learning-based methods (3). The first and the third ones will be explained in the subsequent sections. Focusing on Twitter, many researches using the three different approaches were found: regarding the first approach, [31] classify tweets about alcohol use; [54] use the second approach to classify Hurricane Sandy data (which occurred in 2012); the third approach is used, for example, to detect traffic accidents [55].

Document classification has many applications, such as spam filtering, email categorization, sentiment analysis, etc [56]. Detecting fake news is also a text classification problem [24].

2.1.6 Topic Modelling

Topic Modelling is an unsupervised machine-learning technique that identifies key topics within a set of free-text data, based on statistical probability and correlations among words. It does not require manual labour to classify the free-text data and is well-suited to analyse large volumes of free-text data [57]. Topic models clarify the themes expressed in a collection of documents and elucidate the documents according to those themes [58].

Regarding applying this method to Twitter, some relations were defined between tweets in order to improve the process. These relations are based on hashtags, user mentions, retweets and meaningful words [59]. Topic modelling discloses the hidden semantic structure of tweets. LDA has been widely used for topic modelling in recent studies [60] [61]. This latest study introduces the first supervised LDA model on multi-label document classification.

According to [60], the use of hashtags has the advantage of classifying tweets within a certain topic with some independence of the language. However, it may not be sufficient for detecting subtopics. There are different approaches to classify and cluster text by topics will be explained.

Other techniques have been employed to classify Twitter content like the *fuzzy fingerprints* approach. In 2017, [62] used the called *Twitter Topic Fuzzy Fingerprints* to classify tweets from 2011 London Riots. This method allowed to expand the amount of tweets considered relevant to the topic by 45%. The fingerprints identify the identity of the given classes and the probability of two classes having the same fingerprint must be small. [63] also used fingerprints for topic detection since their attempts to apply LDA methods produced weak results, unless very extensive parameterization was done.

2.2 Machine Learning

There are two main kinds of machine learning algorithms, supervised learning and unsupervised learning; the former is used for classification and regression problems, while the latter is used for instance for clustering, data visualisation or dimensionality reduction.

A supervised learning algorithm assumes that all training examples are labelled with their output values (called target labels). The supervised learning refers to the process of optimizing the machine learning parameters to minimize the error. The unsupervised learning algorithm training examples have no labels, different from the ones in the supervised learning. [64]. A detailed review on many approaches regarding text mining tasks can be analysed in Table 1 on [23].

Numerous models have been proposed in the past few decades for text classification [23]. SVM and decision trees have been used on Document Classification [56]. Logistic Regression, a classification algorithm used to find the probability of event success and failure, has also been used but only Support Vector Machines (SVM), k-Nearest Neighbors (k-NN) and Naïve Bayes (NB) will be explained. Among these three algorithms, SVM typically achieves the best results on text classification tasks [65]. However, each

one of these approaches has advantages and disadvantages, depending on the size of the dataset. A deeper comparison between these three algorithms can be analysed in [66].

2.2.1 Naïve Bayes

NB is a simple algorithm which assumes that features are independent of each other. When the number of features is large, or the correlation between features is significant, the performance of NB decreases [23]. As a simple algorithm, it is also less sensitive to missing data, leading to suboptimal performance.

NB is used, for example, to improve TF-IDF vectorization [67]. Regarding the study of COVID-19 data, Multinomial Naive Bayes (MNB) algorithm showed better results than other traditional ML algorithms.

2.2.2 Support Vector Machines

An SVM approach was explored by Khamar *et al.* in 2013, defining a two-class problem, where the boundary between classes is provided by a hyperplane splitting a high-dimensional space that results from applying a kernel function to the original features in the data. The SVM techniques can be trained using predefined categorized text [65]. It is sensitive to missing data which can impact the accuracy and utility of the resulting classifier.

2.2.3 k-Nearest Neighbors

The k-NN approach is based on the similarity learning approach which is applied for many data analytics and text classification approaches and domains [65]. It is commonly used due to its easier interpretation and low calculation time. The choice of the parameter k is very crucial in this algorithm. To measure a document relevancy to a given query is used the Euclidean distance between the query vector and the document vector [68]. k-NN with TF-IDF term weighting representation scheme performs well in several text classification algorithms [66].

2.3 Deep Learning

DL is a sub-field of machine learning based on artificial neural networks. It is essentially based on learning deep representations, i.e., learning multiple levels of representations and abstractions from data. The term “deep” refers to the number of hidden layers in the neural network. Deep learning models are trained using large sets of labelled data, learning the features right from the data without the requirement for manual feature extraction methods. There are some concerns while working with DL techniques such that its likelihood to perform better with large volumes of data. Also, it is computationally expensive since the training of a deep learning-based system demands high-cost hardware resources [21].

Deep learning models have achieved remarkable success [69] and surpassed the traditional machine learning approaches [70] in many text mining problems, namely in document classification tasks [71] [72]. A detailed review on deep learning approaches regarding text mining tasks can be analysed in [73] Table III. The basic architecture of a deep learning-based approach is depicted in Fig. 3.

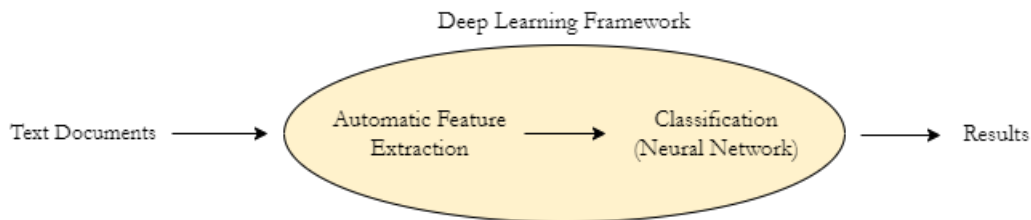


Figure 3: The basic architecture of deep learning-based approach. Adapted from [21]

2.3.1 Convolutional Neural Network

CNNs are neural networks that use mathematical operation called convolution instead of the common matrix multiplication in at least one of their layers. **CNNs** are most used in recognition and classification tasks [74]. These networks have been applied to Twitter data for hate speech classification tasks [75] [76]. A simple exemplification of this network applied to text classification can be analysed in Figure 4.

Many **CNN**-based models have been proposed regarding document classification tasks which are summarised in [70]. For example, Zhang *et al.* [77] examine the impact of different word embedding methods and pooling mechanisms, and find that using non-static word2vec and GloVe outperforms one-hot vectors, and that max-pooling consistently outperforms other pooling methods.

2.3.2 Recurrent Neural Network

While **CNNs** learn to recognize patterns across space, **RNN** is a network concerned with time and context [70]. Its core premise is finding relationships between an item (e.g. a word) and what comes before it and what goes after it. It is one of the most popular architectures, when it comes to NLP problems, because its structure is very fit to process variable lengths of text [49]. **RNN** are designed to work best with sequential data, or data that changes over time, such as textual or speech data. Figure 4 shows a **RNN** model for text classification with a simple sample. This model alongside **CNNs** is not intuitive enough having poor interpretability, especially in classification errors, which cannot be explained due to the non-readability of hidden data. To solve this problem, the attention mechanism (explained in subsection 2.3.4) can improve the interpretability for text classification [23].

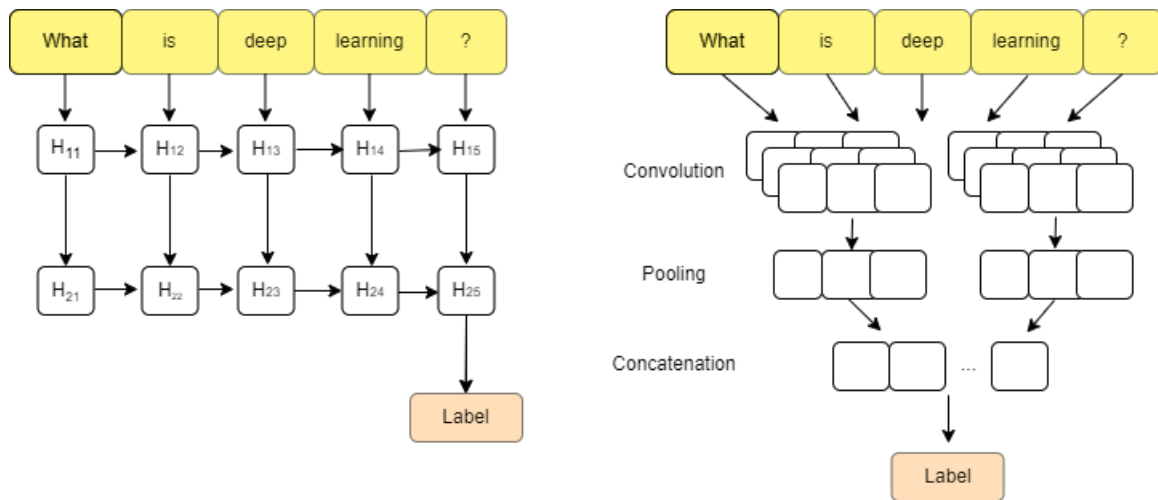


Figure 4: The RNN based model (left) and the CNN based model (right). Adapted from [23]

2.3.3 Long Short-Term Memory

RNNs can be composed of Long Short-Term Memory units to form an [Long Short-Term Memory \(LSTM\)](#) network. They are designed to handle classification and prediction problems on sequential data. The main difference between these networks and RNN is the ability of LSTM to remember information from layers that are too far behind, such as the case of sentences in a paragraph. LSTM networks have a forget gate, as well as an update gate. As the names suggest, the forget and update gates determine whether to pass the current information forward or to discard it [49]. Besides, LSTM and BiLSTM (referred in [51]) have been employed in text classifications tasks for a long time [78].

2.3.4 Attention Mechanism

The attention mechanism, in language models, refers to the part of a neural network that computes the weight of the input data, highlighting the most relevant elements. It was developed to solve the limitations of the encoder-decoder model related to long sentences. In order to know the relevance of a word in a sentence, the attention vector (referred previously) is used to check how that word “attends to” other words. The sum of the values weighted by the attention vector tells us the approximation of the word to the target [51][70]. More specifically, it takes two sentences, turns them into a matrix where the words of one sentence form the columns, and the words of another sentence form the rows, and then it makes matches, identifying relevant context [79]. Table 1 of [78] gives an overview of different models configurations using the attention mechanism.

Despite the fact that this method was originally applied to other algorithms, like RNNs and CNNs, it has proven to perform well on its own. A practical application of this mechanism is on Transformers, which use attention mechanisms to know the relevant context of a certain word and encode that context in the vector that represents the word [79]. An example of the state of the art transformers will be presented

in sections 2.3.6 and 2.3.7.

When using the attention mechanism with LSTM, the classification success is higher when it is trained with fewer data than when it is trained with more data. Jang et. al (2021) [51] proposed an attention-based Bi-LSTM+CNN hybrid model and compared the results to CNNs and LSTMs.

2.3.5 Hierarchical Attention Mechanism

Hierarchical Attention Network (HAN) is based on LSTM and comprises of four sequential levels - word encoder, word-level attention, sentence encoder, and sentence-level attention. Each data sample is divided into a maximum of 40 sentences and each sentence consists of a maximum of 50 words. The word encoder is a bidirectional LSTM that works on word embeddings of individual sentences to produce hidden representation for each word. The word-level attention helps us to extract important words that contribute to the meaning of the sentence. These informative words which conceive the complete meaning of the sentence are aggregated to form sentence vectors. The sentence vectors are processed by another bidirectional LSTM referred to as sentence encoder. The sentence-level attention layer measures the importance of each sentence and sentences which provide the most significant information for classification are summarized to get a document vector that contains the gist of the entire data sample [24].

2.3.6 BERT

Transformers were first introduced by Google in 2017 [80]. As referred previously, Transformer models can see all input tokens at once and, hence, can model dependencies in long sequences in constant time while enabling much higher parallelism. [81]. For a detailed explanation of Transformer architecture, it is recommended to read [80].

Fine-tuning of pre-trained transformer networks, such as BERT, yield state-of-the-art results for text classification tasks [82] [83]. **Bidirectional Encoder Representations from Transformers (BERT)** consists of a NLP model designed to pretrain deep bidirectional representations from unlabelled text and, after that, to be fine-tuned using labelled text for different NLP tasks [84]. The BERT model is bidirectional, which means that during the training phase, it learns information of token's context not only from the left side but also from the right one [85]. It outperforms **Embeddings from Language Models (ELMo)** and **Generative Pre-Trained Transformer (GTP)** [86]. BERT provides two architecture models: $BERT_{BASE}$ and $BERT_{LARGE}$. $BERT_{BASE}$ has 12 encoder layers, 768 hidden layers and 110 million parameters, whereas $BERT_{LARGE}$ has 24 encoder layers, 1024 hidden layers and 340 million parameters [87]. BERT's input representation takes into account the sum of the token and segment embeddings and also the position embeddings.

In terms of its applications, BERT is expected to affect 10% of Google search queries and has been applied to more than 70 languages [88]. Furthermore, many adaptations of BERT model have been

developed, regarding specific tasks. It has remarkable results in many studies [24], for example, for detecting fake news, achieving 98% of accuracy.

Nguyen et al. (2020) proposed BERTweet by pretraining BERT on a large set of English tweets. They made a comparison between BERTweet and other models [89]. On the other hand, DocBert is a specialised BERT model fine-tuned for document classification [88].

For the Portuguese Language domain, BERTimbau was created and it is publicly available ¹¹. This model was developed by training cgBERT models for Brazilian Portuguese [81]. These models were trained on two sizes: Base (12 layers, 768 hidden dimension, 12 attention heads, and 110M parameters) and Large (24 layers, 1024 hidden dimension, 16 attention heads and 330M parameters). Although it was trained for other NLP tasks (Named Entity Recognition (NER), Sentence Textual Similarity (STS) and Recognizing Textual Entailment (RTE)), it may eventually improve results on other NLP tasks, according to the authors [81]. In 2020, Müller et al. [25], released a transformer-based model, named CT-Bert, which was pretrained on a large corpus of Twitter messages on the topic of COVID-19. Although Bertimbau is trained for 3 specific tasks, it is applied to text classification so we want to check how it behaves here.

2.3.7 XLNet

XLNet outperforms BERT in some recent studies [90] [73] [91]. XLNet is a Permutation Language Model and takes bidirectional context into account, as BERT models do. To calculate what the next word is, it calculates the probability based on all permutations of word tokens in a sentence (considering forward and backward context). As opposed to the BERT model, that has a limit on the sequence length (up to 512 tokens), XLNet can handle large documents, as it has no token limit) [92]. XLNet gives the best result for many datasets in this study, in terms of accuracy [73].

Figure 2 compares the previous mentioned DL approaches, including XLNet.

¹¹<https://github.com/neuralmind-ai/portuguese-bert>

Deep Learning Architecture	Novelty Introduced	Feature Extraction	Corpus
HANN	Hierarchical structure	Word Embedding	Yelp, IMDB, Yahoo and Amz
XLNet	Autoregressive language model, Permutation operation	Word Embedding	IMDB, SST-2, Amz-2, Amz-5, Yelp
BLSTM-2DCNN	Two-dimensional max pooling with bi-directional LSTM	Word embedding	Yelp, IMDB, and Amz-2, Amz-5
ALBERT	Lowers memory consumption and increases the training speed of BERT	Word Embedding	SST-2

Table 2: Comparison of different deep learning techniques for document classification. Adapted from [73]

2.4 A case study: Twitter

Section 2.1 mentioned the importance of studying the information shared on social networks due to the daily growth of active users. Considering this, the main focus of this dissertation is a particular social network: Twitter. In this section, the previous detailed explanation of the text mining pipeline process will be applied to this particular case.

Twitter is a social media platform that emerged in 2006. In 2021, it has more than 211 million active users worldwide [93]. It has become a very popular tool to share news, ideas and comments on a wide range of topics and word events [5]. Using Twitter, one can quickly discover the most relevant discussions in the world by trends. On this platform, famous users, such as politicians and celebrities, post a variety of information and have millions of followers. Therefore, this social network plays an essential role in spreading people's thoughts and influencing people's opinions [94].

On Twitter, users share very-short posts, known as *tweets*, and each one of them is currently restricted to 280 characters [95]. Besides the fact that these posts are very short, they tend to have an imprecise and informal language [96], namely noisy vocabulary (slang, emoticons, grammar errors) [5]. Therefore, it is hard to find all relevant information regarding a certain topic on Twitter. Nevertheless, Twitter allows using *hashtags* on tweets which will be important to the development of this work. According to Twitter documentation¹², an *hashtag* is a combination of words initialised with # symbol used to index keywords or topics on Twitter. They appear as #word (e. g., #portugal) and represent topics. Hashtags allow people to easily follow topics they are interested in and most of the used tools to obtain Twitter trends are based on hashtags since the hashtagged tweets are associated with a topic.

However, not all tweets related to a given topic are hashtagged [5] and not always the content of the tweets is related to the hashtags used. The following example illustrates this last case: "*Resolvi começar*

¹²<https://help.twitter.com/pt/using-twitter/how-to-use-hashtags>

a me arriscar também com papel, canetas e tintas... #coronavirus #COVID19" (in English, "I decided to start risking myself also with paper, pens and inks..."). This tweet was hashtagged with "#covid", but the content is not related to the pandemic. According to The Twitter Engagement Report from 2018¹³, about 40% of tweets contain at least one hashtag, which means the other part is not hashtagged and does not have a topic related to the tweet.

Besides the hashtags, Twitter has a recent *context annotations* field for many tweets, that labels tweets according to a list of topics¹⁴ without using machine learning. This is a functionality of the new version of the Twitter API that attributes a topic to a tweet based on a semantic analysis of keywords, hashtags, handles. This classification attributes one or more topics to the tweet and each topic contains a domain object and an entity object. For example, the tweet *"The volume of conversation about COVID-19 is tremendous, which means it requires expertise and computational resources to process. Developers and researchers with that capability and intent to support the public good can apply for access."* is classified with four topics, which are detailed in Figure 5. The four detected topics are from these domains: Brand Category, Brands and Companies, Interests and Hobbies Vertical, and Interests and Hobbies Category.

The approach used by Twitter to attribute these annotations to tweets is similar to the process adopted in this work. However, these annotation are still barely available for Portuguese tweets.

Regarding the use of Twitter for scientific research, the number of Twitter publications has increased significantly since 2006, and this trend is expected to continue in the next years [95]. The research areas involving Twitter include health (for example, to detect *Influenza* epidemics [97]), politics or even the interest of citizens on the impact of marine plastic pollution [60].

Most of the researches involving Twitter use english data. However, other languages have been explored: chinese, using XLNet for classifying censored tweets [91], and arabic [49] [9]. Related to crime, this study [1] intends to classify tweets into crime or non-crime tweets which best result was given by Random Forest. Another recent study [94], presents a framework that can access old Twitter Data and intelligently detect the most relevant tweets on a given topic, even if those tweets do not contain the topic's hashtag. In 2020, the authors of [27] studied toxic detection and, in 2021, [87] achieved state-of-the-art results in rumours detection using Twitter. In terms of topic detection, a recent review over the different approaches using Twitter data has also been done [4].

2.4.1 Portuguese Twitter Data

Regarding not only but mainly Twitter data, a significant amount of data is generated in English, whereas the rest is derived from other world languages.. According to 2018 statistics [98], only 32% of all Twitter messages are written in English. So, by analysing only the English language, part of the raw data of interest is put aside. Therefore, performing analyses in other languages is necessary for extracting useful information from tweets [94]. Concerning the Portuguese language, the studies on this language

¹³<https://mention.com/en/reports/twitter/hashtags/>, accessed: 2022-09-06

¹⁴<https://github.com/twitterdev/twitter-context-annotations>

```
"context_annotations": [  
  {  
    "domain": {  
      "id": "46",  
      "name": "Brand Category",  
      "description": "Categories within Brand Verticals that narrow down the scope of Brands"  
    },  
    "entity": {  
      "id": "781974596752842752",  
      "name": "Services"  
    }  
  },  
  {  
    "domain": {  
      "id": "47",  
      "name": "Brand",  
      "description": "Brands and Companies"  
    },  
    "entity": {  
      "id": "10045225402",  
      "name": "Twitter"  
    }  
  },  
  {  
    "domain": {  
      "id": "65",  
      "name": "Interests and Hobbies Vertical",  
      "description": "Top level interests and hobbies groupings, like Food or Travel"  
    },  
    "entity": {  
      "id": "848920371311001600",  
      "name": "Technology",  
      "description": "Technology and computing"  
    }  
  },  
  {  
    "domain": {  
      "id": "66",  
      "name": "Interests and Hobbies Category",  
      "description": "A grouping of interests and hobbies entities, like Novelty Food or Destinations"  
    },  
    "entity": {  
      "id": "848921413196984320",  
      "name": "Computer programming",  
      "description": "Computer programming"  
    }  
  }  
]
```

Figure 5: Example of a context annotation object for a tweet classified with four different topics (fragment of a tweet payload from Twitter API v2)

are scarce [99], even though it is the sixth most spoken language (232 million native speakers) [100]. Few papers in the literature refer to the use of text mining in Portuguese texts [5]. To further aggravate this scenario, there are few resources such as public datasets with texts in Portuguese, and those used in researches related to this language are rarely available [43]. Nevertheless, some recent investigation has been done.

Souza et al. (2016) made a systematic mapping review of the Portuguese language on text mining tasks, and concluded that text classification (49%) appears as the main text mining task for the Portuguese language [99]. Some recent researches have contributed to the study of the Portuguese language in terms of topic detection on Twitter, including COVID-19 data [5] or political content [101]. In 2021 [102], an overview of data from Twitter users in Brazil was published, related to COVID-19, using *Word2Vec* for preprocessing. On the other hand, a Portuguese annotated dataset for hate speech detection has been developed using Twitter data [103]. In order to test the relevance of their developed annotated hate speech dataset, they deployed pre-trained GloVe word embeddings techniques for feature extraction from the dataset, alongside with a LSTM as the baseline binary classifier for the identification of hate or no-hate in a Portuguese hate speech annotated dataset [74].

Souza et al. [104] released Portuguese BERT models for future researches to benchmark and improved the performance of many NLP tasks in Portuguese. Regarding the impact of using different preprocessing techniques for text mining tasks using Brazilian Portuguese, [105] concluded that different approaches for preprocessing social media texts can improve the classifier predictions. According to [106], 95% of the Portuguese tweets are from Brazil.

It is clear that more investigation over this language is essential so this project is a contribution for the Portuguese-speaking community.

2.5 Related Work

In order to summarise the mentioned studies along this chapter, Table 3 contains an overview of the recent studies that used DL models for document classification tasks. Most of the mentioned studies use Twitter data and the adopted DL model is BERT or XLNet. Some of these datasets are publicly available as the case of ToLD-BR, a twitter dataset annotated according to different toxic aspects like *insult*, *racism* and *xenophobia*.

Regarding the development of frameworks to extract and analyse Twitter data, a complete framework for extracting and classifying tweets in Persian and English was developed [94]. Another recent research [107] proposes a supervised ML approach capable of performing information extraction and classification of emergency-related social media data covering any language. The models here were trained using English data but achieved acceptable performances via zero-shot learning on Spanish and Italian data.

Authors	Corpus	Approach	Metrics
Sharma et al. (2020) [92]	MS-MARCO	XLNet	Accuracy, F1, Precision, Recall
Silva et al. (2020) [27]	ToLD-Br (Toxic Language Dataset for Brazilian Portuguese Tweets) ¹⁵	BERT	Recall, Precision, F1
Garcia et al. (2020) [5]	Topic and Sentiment COVID-19 Twitter Dataset ¹⁶	BERT	Precision, F1
Ahmed et al. (2021) [91]	Censored Tweets	XLNet	F1, Precision, Recall, Accuracy
Kumar et al. (2021) [73]	IMDB, SST-2, Amz-2, Amz-5, Yelp	XLNET	Accuracy
Anggrainingsih et al. (2021) [87]	PHEME (rumour detection, Twitter)	BERT	Accuracy, Precision, F1, Recall
Wani et al. (2021) [24]	Contraint@AAAI 2021 Covid-19 Fake news detection (Twitter, Instagram, Facebook)	BERT	Accuracy (98%)

Table 3: Previous approaches using DL models regarding mainly Twitter data

Regarding other commercial and non-commercial frameworks related to tweets collection and analysis, an overview of existing frameworks is presented in Table 4. By analysing this table, it is visible that there are, at least, five frameworks that extract tweets and allow their visualisation. There is no open-source tool that satisfied the purposed of extraction of this work. The last one was the nearest to fill the requirements from OmniumAI (described in the next chapter) but still does not fill all.

Framework	Require consumer API Keys	Extract in Portuguese	Dynamically Extraction	Labelled Datasets
Social Bus ¹⁷	✓	✗	✗	✗
DMI-TCAT ¹⁸	✓	✗	✗	✗
SMF-Twitter Harvest ¹⁹	✓	✗	✗	✗
Twitter-Ratings ²⁰	✓	✗	✗	✗
Twitter-Watch ²¹	✓	✓	✓	✗
MISNIS [63]	✓	✓	✓	✗

Table 4: Comparison of different extracting frameworks for Twitter (commercial and non-commercial)

As it is described in [101], besides these tools, that are a few commercial tools like Hootsuite²², Sysomos²³ and Brandwatch²⁴ but these ones did not satisfy the need to store data as they do not store data for the client (only allows to analyse the dashboard of the extracted data). Both Twitter-Watch and MISNIS store the data in a way that the user can access. However, the seed of the former is based only on users and the last mentioned framework, MISNIS, developed in 2017, is not publicly available (only by requesting the authors) and does not attribute a label of relevance to the extracted tweets.

2.6 Discussion

A summary of the previous investigations and developed tools for extracting and classifying information from Twitter has been done. Regarding the text mining tasks that will be applied in this research, it is important not only to explore how much data is needed to classify a group of documents correctly per topic, but also to know if the explored approaches work well on short messages like tweets. When it comes to the pre-processing stage, some recent surveys give limited insights into the understanding of the appropriate selection and application of pre-processing on Twitter data [42]. However, *D Cerqueira et al.* [108] concluded that there is no complete proposals or tools for preprocessing social media data for Portuguese. Besides that, when it comes to existing datasets, there are many labelled datasets for sentiment analysis but there are barely no available datasets for Portuguese classification regarding tweets. The few that exist may not related to the topics the researcher might want.

Deep Learning achieves the best results over document classification and Topic Modelling. It is clear that many of the studies only make use of English data. Therefore, it is important to apply these methods, that have promising results regarding the English language, to other languages of interest in order to understand if the applied methods work for this one. The main focus of this dissertation will be the Portuguese language.

In terms of the state-of-the-art models applied to text classification and clustering, XLNet has shown the best results, surpassing BERT. This DL model brings promising results compared to ML models. When it comes to small datasets, traditional models tend to present better results than deep learning models, regarding the limitation of computational complexity. Therefore, some researches study the adaptation of traditional models for specific domains with fewer data [23].

To conclude, it is relevant to apply text mining and classification to Twitter Portuguese data, creating a system capable of extracting and learning which messages are relevant to a given topic, having different practical applications. Before starting the explanation of the development methodology in the next chapter, it is important to define relevance. According to [109], relevance is defined as *"a document meeting an information need that prompted a query"*. In our use case, the *query* is a chosen topic and relevance is the semantically similarity to the pretended topic. This means that if a tweets contains words that are related

²²<https://www.hootsuite.com>

²³<https://www.meltwater.com/en/sysomos-meltwater>

²⁴<https://www.brandwatch.com/>

to the topic they are classified as *relevant*. When "relevance" is mentioned alongside the document, it is related to this meaning.

Development Methodology

This chapter details the methodology designed to approach the problem described before. As the title of this dissertation suggests, this work is divided into two sub-problems: extraction and classification. For both, a high level strategy will be detailed regarding the requirements of OmniumAI. Then, the requirements for the final framework will be described. This framework will integrate the solution developed for these two problems, encompassing an [User Interface \(UI\)](#) that allows the user to extract data and draw conclusions upon that data.

3.1 General Approach

First of all, the strategy to approach the problem had to be defined. As stated in the previous chapter, there is no other existing tool that allows Twitter data extraction and create labelled datasets for a wide variety of topics. Besides that, the available labelled datasets to perform text mining tasks do not satisfy the requirements of the study that will use those datasets. Therefore, the presented solution is based on the text mining process for social media documents and was designed in agreement with OmniumAI.

Having in mind the four phases of the text mining process explained in Section 2.1, Figure 6 illustrates what the framework created in this work will cover at a high-level. First, an extractor, named `Twitter Extractor`, will be developed for the initial phase of the process (data gathering). To do so, an extraction pipeline is defined based on the state-of-the-results and approaches. Then, a classification pipeline is developed to classify the extracted tweets according to its relevance to the research topic with a semi-supervised approach. This process encompasses the second and third phases, as it includes also the text preprocessing tasks. Finally, the `Twitter Observatory` framework is created to integrate both pipelines and, consequently, the three initial phases of the text mining process are covered. The final step depicted in the figure, *Integration with study*, will address use cases of the framework, materialised in this

work by the two cases presented in Chapter 5.

So, to conclude, this can be divided into 3 main stages: the extraction pipeline, the classification pipeline and the development of the final framework. It should be noted that the pipelines presented in this Chapter may be later refined taking into account the objectives of OmniumAI, the company in which this project is inserted.

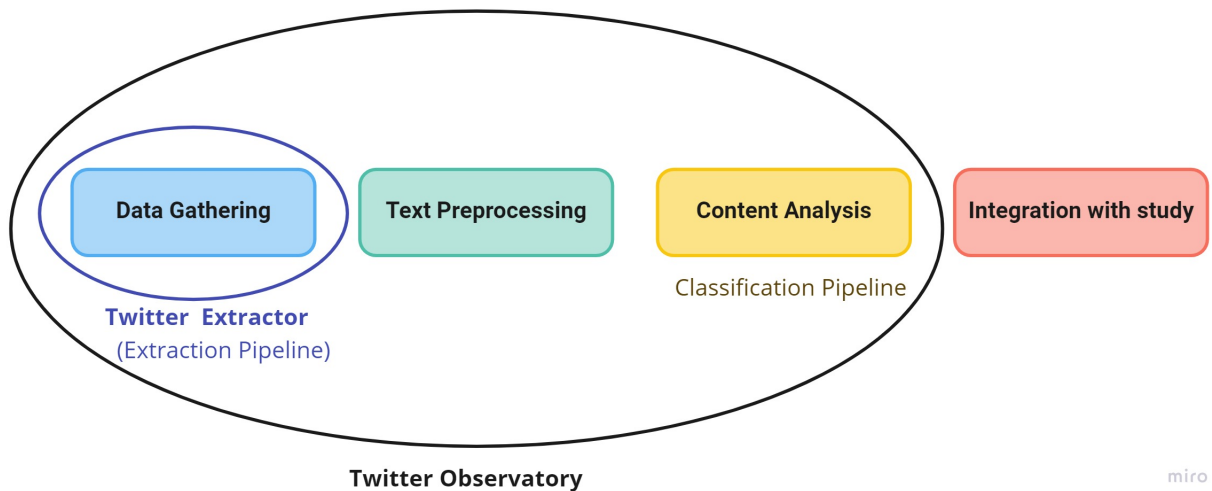


Figure 6: General methodology approach to the identified problem

3.2 Extraction Pipeline

Figure 7 depicts the extraction pipeline to obtain datasets regarding a certain topic. Essentially, an extraction tool is developed, named `Twitter Extractor`, which consists of a wrapper for the Twitter API. This extractor collects data from Twitter according to an initial seed composed of three components of Twitter: hashtags, keywords and users. The hashtag definition was already given in Section 2.4. A keyword is a word, a phrase or a group of words, and a user is a username string, typically a maximum of 15 characters long. Using this seed and the Twitter API, the tweets are extracted with this tool and a first set of tweets is obtained. As one of the main problems of this dataset is attributing a label, for the datasets to be later used for classification models, the labelling approach here is studied based on what was found on the literature. It is important to refer again the definition of relevance adopted in this work and mentioned in Section 2.6: *relevance* is defined as “a document meeting an information need that prompted a query”. In our use case, relevance is the semantically similarity to the pretended information. When “relevance” is mentioned alongside the document, it is related to this meaning.

Then, to refine the extraction process, Topic Modelling is used to find more hashtags and keywords that should be part of the seed, and the extractor collects more data with the new seed. The initial dataset is augmented with the new data and the process repeats, as long as there are more data to extract. This

should be an iterative and continuous process. After this refinement, the final dataset, containing data from all the iterations, is stored in the database.

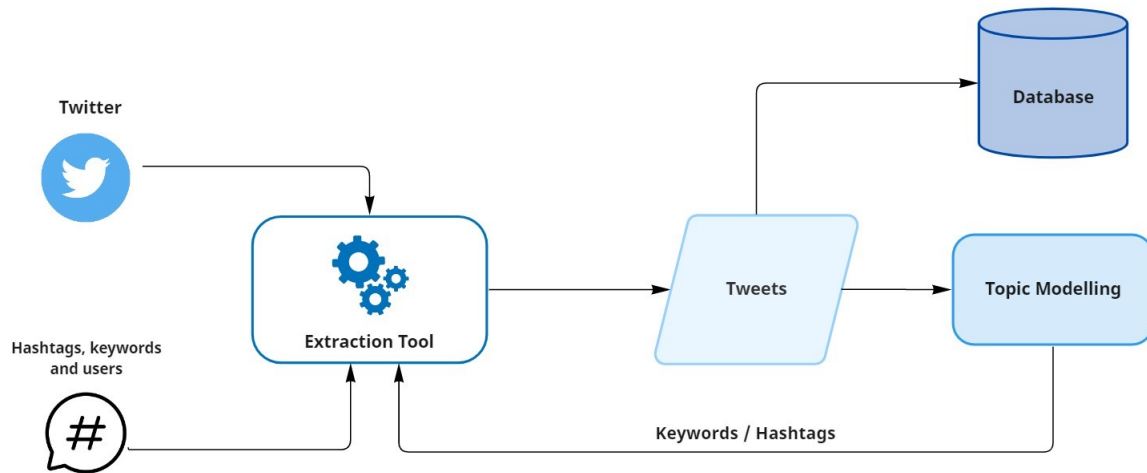


Figure 7: Extraction Pipeline to create a dataset according to a certain topic

3.3 Classification Pipeline

The second task consists of developing a classification pipeline for training and validating ML and DL models, which have been used to obtain state-of-the-art results to classify the tweets collected with the previous pipeline, according to their relevance.

As can be seen in Figure 8, a semi-supervised approach will be adopted to classify the tweets according to their relevance. First, the data obtained in the extraction process is read from the databases. Then, the preprocessing functions are applied, according to the models that will be used. Next, the models found in the state-of-the-art review are applied to the data to obtain a classified dataset according to relevancy, giving a binary label - relevant (1) or irrelevant (0). Finally, the user validates the labels and the verified data is sent to the models to train over the new data and improve the classification results. This is an iterative process. It is worth notice that the *topic dataset*, mentioned in the figure, will be obtained from the extractor and already contain automatic labels, which will be validated in the classification process define in this Section.

It is not a requirement to use this classification pipeline that the input dataset be built with the extractor, the final user may also add existing datasets or create them with other tools.

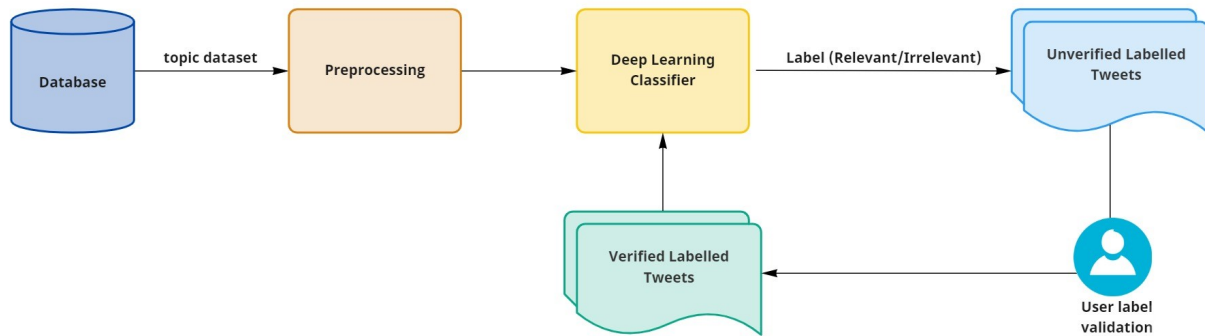


Figure 8: Classification Pipeline for tweets according to their relevancy

3.4 The Final Framework

The two previous pipelines will be integrated in a final framework named *Twitter Observatory*. This framework is also explained in detail in Chapter 4. It will include the three phases of the twitter mining process mentioned in Section 2.1, and will consist of an *UI* for the user interaction and contribution to the classification, as well as visualisation of the stored data. At the end, it will be integrated in the OmniumAI Software and will also incorporate the work developed by Jorge Gonçalves on his master thesis entitled *"Development of tools for sentiment analysis in the Portuguese Language"*, a module for sentiment analysis classification. The *UI* is designed to allow users to validate the dataset labels, add data to the datasets, extract data, apply preprocessing functions and models, and visualise the stored data. Besides that, must contain also a exploratory data analysis module to allow the analysation of the data extracted.

Summary

This chapter gives an overview of the methodology conducted in this project, detailing the global functions of the final framework. This is the high-level proposed solution to the problem identified, which is divided in three main parts: extraction, classification and interaction with the user, via the *UI*. Each one of these parts has functional requirements, defined by OmniumAI, which will be explained in the next Chapter, starting with the Twitter Extractor.

Implementation

This chapter details the implementation of the `Twitter Observatory`, including the `Twitter Extractor` and the classification pipeline. Firstly, the `Twitter Extractor` is explained having in mind the requirements needed for the extraction. Then, the classification modules are defined, using pre-trained models and the pipelines of `OmniumAI`. Finally, the `UI` is presented.

4.1 Twitter Extractor

In order to have enough data for the analysis, it was necessary to start extracting tweets according to some requirements. Since there was no tool yet that satisfied the purposes of this work, as explained in Section 2.5, it was necessary to develop our own extraction tool.

4.1.1 Functional Requirements

The functional requirements for the extraction tool that were established in the beginning of this project, agreed with `OmniumAI`, were:

1. Extract tweets from a specific period of time (recent or from years ago), without any limitations;
2. The initial seed defined for the extraction must be a combination of hashtags, keywords and users;
3. Be able to capture tweets according to a specific language (in this work, Portuguese tweets from Portugal or from Brazil, depending on the search topic);
4. Combine mandatory and optional seed parameters (keywords, hashtags and users) in an easy way and be user-friendly¹ (to facilitate the access by people outside the software engineering area);

¹<https://techterms.com/definition/user-friendly>

5. Surpass Twitter API limitations (detailed below in Section 4.1.2);
6. Save data in a database that facilitates the subsequent analysis;
7. Collect initially a set of relevant tweets and then dynamically find new potentially relevant ones.

Although the requirements mentioned in Section 4.1.1 are similar to Twitter-Watch requirements (the tool mentioned in Section 2.5), it still does not satisfy requirement 4 as the initial seed of this framework only has users (and not hashtags or keywords). When the extraction for a topic is started, the user that defines the initial seed may not know the list of users that tweet relevant content about the topic. Instead, it is possible that he might only know the keywords or the hashtags that are related to the topic so it is important that the hashtags and keywords compose the initial seed for the extraction.

Since this tool is for commercial use and must adhere to the law, the Twitter Developer Terms² were consulted before the entire process of developing the extractor, so we could be clear about what was permitted and what was not. Therefore, it was made clear by reading this documentation that only the Tweets ID and not the whole text of the tweet can be sent when redistributing Twitter material (extracted by an entity via the Twitter API).

4.1.2 Twitter API Limitations

In order to understand the potential accesses the developers could have to the extract data, the initial stage in the tool development process was to analyse the Twitter API documentation. *Twitter API v2* is the most recent version of the API, having three access levels which are summarised in Table 5. Both *Elevated* and *Academic Research* require submitting a request to Twitter to have the credentials to access the *Application Programming Interface (API)*.

Access	Essential	Elevated	Academic Research
Tweets Caps per month	500k	2M	10M
Search Tweets query length	512 characters	512 characters	1024 characters
Access to full-archive search Tweets	X	X	✓
Range limit to retrieve tweets	last 7 days	last month	since 2006

Table 5: Twitter API access levels

²<https://developer.twitter.com/en/developer-terms/more-on-restricted-use-cases>

At the beginning of this tool development, a request for the elevated access was submitted, but no response was given. Besides, since the Academic Research is only for research purposes which is not our case, we decided to surpass this limitation by finding a tool that did not required the API access credentials. For that, the next step was to find relevant wrappers written in Python (the chosen language for the development of this tool as a requirement of OmniumAI) for the Twitter API.

Regarding the Python wrappers for the Twitter API, there are quite a few to mention, although many of them require API access so they were not a viable option for this project. As detailed in the requirements list, this project needed tweets older than 30 days, specifically from years ago as well real time extraction, taking much profit from the API as it was needed. Table 6 summarises the major benefits and disadvantages of each wrapper found.

Package	Require API Credentials	Major benefit	Disadvantage
Tweepy ³	✓	make complex queries	only seven days tweets
Python-Twitter ⁴	✓	extract tweets by ID	request limits
GetOldTweets3 ⁵	✗	get old tweets	poor documentation
TwitterScraper ⁶	✗	surpass API limitations	poor documentation
Twint ⁷	✗	surpass API limitations	extraction is slower than Twitter API

Table 6: Python wrappers for the Twitter API

All the mentioned wrappers in Table 6 did not satisfy the purpose except Twint. Therefore, the extractor was developed based on Twint. This tool extracts tweets according to search parameters in a similar way to what is done in Twitter application (web or mobile) when the users search for tweets. It collects tweets in a similar way of Twitter API, but it surpasses its limitations. Utilising Twitter's search operators, Twint allows scraping Tweets with specific hashtags and keywords alongside with taking out sensitive information, such as e-mail addresses from Tweets. This tool is under the MIT license⁹.

As the current version of Twint has some bugs and limitations, it was necessary to use a more recent version of this tool (a fork of the original repository)¹⁰ The main problem consisted in the fact that Twint did not extract tweets before a certain period of time and stopped running with a message saying there

⁹<https://choosealicense.com/licenses/mit/>

¹⁰<https://github.com/woluxwolu/twint>.

was no more data, while in fact there was. Due to lack of documentation in some parts, it was difficult to find out these problems. In order to surpass these limitations, besides the fact we used the more recent version of Twint, the tweets were extracted day by day, creating a new "Search" instance for each day of the required period of time.

The search is made using the query operators¹¹. It is worth noticing that the Search operators do not rely on *completeness*, but rather on *relevance*, meaning that it does not make an exhaustive search for all Tweets, but simply for the more relevant ones. According to what was possible to find in Twitter documentation, *relevance is based on the popularity of a Tweet (e.g., when a lot of people are interacting with or sharing via Retweets and replies), the keywords it contains, and many other factors*¹², but it does not specify numbers. It does not collect private tweets, which can only be extracted by the followers of the user's tweet, according to Twitter documentation¹³, so there will always be a part of tweets that will not be collected.

Besides these restrictions, it is worth noticing that it is only possible to search tweets since March 2006, when the first tweets were released, so this is the "since" limit in the search parameters defined in this tool. It is documented that when streaming real time, it only collects 1% of the tweets¹⁴.

4.1.3 Twitter Extractor Implementation

Regarding the implementation of the extractor, Figure 9 gives a top-level view of the software's architecture. The Twitter extractor depicted in the figure is a Flask API, implemented in Python, with a connection to MongoDB to save data. The API fetches the tweets from Twint, using the search operators defined by Twitter, as mentioned earlier. The software that interacts with this API has to give the extraction parameters as input, which are the initial seed and other configurations like language, location and date.

The Flask framework¹⁵ is a lightweight framework in Python that allows to implement API in a clean and easy way. When comparing with other Python frameworks, like Django or FastAPI, Flask is customisable and modularised, which is a major benefit for an application that is supposed to increase in the future. MongoDB was chosen due to its availability and scalability. One of the major benefits of using MongoDB is that it ensures no duplicate documents exist, combining this with Twitter object model's `_id` field, eliminating redundancy. The developed API is properly documented using Swagger¹⁶ for the API and the NumPy Docstring format¹⁷ for the Flask code.

Taking advantage of Flask's modularization, the extractor source code is divided into several modules: database, containing all the functions regarding the interaction with the database; exceptions, to define

¹¹<https://developer.twitter.com/en/docs/twitter-api/tweets/search/integrate/build-a-query#list>

¹²<https://help.twitter.com/en/using-twitter/top-search-results-faqs>

¹³<https://help.twitter.com/en/safety-and-security/public-and-protected-tweets>

¹⁴<https://developer.twitter.com/en/docs/twitter-api/tweets/volume-streams/introduction>

¹⁵<https://flask.palletsprojects.com/en/2.2.x/>

¹⁶<https://swagger.io/>

¹⁷<https://numpydoc.readthedocs.io/en/latest/format.html>

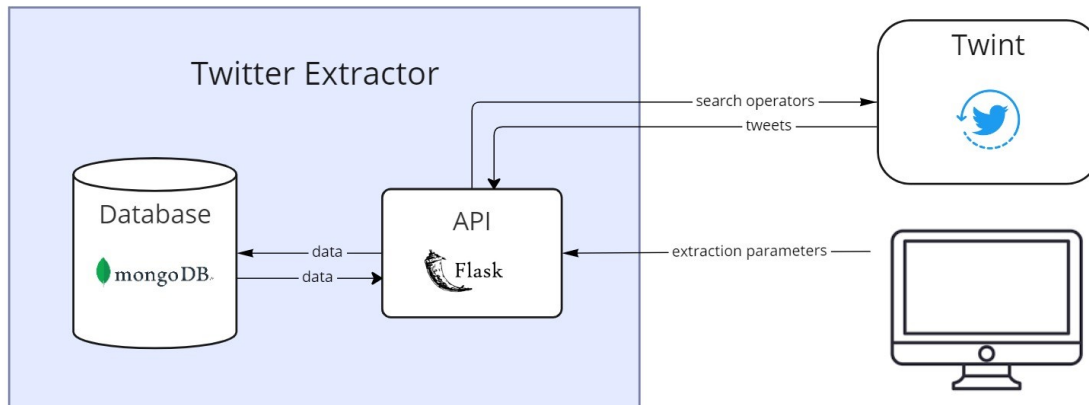


Figure 9: Twitter Extractor architecture diagram

the exceptions regarding the functions defined; the [Exploratory Data Analysis \(EDA\)](#), that contains the functions to the data analysis (detailed in Section 4.1.4).

To collect relevant information, a seed from where to start extracting tweets had to be defined. It was decided that the initial seed for the extraction was composed of **hashtags**, **keywords** and **users**. Each of these components is explained in Section 4.1.2. The main purpose of defining a composed seed with these three components is to explore the largest number of tweets to get relevant data for posterior analysis.

The main endpoints for this framework are *tweets/from-seed*, *tweets/from-user* and *stats*. The most important is *tweets-seed* which extracts tweets according to an initial seed of hashtags, keywords and users. This endpoint triggers a script that runs daily in order to collect new tweets according to the search parameters (that are dynamically adapted according to results of Topic modelling performance, explored in Chapter 5). Figure 10 summarises graphically the developed endpoints, which are explained below.

4.1.3.1 Tweets-seed

This is the most important endpoint as it is responsible for collecting tweets according to the search seed, which includes keywords, hashtags and users. Regarding the input parameters for this endpoint, each one of the seed is divided into optional and mandatory data which means that the seed is composed of 6 lists: mandatory hashtags, optional hashtags, mandatory keywords, optional keywords, mandatory users and optional users. It is not mandatory that all these parameters are filled in at the beginning of the extraction. The decision to divide every seed parameter into mandatory and optional was based on the fact that later, according to the algorithm to increase the dataset, the optional parameters can become mandatory (this will be detailed in Section 4.1.5). In practice, this division allows to search for tweets that contain, for example, *#hashtag1* OR *#hashtag2* or both, which is very useful at the beginning of the extraction, as one may not know which combinations are mandatory and which are optional. One may

Tweets		Stats		Databases	
Represents tweets that are extracted		Represents the statistical data about the extraction for a database		Represents the databases containing data for a specific topic	
GET	• tweets/<db_name> - Return tweets from the database with db_name	GET	• stats - Return statistical data about tweets extraction	GET	• databases - Return stored databases names
POST	• tweets/from-seed - Return extracted tweet according to the seed				
POST	• tweets/from-user - Return tweets from a specific user				

Figure 10: Twitter Extractor API - Endpoints Diagram

infer that for COVID-19, tweets that contain the word *"pandemia"* must also contain the word *"covid"* for the tweet to be relevant, but it is not necessarily like that.

As stated above, a keyword is considered one of three things: a word, a phrase or a group of words. As so, it had to be well defined in the seed to distinguish the three types. For example, searching for *"guerra da Ucrânia"* (in english, *"Ukrain War"*), is considered a group of words and therefore it will return all tweets that have the word *guerra* AND the word *da* AND the word *Ucrânia*, not necessarily in this order. By searching for *"guerra OR da OR Ucrânia"*, each word is considered an isolated word so it returns tweets containing one of the three words or two of these words or even these three words. Finally, searching by *"guerra da Ucrânia"*, will return tweets containing this exact phrase.

The other available parameters are: *language*, *since date*, *until date* and *location* (optional parameter). The since and until parameters are in format *YYYY-MM-DD*. "Since" refers to the initial day from where to extract data. For example, if since is *"2020-02-02"*, the extractor will extract tweets that were posted since that date, inclusively. "Until" refers to the last day to extract data. As so, if until is *"2021-01-04"*, the extractor will collect tweets that were written until that day (including). If no data is specified, it will extract tweets from the day the endpoint is being called. The *lang* parameter is a two character code indicating the language in which we want to extract tweets, among the available language codes¹⁸. For Portuguese tweets, the language code is *"pt"*. The Twint language filter does not work perfectly because there are tweets too small to be able to detect the language. Therefore, some extracted tweets may be in Italian or Spanish (as checked in the extracted datasets, explained in Section 5.1).

In terms of tweets location, initially we were not collecting only from Portugal, but then adapted the filter to the Portugal area to have only Portuguese tweets. For data collection in Portugal, a location filter was applied with coordinates 38.82687, -16.49414 and a radius of 1100 km (in order to include Azores and Madeira). The resulting area can be seen in the Figure 11.

¹⁸https://en.wikipedia.org/wiki/List_of_ISO_639-1_codes

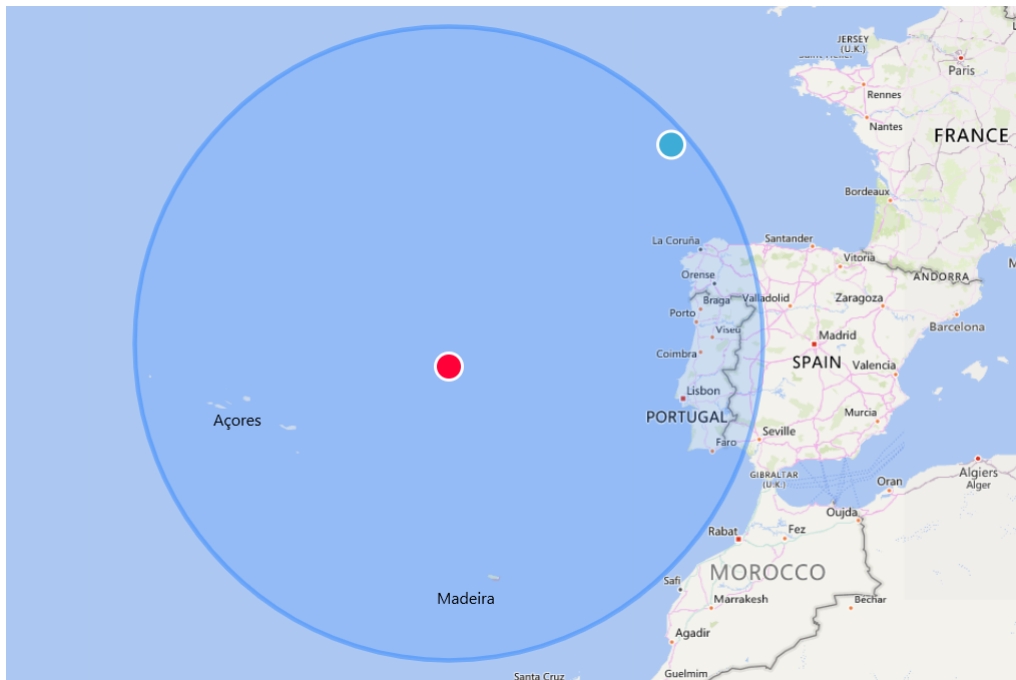


Figure 11: Collecting area of European Portuguese tweets

To illustrate the described request, Figure 12 shows an example of a request body for this endpoint, containing all the parameters explained.

```
{
  "db_name": "covid_pt_br",
  "seed_hashtags": {
    "mandatory": [],
    "optional": ["#vaificartudobem", "#omicron", "#vacinasovid", "#aquinaacovid", "#distanciasocial", "#covid", "#wuhancoronavirus",
      "#wuhancoronavirus", "#chinacoronavirus", "#coronavirus", "#coronaviruschina", "#coronavirus", "#covid", "#covid19",
      "#vacinacovid", "#viruschines", "#sarscov2", "#vacinacacovid", "#covid19brasil"]
  },
  "seed_keywords": {
    "mandatory": [],
    "optional": ["covid19", "covid", "coronavirus", "pandemia", "vacina covid", "virus chines"]
  },
  "since": "2021-08-01",
  "until": "2021-09-01",
  "seed_users": {
    "mandatory": [],
    "optional": []
  },
  "lang": "pt"
}
```

Figure 12: Example of a request body for tweets-seed endpoint

After extracting tweets that fit in the search parameters, in particular with the seed, they are stored in the database with a label of 1 by default, which means they are considered relevant. (the user later verifies this label in the final platform to validate the results). Then, to collect tweets non related to the topic, this endpoint is called with no seed which means the collected tweets do not have a predefined topic. Doing so, an initial binary classification dataset can be obtained to apply then the classification models. After having the dataset, it is stored in the database with the name defined in the input parameters. If the database already exists, it extracts to the existing database. If not, a new database is created. Each

database is composed of 5 collections: tweets, hashtags, keywords, users and stats. These collections are detailed in Table 7.

Collection	Description	Fields
Tweets	Extracted tweets	Tweet, Label, Created_at, hashtags
Hashtags	All the hashtags contained in the extracted tweets	Hashtag, label
Keywords	All the hashtags contained in the extracted tweets	Keyword, label
Users	Authors of the extracted tweets	user, label
Stats	Statistical data about the extracted tweets	Mean extraction time per tweet, data per day, tweets length, correlation matrix

Table 7: Collections of a database

Daily Collection

Using Threads in Python, a script was created to run every day in order to collect more data to the dataset. The script runs daily to collect new tweets based on the initial search parameters, but these parameters are dynamically changed overtime according to the new relevant hashtags and keywords found for the topic in question.

4.1.4 Exploratory data analysis module

The [EDA](#) module is one of the main modules of the extractor. It provides analysis functions to the created datasets, which will be later used on the Twitter Observatory framework (explained in Section 4.3). This will allow users to visually evaluate the quality of the dataset. To perform this analysis, the dataset is first read from the *mongo* collection to a pandas dataframe (by chunks to improve performance) and then some functions are implemented.

The defined functions analyse not only dataset variables but also the relationship between them. The first function is responsible for grouping the number of tweets extracted by day, based on statistical data stored in *stats* collection. Figure 13 shows the resulting plot, containing data of a an extracted COVID-19 dataset, as an example of the output of this function



Figure 13: COVID-19 dataset - data extracted per day

4.1.5 Topic Modelling

In order to increase the extracted datasets with relevant data for the topic, that was not collected in the first round of extraction, a mechanism was created using topic modelling, where one can locate topics on the collected dataset. This allows to increase the existing keywords based on the most significant words that were extracted for each subject. So, the focus here was to find automatically how to fetch these hashtags based on the initial collected tweets using the initial seed. If these hashtags are important, it is necessary to find out how to reach these tweets. Therefore, the approach to extract more relevant tweets was to use the [LDA](#) model, since it has been used in recent similar studies (cited in Section 2.1.6 of the State of the Art).

For [LDA](#) model training, two libraries were considered: *Sklearn*¹⁹ and *Gensim*²⁰. These libraries contain both an online version of [LDA](#) which has many advantages like providing a faster training and have the possibility to be updated in real-time. As tested by [101], the [LDA](#) model from *Sklearn* for tweets is faster than the one from *Gensim* and the first model was used.

The code developed for [LDA](#) model training is a python module inserted in the extractor. After the first extraction, it is applied to find relevant hashtags and keywords to use in the second iteration of the extractor.

4.2 Tweets Classification

After having datasets to work on, the next step is to classify them according to its relevancy. This section explains the development of the classification pipeline, based on the pipeline presented in Section 3.3. First, the preprocessing techniques used are detailed and then the [ML](#) and [DL](#) models defined are also

¹⁹<https://scikit-learn.org/stable/>

²⁰<https://radimrehurek.com/gensim/>

described. This pipeline was developed in Python, using PyTorch for the ML functions as a requirement from OmniumAI.

4.2.1 Preprocessing

Starting with the preprocessing module, it has the main goal of transforming the data collected with the Twitter Extractor into the formats necessary for both ML and DL models, as well as the LDA module. The adopted pipeline for tweet preprocessing was the same used in [42] due to the reasons explained in section 2.1.2. These preprocessing tasks are then applied to data that will be used to test ML and DL models. This is a twelve step pipeline, which consists of the functions mentioned in Table 8.

The pipeline defined in this table is designed to be applied to English texts, since the existing packages that perform some functions only exist for English language. To illustrate the preprocessing functions, the table's column *Preprocessing results* contains the result of applying each preprocessing function to this example tweet (taken from [42]): @UnitedAirlines Cooool I'm :) with servc! You ROCKED #urgr8 <http://ow.ly/Vlbf0>.

The Python package *Ekphrasis*²¹[110] was used for contractions' expansion, emoticons replacement (for example *:/* to *sad*) and word segmentation (which in case of tweets is hashtags splitting, for example, from *#ClimateCrisis* to *Climate Crisis*). This package is a text processing tool for text from social media, such as Twitter or Facebook, that performs tokenization, word normalization, word segmentation (hashtags splitting) and spell correction. To replace elongated words, *wordnet* from NLTK Corpus was used. The function verifies if the word to correct exists in the lexicon. If not, it is replaced to its basic form. As so, for example the word *alsooo* will become *also*. Finally, *Spacy*²² is employed for lemmatization, which is the process of obtaining a word's lemma.

For Portuguese language, the pipeline had to be adapted since there are no packages specifically for Portuguese to define some of the functions of the preprocessing pipeline, as to name hashtag splitting, contractions expanding and for example Portuguese hashtag splitting. As so, the preprocessing pipeline for Portuguese language is the following: remove urls, mentions, hashtags and replace emoticons; replace emojis; correct spelling; replace elongated words; remove punctuation; remove numbers; remove stopwords; lemmatize words.

The Portuguese lemmatization function was developed based on a script²³ developed by Ricardo José Lima, that intends to improve the lemmatization available in Spacy for portuguese. Despite the fact that there are still some issues with verb conjugation (for example the lemma of "confira" is "confrer"), the accuracy of lemmatization with the proposed method increased from 80.5% to 97.3% in a corpus of 861

²¹<https://github.com/cbaziotis/ekphrasis>

²²<https://spacy.io/>

²³https://github.com/ricardojosehlma/lemma_spacy_pt/blob/master/lematizacao_spacy_pt.py

Preprocessing function	Preprocessing results
1. Remove URIs, user mentions and hashtag symbols	<i>Cooooo I'm :) with servc! You ROCKED urgr8</i>
2. Replace emoticons and emojis	<i>Cooooo I'm happy with servc! You ROCKED urgr8</i>
3. Replace Slang and Abbreviations	<i>Cooooo I'm happy with servc! You ROCKED youare-great</i>
4. Correct spelling mistakes	<i>Cooooo I'm happy with service! You ROCKED youare-great</i>
5. Expand contractions	<i>Cooooo I am happy with service!You ROCKED youare-great</i>
6. Replace elongated words	<i>Cool I am happy with service! You ROCKED youare-great</i>
7. Remove punctuation	<i>Cool I am happy with service You ROCKED youare-great</i>
8. Lower-case words	<i>cool i am happy with service you rocked youaregreat</i>
9. Word segmentation	<i>cool i am happy with service you rocked you are great</i>
10. Remove numbers	<i>cool i am happy with service you rocked you are great</i>
11. Remove stopwords	<i>cool happy service rocked great</i>
12. Lemmatization	<i>cool happy service rock great</i>

Table 8: Preprocessing tasks

lemmas, according to the author. In addition to this function, a package developed for a dissertation project, the *NLPyPort*²⁴[111], was also investigated, but the results were similar and it was significantly slower than the first alternative. Therefore, we chose to employ the first function.

To optimize the application of the preprocessing pipeline to a dataframe, the package *swifter* from Python was used, which applies a function to a pandas dataframe in the most efficient way possible. The

²⁴<https://github.com/NLP-CISUC/NLPyPort>

final preprocessing module works for English and Portuguese, and its application to a specific language only requires changing the language parameter on the arguments.

4.2.2 Classification Models

The available ML models to train the data are the ones developed in OmniumAI: Random Forest, Support Vector Machines, Logistic Regression, from Sklearn package. Since the DL models are not implemented yet in OmniumAI, we had to define them. All the implemented models used PyTorch, as the tool agreed to use with the company. The developed pipelines are defined in Jupyter Notebooks, which are converted to python and execute by the application (explained in Section 4.3).

Figure 14 contains the pipeline used for Deep Learning Models. After passing the unverified labelled tweets through the preprocessing pipeline, feature extraction is performed using BERT embeddings. Then, the classification models are applied and the label is returned.

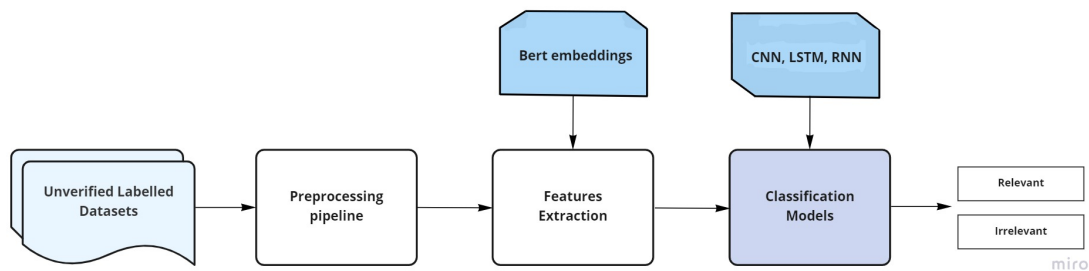


Figure 14: Pipeline used for Deep Learning models

For the tokenization, we used the BertTokenizer from Hugging Face, and more specifically the version of *neuralmind/bert-base-portuguese-cased*, which was trained with the brWaC corpus (Brazilian Web as Corpus), a crawl of Brazilian webpages, containing 2.68 billion tokens from 3.53 million documents. This tokenizer returns two outputs: *input_ids* and *attention_masks*. The first contains the indices of the words (obtained from the vocabulary file which is, in case, the *bert-base-portuguese-uncased*), and the second is an array of ones and zeros that indicates if each index corresponds to a word token (1) or a padding token (0).

Then, for the model, we used *BertForSequenceClassification*, also from *neuralmind/bert-base-portuguese-cased*. This version, called BERTimbau Base (referred in Section 2.3.6 of the State of the Art) is a pretrained BERT model for Brazilian Portuguese that achieves state-of-the-art results in three NLP tasks: NER, STS and RTE. It is available in two sizes: Base and Large. Due to computational resources limitations we used the Bert Base. BERT models have a limitation of 512 tokens per document but this was not a problem since tweets have at most 280.

4.3 Web Application

To conclude the chapter, this section presents the final framework architecture and its main functionalities, according to the methodology and requirements defined in Chapter 3. First, some decisions are explained and, then, the *Twitter Observatory* user interface is described.

4.3.1 Architecture

The application was developed in React²⁵, a javascript library for building user interfaces, and the choice to do so was motivated by the fact that OmniumAI uses this library for its frontend. For storage, *Redux*²⁶ is used. Redux is also a javascript library for managing and centralising the application state which makes the stored data to be independent from the application components.

Figure 15 presents the general diagram of the framework and contains the workflow of data. Since it is an application developed in React, the framework is divided in components. These components trigger the actions that get data from the store. To do so, the redux actions make HTTP requests to the Twitter Extractor or to other incorporated module, which in this case is the Sentiment Analysis module developed by Jorge Gonçalves mentioned in Chapter 3, and the results are then returned to the component, updating it. The resulting UI is hosted in the servers of OmniumAI.

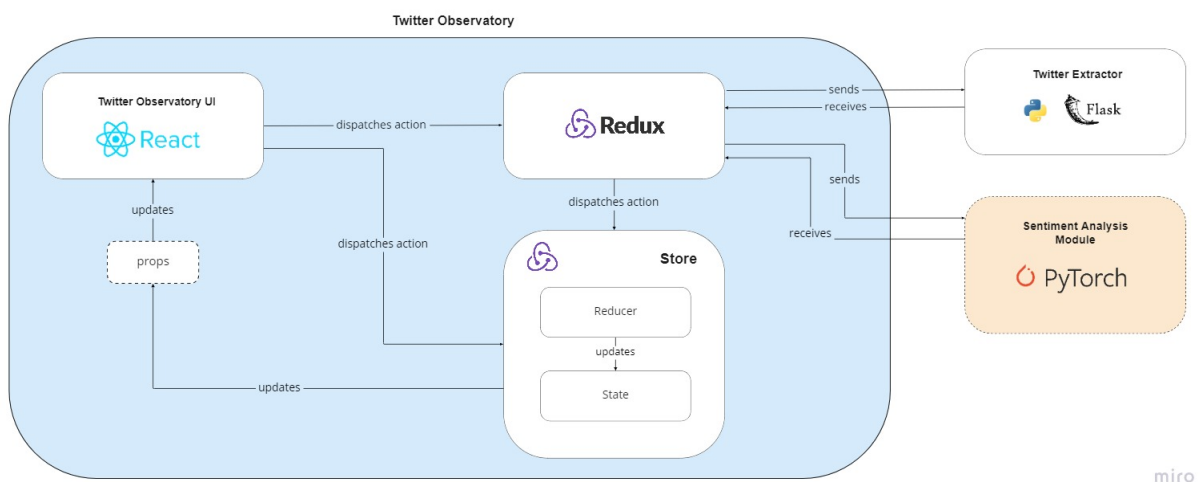


Figure 15: Twitter Observatory architecture diagram

4.3.2 Components

To have a clean and easy-to-use interface, the Material Kit React²⁷ was used, which is built with React Material UI components. The current version of the UI has 4 four main pages: homepage, database,

²⁵<https://reactjs.org/>

²⁶<https://redux.js.org/>

²⁷<https://github.com/minimal-ui-kit/material-kit-react>

extractor and running scripts. The homepage contains a list of existing databases, each redirecting to its own page. The database page is composed of plots of the [EDA](#) module and a zone that allows the user to change the dataset, either to modify labels, delete tweets or apply transformers of OmniumAI. Every time the user changes labels, the database is updated and the database field "verified" of the corresponding edited tweet turns to 1. Figure 16 gives an overview of a portion of the database page for a COVID-19 related dataset. In addition to general information, the characteristics of each column of the dataset are detailed, as well as the distribution of the labels and the amount of data that has been extracted per day.

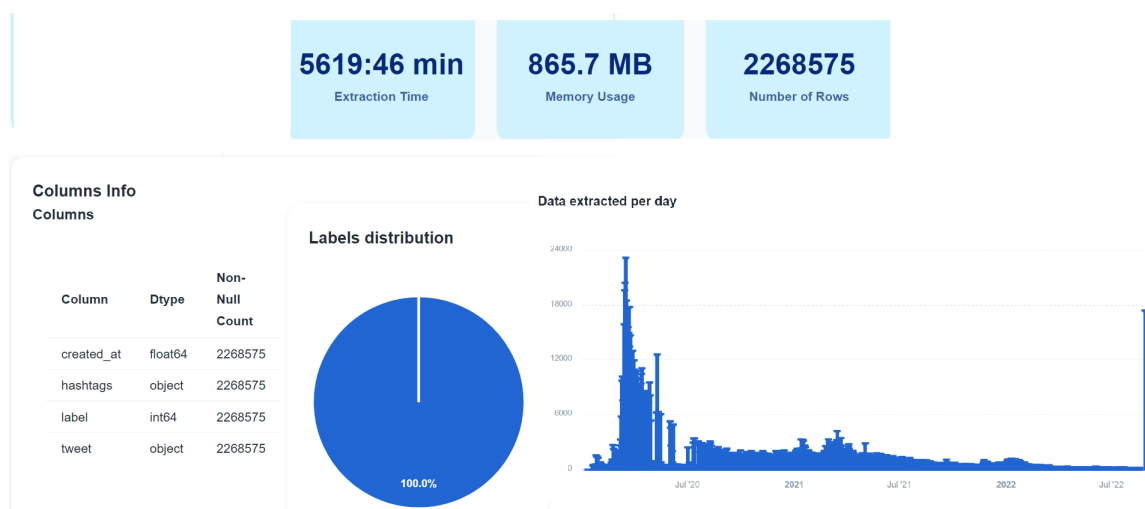


Figure 16: Portion of the Twitter Observatory database page for the COVID-19 dataset obtained with the Twitter Extractor

The [EDA](#) section contains all the analysis of the dataset, including columns information, the dataset itself (presented by chunks) and graphics regarding the extraction, the hashtags correlation and the labels distribution. For defining the graphics that are presented in this page, we used *React-ApexCharts*²⁸, which is a wrapper component for *ApexCharts*²⁹ to integrate in a React application.

Figure 17 contains the [UI](#) page that allows a user to perform the extraction. This component consists of a form with the fields needed to perform an [API](#) call to tweets-seed endpoint of Twitter Extractor. There is also another page that allows the user to add new datasets to the database without using the extractor, in csv format.

Summary

The `Twitter Extractor` is a Flask framework developed for the purpose of extracting relevant tweets according to a certain topic to obtain and store datasets for posterior analysis. It was developed by modules, each one addressing a specific task regarding database operations, [EDA](#) analysis and Twitter

²⁸<https://apexcharts.com/docs/react-charts/>

²⁹<https://apexcharts.com/>

The screenshot shows a web application interface for extracting tweets. On the left is a sidebar menu with three items: 'Home' (with a globe icon), 'Extract Tweets' (with a circular arrow icon), and 'Running' (with a play icon). The 'Extract Tweets' item is currently selected. The main content area is titled 'Extract tweets from seed'. It contains several input fields arranged in a grid-like fashion. The first row has 'mandatory hashtags' and 'optional hashtags'. The second row has 'mandatory keywords' and 'optional keywords'. The third row has a single field for 'users'. The fourth row has 'since' and 'until'. The fifth row has a single field for 'lang'. The sixth row has a single field for 'db_name'. At the bottom of the form is a prominent blue button labeled 'Extract'.

Figure 17: Twitter Observatory extraction page

collection of data. It uses an open source extraction tool for Twitter, named Twint, which makes use of search operator from Twitter API. Each process of the extraction pipeline was explained in detail starting by the architecture, followed by the seed composition, the extraction parameter and the topic model developed to improve the seed and, consequently, the dataset. The labelling processing, at this initial phase, consists of automatically assigning the initial extracted tweets with 1 for relevant tweets and with 0 for those non related to seed. This unverified labels will then be validated by the user in the [UI](#) interface.

Then, the classification pipeline is described, starting with preprocessing techniques adopted and then described the models used. The code regarding the classification is defined in *jupyter notebooks* that can be converted to python scripts when training or predicting with the extracted data.

The final framework, Twitter Observatory, was also presented starting with the architecture and then explaining the technologies used and the components developed. This framework integrates the previous pipelines and a [UI](#) that allows users to interact with the extracted data. This [UI](#) is defined by modules, which in React are called components, and it is easy to introduce more modules as it is needed. In the next chapter, the results obtained with this framework are presented in two selected use cases.

Results and Discussion

The previous chapter described in detail the developed framework entitled *Twitter Observatory*. This chapter collects the results obtained with this framework and interprets them regarding the initial purpose of collecting relevant data for a topic of study. First, the extractor is validated by analysing the two datasets obtained with different seeds. Then, the results of the classification models are presented and, finally, the Twitter Observatory's final functionalities are discussed.

5.1 Datasets

The first analysis step consists of evaluating the datasets extracted using the Twitter Extractor and how much data can be collected regarding a certain topic. For our use case, two topics were selected: COVID-19 and the Russian invasion of Ukraine in 2022 (sometimes referred in this document as *Ukraine War*). The main focus was to study data in Portuguese, as was mentioned before, and so these topics were chosen not only based on popularity but also on the existing amount of data in Portuguese regarding these topics.

Because of the significance of COVID-19 in the last two years, this topic was the first to be picked. Between 1 February 2020 and 31 March 2021, there were at least 2,372,648 tweets in Portuguese on this topic, making Portuguese the third language with the most tweets about COVID-19 [112]. Then, the Russia invasion of Ukraine was another chosen topic because it happened in the current year of this project's development and there were at least 899,671 tweets in Portuguese about this topic [113].

As the Portuguese Language is divided in European Portuguese and Brazilian Portuguese we had to decide whether to use only one variant or both. Although there are some significant differences between the two variants [114], the final decision was to use both Brazilian and European Portuguese tweets to have more data. According to [115], in a dataset of Portuguese tweets from January 1 to May 31, 2020,

related to COVID-19 data, only 4% were not from Brazil.

To collect relevant datasets we started by defining the initial seed of keywords, hashtags and users regarding the chosen topic, and then compared them with extracted datasets found in the literature related to the topic. By the time of extraction, our datasets were classified with a binary-classification: 1 (relevant) and 0 (irrelevant). This classification is made based on the definition of relevance given in Section 2.6. A first approach to the labelling process (before atributing automatically a label when extracting), was done using *Label Studio*¹, but it was time consuming as a tweet could take around five seconds to label, which means that a 1,000,000 entries dataset would take 277 hours to label. So, and since the manual classification was not the main purpose of this work, it was decided to assume that tweets extracted under the initial condition of search (the initial seed) were necessarily relevant for the topic in question. This means that in our case we are just classifying the tweet as relevant or irrelevant if it is semantically related to the topic without looking at its context. This latter case would imply a human validation. To exemplify, a tweet containing the word "*pandemia*" (in english, *pandemic*) does not need to be necessarily related to COVID-19 because it does not contain any useful information of the topic, according to the use case, but it is semantically related, as this one: "*Depois da pandemia, os manuais escolares voltaram a ser usados*" (in english, "*After the pandemic, textbooks are back in use*") can be considered irrelevant for the COVID-19 case, although it contains the word "*pandemia*".

Still, on the subject of tweet labelling, we also tried to use Tweets context annotations², a functionality of Twitter API that attributes a topic to a tweet based on a semantic analysis of keywords, hashtags, handles. This classification attributes one or more topics to the tweet and each topic contains a domain object and an entity object. An example of context annotations applied to a tweet was given in Section 2.4. This classification process behind the context annotations is similar to the process adopted in this work, and so this was presented as a good alternative for extracting tweets according to the pretended topic and have the tweets automatically labelled. Nevertheless, after a few tests, it was noticed that there are barely no annotated tweets for the Portuguese language. These tests consisted in searching for tweets with annotations that are listed here: <https://github.com/twitterdev/twitter-context-annotations>.

Table 9 presents a general overview of the extracted datasets regarding the two topics chosen. Besides these two, a third dataset, from CrisisLex dataset (mentioned in Section 2.1.1), is also presented in the table. This dataset will be used for validating the DL results obtained. We now present a thorough explanation of each dataset.

¹<https://labelstud.io/>

²<https://developer.twitter.com/en/docs/twitter-api/annotations/overview>

Topic	Number of Tweets	Location	Time of Extraction
COVID-19 pandemic	2,268,575	Portugal and Brazil	93 hours
2022 Russian invasion of Ukraine	219,887	Portugal and Brazil	2h48 min.
2013 West Texas Explosion	10,007	All countries	N/A

Table 9: Extracted Datasets characteristics

5.1.1 COVID-19

Starting with the COVID-19 topic, the words related to the pandemic were aggregated in the optional list of keywords and the optional list of hashtags, regarding the parameters for extraction used by Twitter Extractor. These hashtags were obtained from three datasets: [112], [116] and [5]. The first dataset, *TBCOV*, contains 2,014,792,896 tweets related to COVID-19, collected from 1 February 2020 till 31 March 2021, and includes sentiment labels, named entities and user information. It was extracted based on a 800 multilingual tokens set (keywords and hashtags). Regarding Portuguese tweets, it contains 2,372,648 tweets. The second dataset contains 12,278,028 tweets in Portuguese collected from February 1, 2020 till May 1, 2020 using also a multilingual set of keywords and hashtags. The third one contains 3,155,277 tweets in Portuguese regarding COVID-19 data. Based on the hashtags and keywords used to extract these datasets, our extraction seed was defined and is presented in Appendix Appendix B. Many of the hashtags present in this seed are just variants of a word with spelling errors, as to name some examples: regarding "coronavirus", there is "coronoavirus", "coronovirus" and others. To add more tweets to the dataset, a second round of extraction was performed but based on a list of users (presented in Appendix A). This list was created based on official accounts (*authentic* accounts with the blue badge, according to Twitter) from Portuguese newspapers and health accounts, and other manually found based on the followers of those accounts.

In which concerns to the labelling process, all the tweets extracted with this seed were classified as relevant (label 1) because all the tweets contain, at least, a keyword or an hashtag related to the topic and so it is considered relevant according to definition given in Section 2.6. To collect non-related COVID-19, a Twitter Extractor API call was made with an empty seed. The response consisted of tweets from the days the request was made (in august), which are random tweets normally without an hashtag associated.

Figure 18 shows the data extracted per day for this dataset using our extraction tool, regarding only tweets labelled as relevant.

When comparing our dataset to the three datasets found in the literature about COVID-19 pandemic and mentioned above, the former has a lower number of entries. This is happening for two main reasons: it is not possible to collect all tweets satisfying the seed because around only 1% of the tweets is collected, unless the academic research access level of Twitter API is used (which is not our case); some of the

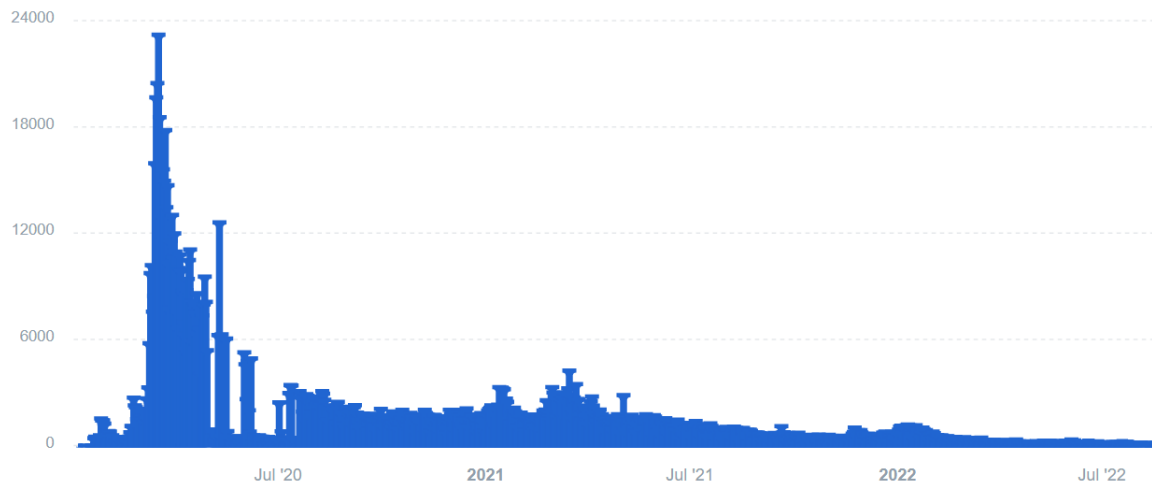


Figure 18: COVID-19 dataset - data extracted per day labelled as relevant

tweets from the datasets found in the literature has already been deleted. This was concluded when confronting the extracted data with the existing datasets for COVID-19. By comparing the tweets ID, it was possible to confirm that many of the collected tweets in those datasets were already deleted in Twitter by the time of our extraction and, therefore, were not possible to collect. The daily collection script kept running alongside the months the data was being extracted but the dataset did not increase significantly since in the last months, COVID-19 related tweets has been decreasing (as shown in Figure 18).

To analyse this dataset, we used the functions provided in the [EDA](#) module made available in Twitter Extractor. Since this module is integrated in the developed [UI](#), the graphics presented in this subsection can all be found in the database page of the [UI](#). A first observation of the dataset shows that COVID-19 related tweets length is higher than non COVID-19 tweets length. Figure 19 illustrates the length of tweets related to COVID-19 and the length of tweets not related to the topic. This difference can be explained by that fact that many of COVID-19 related tweets are informative posts from online newspapers and health organisations accounts. These tweets contains numbers of detected cases, the evolution of pandemic in many countries and the daily reports provided on the news. On the other hand, non related COVID-19 tweets are simply messages from users, often written in slang, and sometimes are just a short answer to another tweet or a comment on a trending topic.

In which concerts to hashtags analysis, Figure 20 show the correlation between the hashtags present in COVID-19. This chart was obtained in the database dashboard of the [UI](#).

Figure 21 contains the most common bigrams either for relevant tweets or irrelevant ones regarding this dataset, after tweets preprocessing. It shows that is no similarity between the content of tweets from the two labels, which indicates that it could be a good dataset for training the models.

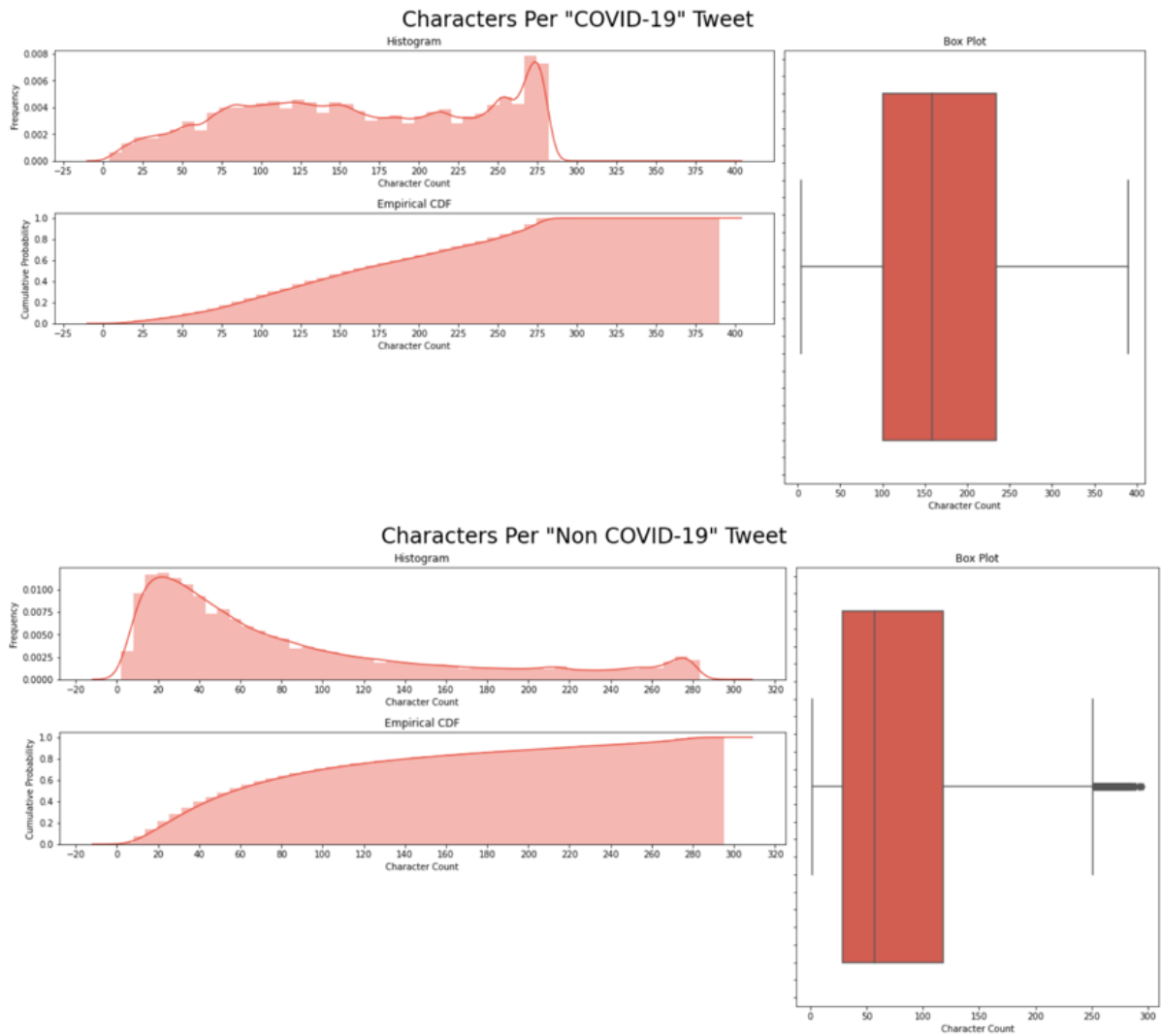


Figure 19: Characters frequency of COVID-19 dataset tweets

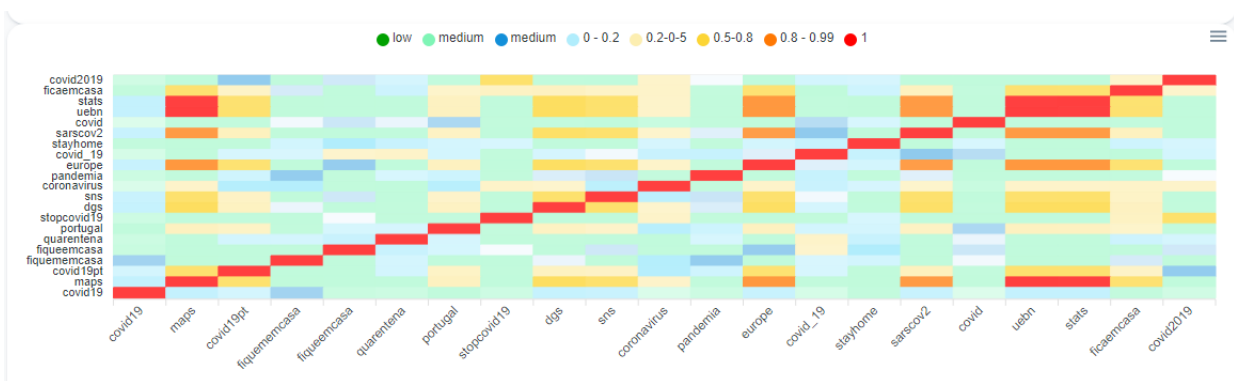


Figure 20: Hashtags correlation for COVID-19 dataset

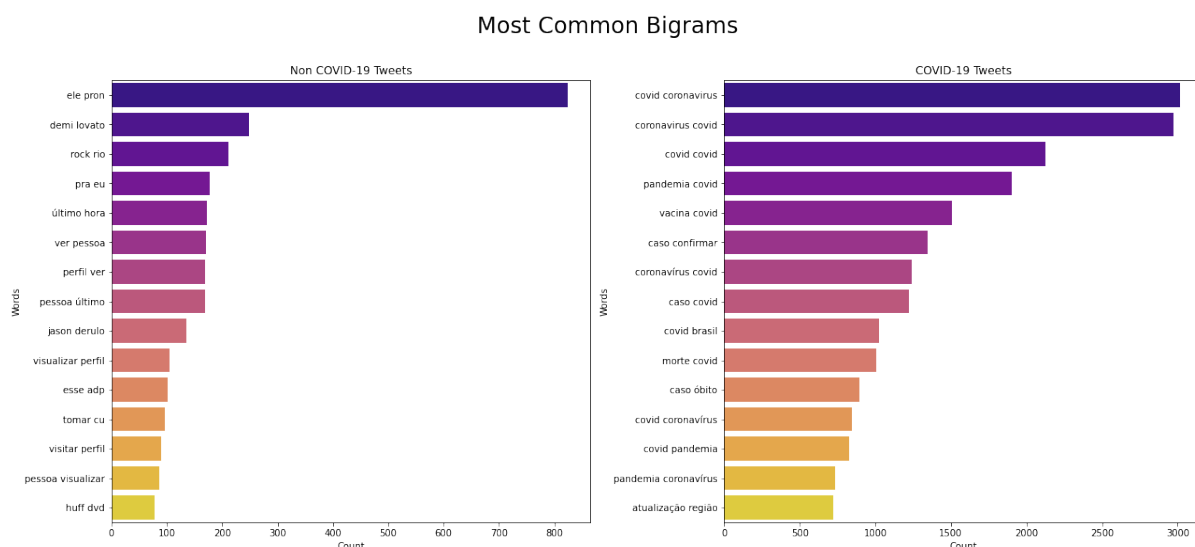


Figure 21: Most common bigrams for COVID-19 dataset related tweets

5.1.2 Russian invasion of Ukraine

This topic became very popular on Twitter at the beginning of the year since the war began in Ukraine in the final days of February [117]. We started collecting data regarding this topic in the beginning of March. The initial seed to extract this dataset was based on [113], which contains the top 15 hashtags for this topic: *ukraine*, *russia*, *putin*, *standwithukraine*, *kyiv*, *ukrainerussiawar*, *stopputin*, *ukrainerussianwar*, *russian*, *ukraineunderattack*, *nato*, *stoprussia*, *kiev*, *ucrania*, *ukrainian*. The Appendix C contains the seed that we used to extract this dataset, using the Twitter Extractor. The time period of data collection for this dataset is from first of february of 2022 to eighth of august of 2022.

Unlike what was done for the COVID-19 dataset, it is not possible to directly compare the number of tweets in our dataset and the dataset discovered in the literature because the latter only contains tweets in English. However, to include non-related tweets to the topic the process was similar to the process employed in the COVID-19 dataset. To create a balanced dataset then tweets from "today" or "yesterday" are extracted, regarding the day of extraction. Besides, some of those irrelevant tweets (the shorter ones, with barely no corrected words, just slang or stop words) were classified as Portuguese tweets by Twitter but are in Italian ou Spanish. This may happen due to the fact that they are too short.

Figure 22 contains a chart with data extracted per day for this dataset using our extraction tool, regarding only tweets labelled as relevant. The high volume of data at the beginning of March is explained by the fact the Russian Invasion of Ukraine began at the end of February and so people then started to talk about this topic frequently.

Data extracted per day

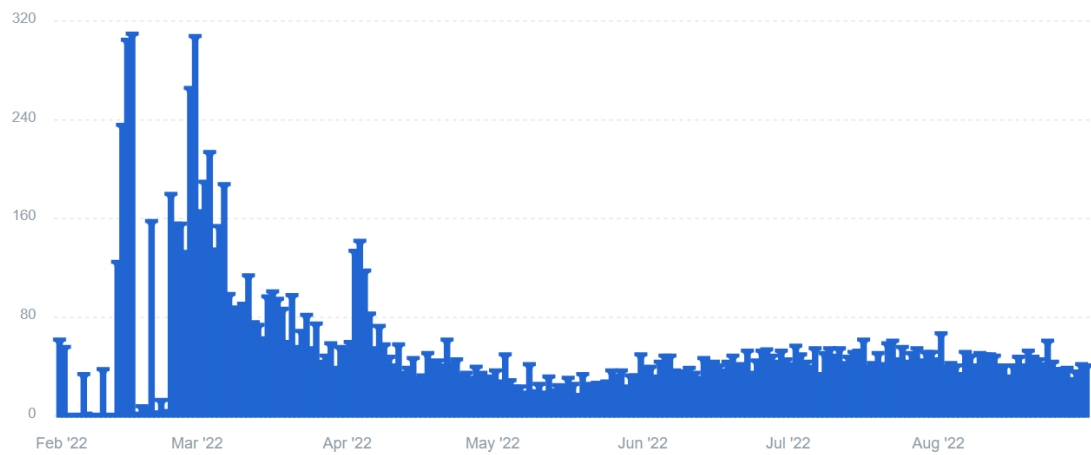


Figure 22: Russian Invasion of Ukraine dataset - data extracted per day labelled as relevant

Similarly to COVID-19 dataset, Figure 23 contains the most common bigrams for relevant and non-relevant tweets for the Russian Invasion of Ukraine dataset. The most common words are significantly different.

Most Common Bigrams

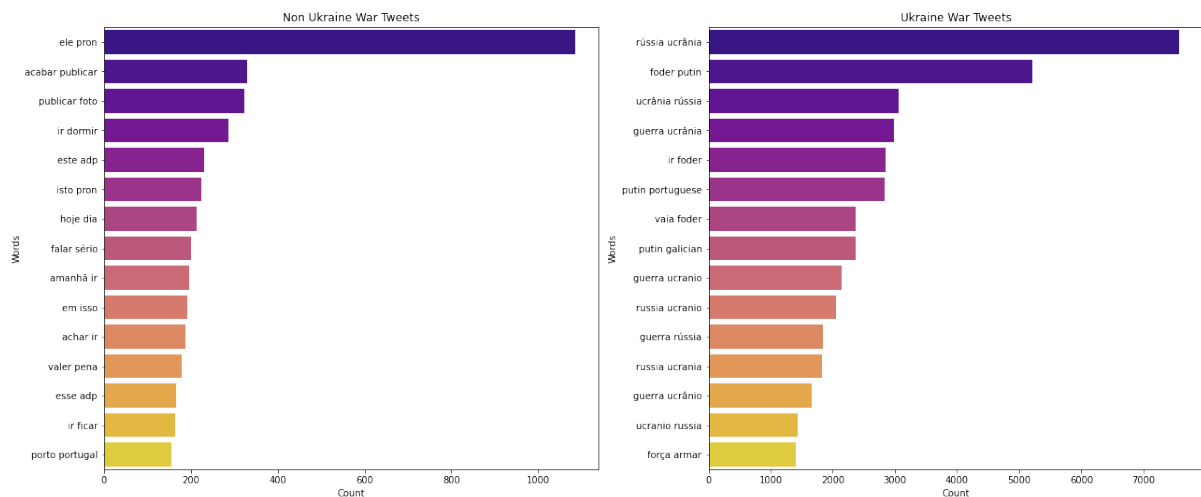


Figure 23: Most common bigrams in tweets of Ukraine War dataset

Figure 24 shows the tweets length regarding related and non-related tweets to the topic of Russian Invasion of Ukraine. In general, and similarly to the previous dataset, non related to the topic tweets are shorter then the other ones. This may happen due to the fact related tweets are exposing information unlike the non related, which are more like simply messages to friends.

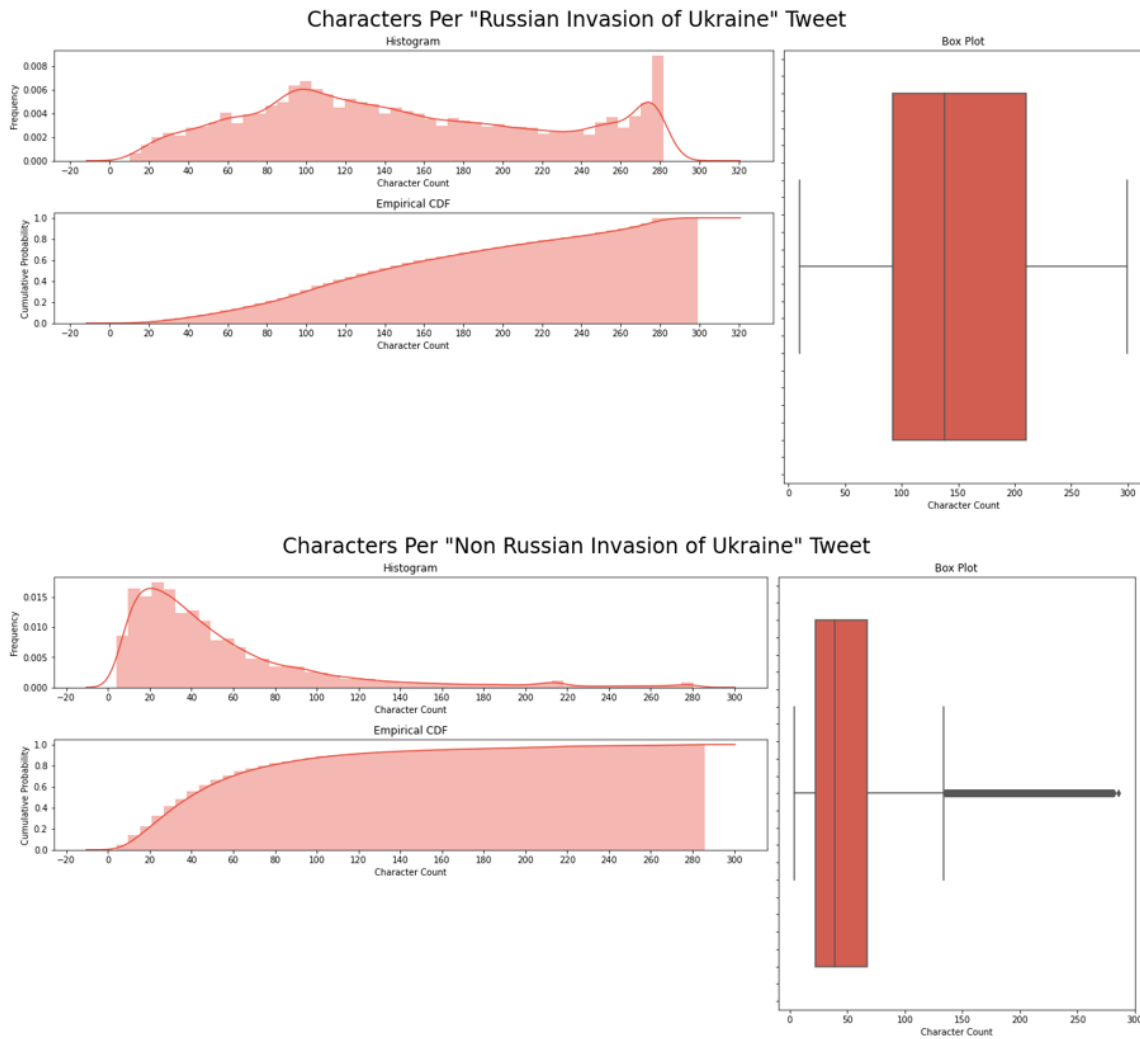


Figure 24: Characters frequency of COVID-19 dataset tweets

5.1.3 Validation Dataset - CrisisLex

To validate the classification results with a verified labelled dataset, we decided to use the best dataset of relevant and irrelevant tweets found for document classification in the literature: the CrisisLex dataset (mentioned in Section 2.1.1 of the State of the Art). The CrisisLex consists of various disaster-related datasets were obtained from Imran et al., 2013, Olteanu et al., 2014. These tweets were gathered throughout seven crises that happened in 2012 and 2013, including both natural and man-made crises. These seven crises included the 2011 Joplin tornado, the 2012 Sandy hurricane, the 2013 Alberta floods, the 2013 Boston bombings, the 2013 Oklahoma tornado, the 2013 Queensland floods, and the 2013 Texas explosion. This set of datasets contains 70,000 tweets having binary labels of relatedness such as relevant or irrelevant regarding the topic. The labeling of messages was done through the crowdsourcing platform Appen³. Since we could not find a labelled dataset related to document classification in

³<https://appen.com/>

Portuguese, we selected one of the datasets from CrisisLex and then translated it using a multilingual model from HuggingFace, the *mBART-50*⁴. This model is pre-trained with the Opus-100 dataset, which contains sentence-pair in many combinations of languages sources and targets. Before using this model, *TextBlob*⁵, from Python, was testes but with no successful results since it uses Google Translator and, consequently, has usage limits per day.

5.2 Twitter Extractor Evaluation

After evaluating the datasets obtained, it is important to understand the Twitter Extractor performance and how it can be improved in the future. One important part of this tool development was tracing its performance in time, so we could measure the computational resources used and evaluate what could be optimised. This tool was developed in a machine with a x86_64 architecture running Windows 10, having having 4 Central Processing Units (CPUs), 236 GB of [Solid State Drive \(SSD\)](#) disk space and 8GB of [Random Access Memory \(RAM\)](#).

To monitor this application and check the results in order to optimise the extractor we used Elastic-apm⁶, from Python. In terms of computational resources regarding the development environment, it used in average 96% of the available memory and the mean extraction time of a tweet was 0.06 seconds. The general performance of the extractor can be optimised by paralleling its execution when extraction includes long periods of time, as it was the case of COVID-19 data. This means that each instance of Twint that is created by this tool to extract tweets for each day could be executing at the same time. The deployment of this tool to production was done in an OmniumAI server.

5.3 LDA results

As explained in the previous chapter, to improve the extraction with more tweets that may have not been extracted before using only the initial seed, the [LDA](#) was applied to cluster tweets by subtopics of the main topic and then check if the words that characterise each topic are already on the seed or not. If not, the seed is increased with the new keywords and the extraction is again performed. To train the [LDA](#) model over the developed datasets, a sampling was done to the dataset since these datasets were too big to train the model with the available resources (a *Tesla P100-PCIE-16GB* GPU). A stratified sampling was applied to have both classes balanced. The search space used to optimise the model training is presented in Table 10.

⁴<https://huggingface.co/Narrativa/mbart-large-50-finetuned-opus-en-pt-translation>

⁵<https://textblob.readthedocs.io/en/dev/>

⁶<https://pypi.org/project/elastic-apm/>

Parameters	Values
Number of topics	10
Random State	45
Number of Components	[10, 15, 20, 25, 30]
Learning Decay	[0.5, 0.7, 0.9]

Table 10: PyLDAviz print-screen of the visualisation of a topic from COVID-19 dataset

We used pyLDAvis⁷, an interactive visualisation tool, to interpret the results from LDA. Figure 25 shows how this package presents the results and exemplifies the analysis of a detected topic on COVID-19 dataset. As also concluded in [63], the results of applying the LDA model to our data did not bring good results as it is difficult to distinguish the different topics detected by the model.

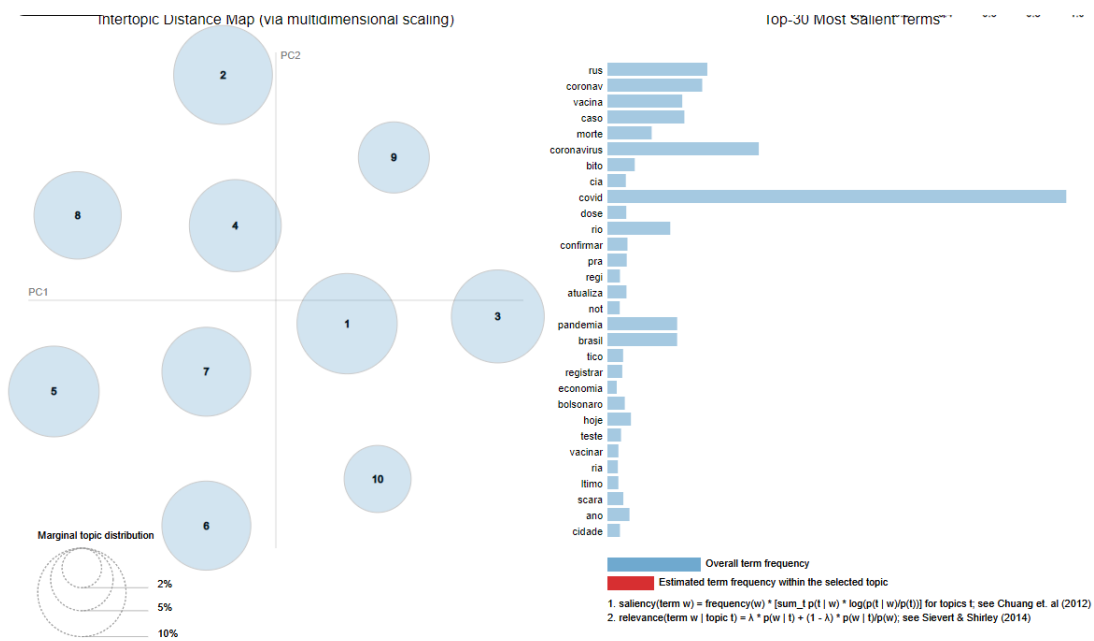


Figure 25: LDA Topics for COVID-19 dataset

⁷<https://github.com/bmabey/pyLDAvis>

5.4 Deep Learning Results

The DL results related to the models implemented in the pipeline classification are presented here. Since the first dataset, related to COVID-19, was quite large and the computer resources were limited, the strategy adopted here was to sample the dataset. A stratified sampling was applied, which consists in sampling the dataset in way that the size of each category in the resulting dataset is proportional to the size of that category in the original population. This was performed in order to have a balanced dataset. This dataset was then divided into train, validation and test sets and the number of tweets regarding each set is presented in Table 11. The sampled dataset contains 100,000 tweets.

The first DL model used in this work was BERTimbau from *HuggingFace*. As stated in Section 2.3.6, this model achieved the state-of-the-art performance on three Portuguese NLP tasks: STS, RTE and NER. Regarding document classification tasks, this model was applied to the obtained portuguese datasets to check if it could also achieve good results to document classification tasks. We used the BERTimbau base version which is composed of twelve layers. Before applying the model, the BertTokenizer, also from HuggingFace *neuralmind*, was applied to the tweets.

Dataset	Documents	Relevant	Irrelevant
COVID-19 Training	53,600	35,963	17,638
COVID-19 Validation	13,400	9008	4393
COVID-19 Test	18,971	22,243	10,757
Ukraine War Training	117,854	59,079	58,776
Ukraine War Validation	29,464	14,482	14,983
Ukraine War Test	72,560	36,181	36,279

Table 11: Train, validation and test split of COVID-19 and Russian Invasion of Ukraine datasets for BERTimbau model

Since BERT authors recommend between two and four epochs, we trained with three epochs. The optimiser used was AdamW and the batch size was 32. Regarding BERTimbau obtained results, validation accuracy and validation F1-Score are 0.96 and 0.97 respectively. Table 12 contain the values obtained for the two extracted datasets test sets, regarding accuracy and F1-Score metrics. The results achieved by the model lead us to conclude that the predictions made for the test set since the pattern to detect the relevant tweets is easy to learn. As seen in the previous sections of this chapter, the tweets related to the topic, in both cases, are bigger than the non related and the most common words present on each dataset are totally different.

Dataset	Accuracy	F1-Score
COVID-19	0.96	0.97
Russian Invasion of Ukraine	0.99	0.99

Table 12: Results obtained on the test set for COVID-19 dataset and for Russian Invasion of Ukraine dataset

5.5 Discussion

Since there was no datasets found in the literature with Twitter Portuguese data labelled for document relevance it was necessary to create our own datasets and for that, two use cases were selected: COVID-19 and Russian Invasion of Ukraine. Regarding these two topics, two datasets were created using the Twitter Extractor to test not only the extractor performance but also the classification models results when applied to these datasets.

The process adopted for the extraction was to define a seed based on known keywords and hashtags found in the literature. Based on that seed, the relevant tweets for each dataset were extracted. To add irrelevant tweets to each dataset, an empty seed was the input for the extract to collect random tweets. This means that by the time of extraction, the tweets were automatically labelled by the extractor. This labelling process can be further improved using the semi-supervised approach that is implemented in the UI but not tested with real users.

To test the predictions of the models trained with these two datasets, we used a third dataset, *CrisisLex* which was found in the literature and contain English from natural disasters. This dataset was already manually labelled with a binary classification: relevant (0) and irrelevant (1). This dataset was translated to portuguese using a translation model and then BERTimbau was applied to predict the labels.

Conclusion

6.1 Summary of the work

As explored in the State of the Art, most of the Twitter and text mining related works rely only in studying and improving the document classification approaches, using existing datasets or extracting data with the Twitter API without a rigorous or scientific criterion. When a new dataset is needed and no other existing dataset in the literature satisfies the requirements of the study, there is no generic framework that allows obtaining a labelled dataset to the specific topic of research. Based on that fact, this project aimed to explore not only the existing tweets classification approaches, but mainly the development of a generic extraction method for Twitter.

Regarding its implementation, this work followed the four-phase text mining process (mentioned in Section 2.1), starting by collecting data (Chapter 4.1), followed by document classification (Section 3.3), which includes the preprocessing step (Section 4.2.1), and finally the analysis of the results (Chapter 5). The main purpose of this work was to develop a framework capable of extracting relevant datasets, regarding a certain topic, to provide useful data for many tasks as classification, sentiment analysis or other text mining tasks. This final framework extracts tweets regarding an initial seed of users, hashtags and keywords and is updated daily. Then, all useful analysis of these datasets can be visualised by the user in the developed UI.

Extracting high-quality data is a challenge when it comes to short data, as it is the case with 280 character tweets, and the labelling process is not easier. The adopted strategy to automatically label tweets using our tool, was to follow the process that Twitter uses to attribute context annotations to tweets. This labelling process can contain a percentage of error, but it was not restrictive to achieve good datasets. Using the seed defined in Chapter 5 and the Twitter Extractor, it is possible to obtain similar datasets to the ones created for this work and reproduce the classification results. As a future work, further mentioned,

it would be important to improve these results by refining the labelling process using manual labelling, using the Twitter Observatory UI.

Although the bigger purpose of this work was the data extraction and subsequent analysis, focusing on extracting relevant data for a topic of interest, some state-of-the-art DL models, namely BERT and XLNet, were tested to classify the extracted data in a binary classification: *relevant* or *irrelevant*. The results achieved were good. However, it is important to note that these results tend to be biased since the labelling processing was automatic and there was no human validation. Therefore, the pattern used to label the tweets is easier to detect by the models.

To test the classification pipeline, two datasets were extracted, the first related to the COVID-19 pandemic and the other related to Russian invasion of Ukraine. This datasets contain 2,268,575 and 109,938 Portuguese tweets, respectively. The best achieved metrics for Bert was 0.99 for both accuracy and F1-Score for Russian Invasion of Ukraine dataset. As stated before, these results tend to be biased. And this was not not the main focus of this project, since the main goals were the extraction and posterior analysis of the obtained data. Since the framework was developed by modules, the intention here is to add more classification modules regarding other text mining tasks.

Regarding the initial hypothesis (referred in Section 1.2), can a structured extraction method, as well as a defined classification approach lead to obtain useful datasets for achieving the state-of-the-art results for document classification tasks? The answer was presented alongside this conclusion. The datasets obtained and the classification results achieved validate this hypothesis.

In which concerns the utility of this work, the developed framework can eventually lead us to solve real life problems in a timely manner. Although there is still much space to improve the Twitter Observatory, having a framework capable of constantly collecting tweets, improving the extraction and classifying them as being relevant or irrelevant to a search topic will probably be useful for many situations, including natural disaster situations, politics events predictions or scientific research.

6.2 Contributions

As far as we know about what was previously done in this field, no other related work studied the whole text mining process for Portuguese tweets. A state-of-the-art summary of Portuguese twitter data gathered so far was conducted; a generic framework was developed, allowing the extraction of tweets in many languages, preprocessing them and analysing the data in a developed UI.

So, the main contributions of this dissertation are:

- A review of the state-of-the-art results for document classification tasks regarding Twitter data and of the related work and existing tools for tweets extraction;
- An extractor, based on a Flask API, *Twitter Extractor*, capable of collecting tweets with no temporal limit and for all languages and locations and automatically labelling those tweets;

- A web application, with a [UI](#), that allows users to visualise and analyse the datasets extracted with the Twitter Extractor and edit the unverified automatic labels;
- A framework to extract and classify tweets, *Twitter Observatory*, that includes the extractor and the web application, and was integrated into OmniumAI software.

This final framework also incorporates the work developed by Jorge Gonçalves for his dissertation, related to sentiment analysis.

6.3 Future Work

Since the developed framework is the first approach to the four phases of text mining regarding Portuguese Twitter data, there are many modules that can be refined and improved. Therefore, the prospects for future work are now detailed.

Labelling Process. Since the tweets can be classified as relevant according to the context/user case or simply by semantic similarity, the labelling process can be improved by adding a third label, for example "Can't decide", which the user attributes to the tweet whenever he does not know the classification criteria. To test the user validation of the automatic labels would also be important.

Twitter Configurations. The Twitter Extractor was developed based only on keywords, hashtags and users but a deeper exploration of the context annotators given by the Twitter API, which attribute a topic to a tweet, could be explored in order to check if this could improve the extraction criteria and seed. Also, the evaluation of number of retweets could be explored to eventually be a factor to check if the tweet is relevant, as it would also be important to explore user's accounts parameters as *followers* and *following* to eventually reach more useful accounts to be part of the seed. Although the context annotations are not much explored for Portuguese, they can be used to extract English data, for example. In essence, it would be important to understand how the relevance defined by Twitter could contribute to the label attribution that is being done.

Tweets Preprocessing. Since tweets are short documents (with a maximum of 280 characters), the preprocessing pipeline has a bigger impact on the classification part. Having that in mind this can be highly explored and a benchmark of different preprocessing pipelines can be done. Reaching the best pipeline can highly improve the model training and predictions.

Topic detection. The [LDA](#) exploration that has been started in this project can be further improved and tested to check if better results can be achieved. Furthermore, a deeper exploration of this approach can lead us to know the minimum number of documents necessary to detect the topics of the documents, for example. Besides that, fuzzy fingerprints approach can be studied and compared with [LDA](#) results to conclude which technique fits better for tweets topic detection and, consequently, improve the relevant tweets extraction with the best technique.

Extractor Extension. Although this framework was developed in a generic way and by modules, only Twitter data was explored since it was the selected use case. However, it would be easy to extend the

extractor to handle information from other social platforms. This would just imply to create a new python module, that would work as a wrapper for the respective social network [API](#). Also, we could explore the extractor for different topics other than the ones explored in this dissertation and extend the classification pipeline for other languages.

Usability Tests. The framework should be tested with real users to confirm if the functionalities are easy to use and to check if those interactions with the framework improve the classification results, particularly the labels validation task. To do such tests, a questionnaire could be done with questions that inquire about the [UI](#) functionalities and how they are reaching their purpose. The results from such questionnaire can lead to improve the [UI](#).

Bibliography

- [1] S. Lal et al. "Analysis and Classification of Crime Tweets". In: *Procedia Computer Science* 167 (2020). International Conference on Computational Intelligence and Data Science, pp. 1911–1919. issn: 1877-0509. doi: <https://doi.org/10.1016/j.procs.2020.03.211>. url: <https://www.sciencedirect.com/science/article/pii/S1877050920306761> (cit. on pp. 1, 17).
- [2] *Twitter Basics*. <https://media.twitter.com/en/twitter-basics>. (Accessed on 03/14/2022) (cit. on p. 1).
- [3] M. M. Agüero Torales et al. "Machine Learning approaches for Topic and Sentiment Analysis in multilingual opinions and low-resource languages: From English to Guarani". In: (2022) (cit. on p. 1).
- [4] Z. Mottaghinia et al. "A review of approaches for topic detection in Twitter". In: *Journal of Experimental & Theoretical Artificial Intelligence* 33.5 (2021), pp. 747–773 (cit. on pp. 1, 17).
- [5] K. Garcia and L. Berton. "Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA". In: *Applied Soft Computing* 101 (2021), p. 107057. issn: 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2020.107057>. url: <https://www.sciencedirect.com/science/article/pii/S1568494620309959> (cit. on pp. 2, 16, 19, 20, 44).
- [6] D. A. Pereira. "A survey of sentiment analysis in the Portuguese language". In: *Artificial Intelligence Review* 54.2 (2021), pp. 1087–1115 (cit. on p. 2).
- [7] M. BAYGIN. "Classification of Text Documents based on Naive Bayes using N-Gram Features". In: *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*. 2018, pp. 1–5. doi: [10.1109/IDAP.2018.8620853](https://doi.org/10.1109/IDAP.2018.8620853) (cit. on pp. 2, 9).
- [8] B. Behera, G. Kumaravelan, and P. Kumar. "Performance evaluation of deep learning algorithms in biomedical document classification". In: *2019 11th International Conference on Advanced Computing (ICoAC)*. IEEE. 2019, pp. 220–224 (cit. on p. 2).
- [9] R. Bayari and A. Bensefia. "Text mining techniques for cyberbullying detection: state of the art". In: (2021) (cit. on pp. 2, 17).

-
- [10] XLNet, ERNIE 2.0, And RoBERTa: What You Need To Know About New 2019 Transformer Models. en-US. Aug. 2019. url: <https://www.topbots.com/ai-nlp-research-big-language-models/> (visited on 01/01/2022) (cit. on p. 2).
 - [11] P. Beytül et al. "Visual gender biases in wikipedia: A systematic evaluation across the ten most spoken languages". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 16. 2022, pp. 43–54 (cit. on p. 2).
 - [12] H. Hassani et al. "Text Mining in Big Data Analytics". In: *Big Data and Cognitive Computing* 4.1 (2020). issn: 2504-2289. doi: [10.3390/bdcc4010001](https://doi.org/10.3390/bdcc4010001). url: <https://www.mdpi.com/2504-2289/4/1/1> (cit. on p. 4).
 - [13] X. Xie et al. "A novel text mining approach for scholar information extraction from web content in Chinese". In: *Future Generation Computer Systems* 111 (2020), pp. 859–872. issn: 0167-739X. doi: <https://doi.org/10.1016/j.future.2019.08.033>. url: <https://www.sciencedirect.com/science/article/pii/S0167739X19305084> (cit. on p. 4).
 - [14] A. Kumar, V. Dabas, and P. Hooda. "Text classification algorithms for mining unstructured data: a SWOT analysis". In: *International Journal of Information Technology* 12 (Feb. 2018). doi: [10.1007/s41870-017-0072-1](https://doi.org/10.1007/s41870-017-0072-1) (cit. on pp. 4, 9).
 - [15] *What is text mining? A beginner's guide*. url: <https://monkeylearn.com/text-mining/> (visited on 01/01/2022) (cit. on p. 4).
 - [16] D. Antons et al. "The application of text mining methods in innovation research: current state, evolution patterns, and development priorities". In: *R & D Management* 50 (2020), pp. 329–351 (cit. on p. 5).
 - [17] *Text mining: Text mining*. url: <https://subjectguides.library.american.edu/TextMining> (visited on 01/01/2022) (cit. on p. 5).
 - [18] P. Rajapaksha, R. Farahbakhsh, and N. Crespi. "Scrutinizing news media cooperation in facebook and twitter". In: *IEEE Access* (2021), pp. 1–14 (cit. on p. 5).
 - [19] E. Kotzé, B. A. Senekal, and W. Daelemans. "Automatic classification of social media reports on violent incidents in South Africa using machine learning". In: *South African Journal of Science* 116.3-4 (2020), pp. 1–8 (cit. on p. 5).
 - [20] A. S. Halibas, A. S. Shaffi, and M. A. K. V. Mohamed. "Application of text classification and clustering of Twitter data for business analytics". In: *2018 Majan international conference (MIC)*. IEEE. 2018, pp. 1–7 (cit. on pp. 5, 6, 9).
 - [21] A. Dhar et al. "Text categorization: past and present". In: *Artificial Intelligence Review* 54 (Apr. 2021), pp. 1–48. doi: [10.1007/s10462-020-09919-1](https://doi.org/10.1007/s10462-020-09919-1) (cit. on pp. 5, 8, 9, 11, 12).
 - [22] C. Zong, R. Xia, and J. Zhang. *Text Data Mining*. Springer, 2021 (cit. on p. 5).

- [23] Q. Li et al. "A survey on text classification: From shallow to deep learning". In: *arXiv preprint arXiv:2008.00364* (2020) (cit. on pp. 5, 10–13, 21).
- [24] A. Wani et al. "Evaluating deep learning approaches for covid19 fake news detection". In: *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing EmERGE ncy Si tuation*. Springer. 2021, pp. 153–163 (cit. on pp. 5, 9, 14, 15, 20).
- [25] M. Müller, M. Salathé, and P. E. Kummervold. "Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter". In: *arXiv preprint arXiv:2005.07503* (2020) (cit. on pp. 5, 15).
- [26] C. Shorten, T. M. Khoshgoftaar, and B. Furht. "Deep Learning applications for COVID-19". In: *Journal of big Data* 8.1 (2021), pp. 1–54 (cit. on p. 5).
- [27] J. A. Leite et al. "Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis". In: *arXiv preprint arXiv:2010.04543* (2020) (cit. on pp. 6, 17, 20).
- [28] S. E. Jordan et al. "Using Twitter for public health surveillance from monitoring and prediction to public response". In: *Data* 4.1 (2019), p. 6 (cit. on p. 6).
- [29] M. Habdank, N. Rodehuts Kors, and R. Koch. "Relevancy assessment of tweets using supervised learning techniques: Mining emergency related tweets for automated relevancy classification". In: *2017 4th International conference on information and communication technologies for disaster management (ICT-DM)*. IEEE. 2017, pp. 1–8 (cit. on p. 6).
- [30] A. Olteanu et al. "Crisislex: A lexicon for collecting and filtering microblogged communications in crises". In: *Eighth international AAAI conference on weblogs and social media*. 2014 (cit. on p. 6).
- [31] Y. Aphinyanaphongs et al. "Text classification for automatic detection of alcohol use-related tweets: A feasibility study". In: *Proceedings of the 2014 IEEE 15th international conference on information reuse and integration (IEEE IRI 2014)*. IEEE. 2014, pp. 93–97 (cit. on pp. 6, 9).
- [32] A. A. Khaleq and I. Ra. "Twitter analytics for disaster relevance and disaster phase discovery". In: *Proceedings of the Future Technologies Conference*. Springer. 2018, pp. 401–417 (cit. on p. 6).
- [33] A. Suarez et al. "A data collection for evaluating the retrieval of related tweets to news articles". In: *European Conference on Information Retrieval*. Springer. 2018, pp. 780–786 (cit. on p. 6).
- [34] V. Behzadan et al. "Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 2018, pp. 5002–5007 (cit. on p. 6).
- [35] A. Alessa, M. Faezipour, and Z. Alhassan. "Text classification of flu-related tweets using fasttext with sentiment and keyword features". In: *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2018, pp. 366–367 (cit. on p. 6).

- [36] S. Urchs et al. "MMoveT15: A Twitter Dataset for Extracting and Analysing Migration-Movement Data of the European Migration Crisis 2015". In: *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*. IEEE. 2019, pp. 146–149 (cit. on p. 6).
- [37] O. Edo-Osagie et al. "Twitter mining using semi-supervised classification for relevance filtering in syndromic surveillance". In: *PloS one* 14.7 (2019), e0210689 (cit. on p. 6).
- [38] Y. Matalon et al. "Using sentiment analysis to predict opinion inversion in Tweets of political communication". In: *Scientific reports* 11.1 (2021), pp. 1–9 (cit. on p. 6).
- [39] J. You, J. Lee, and H.-Y. Kwon. "A complete and fast scraping method for collecting tweets". In: *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE. 2021, pp. 24–27 (cit. on p. 6).
- [40] F. Gurcan and N. E. Cagiltay. "Research trends on distance learning: a text mining-based literature review from 2008 to 2018". In: *Interactive Learning Environments* 0.0 (2020), pp. 1–22. doi: [10.1080/10494820.2020.1815795](https://doi.org/10.1080/10494820.2020.1815795) (cit. on p. 7).
- [41] D. Tao, P. Yang, and H. Feng. "Utilization of text mining as a big data analysis tool for food science and nutrition". In: *Comprehensive Reviews in Food Science and Food Safety* 19.2 (2020), pp. 875–894. doi: <https://doi.org/10.1111/1541-4337.12540>. eprint: <https://ift.onlinelibrary.wiley.com/doi/pdf/10.1111/1541-4337.12540>. url: <https://ift.onlinelibrary.wiley.com/doi/abs/10.1111/1541-4337.12540> (cit. on p. 7).
- [42] U. Naseem, I. Razzak, and P. W. Eklund. "A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter". In: *Multimedia Tools and Applications* 80.28 (2021), pp. 35239–35266 (cit. on pp. 7, 21, 36).
- [43] D. N. de Oliveira and L. H. de Campos Merschmann. "Joint evaluation of preprocessing tasks with classifiers for sentiment analysis in Brazilian Portuguese language". In: *Multimedia Tools and Applications* 80.10 (2021), pp. 15391–15412 (cit. on pp. 7, 19).
- [44] S. M. *Feature Extraction and Embeddings in Natural Language Processing*. <https://www.analyticsvidhya.com/blog/2021/07/feature-extraction-and-embeddings-in-nlp-a-beginners-guide-to-understand-natural-language-processing/>. (Accessed on 01/01/2022). July 2021 (cit. on p. 8).
- [45] I. Yahav, O. Shehory, and D. Schwartz. "Comments Mining With TF-IDF: The Inherent Bias and Its Removal". In: *IEEE Transactions on Knowledge and Data Engineering* 31.3 (2019), pp. 437–450. doi: [10.1109/TKDE.2018.2840127](https://doi.org/10.1109/TKDE.2018.2840127) (cit. on p. 8).
- [46] J. Chen, Y. Tao, and H. Lin. "Visual exploration and comparison of word embeddings". In: *Journal of Visual Languages & Computing* 48 (2018), pp. 178–186 (cit. on p. 8).

- [47] A. Sawczyn et al. "Fact-checking: relevance assessment of references in the Polish political domain". In: *Procedia Computer Science* 192 (2021). Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021, pp. 1285–1293. issn: 1877-0509. doi: <https://doi.org/10.1016/j.procs.2021.08.132>. url: <https://www.sciencedirect.com/science/article/pii/S1877050921016215> (cit. on p. 8).
- [48] D. Raihan. *Deep learning techniques for text classification*. Apr. 2021. url: <https://towardsdatascience.com/deep-learning-techniques-for-text-classification-78d9dc40bf7c> (cit. on p. 8).
- [49] A. Elnagar, R. Al-Debsi, and O. Einea. "Arabic text classification using deep learning models". In: *Information Processing & Management* 57.1 (2020), p. 102121. issn: 0306-4573. doi: <https://doi.org/10.1016/j.ipm.2019.102121>. url: <https://www.sciencedirect.com/science/article/pii/S0306457319303413> (cit. on pp. 8, 12, 13, 17).
- [50] V. Gaikwad and Y. Haribhakta. "Adaptive GloVe and FastText model for hindi word embeddings". In: *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*. 2020, pp. 175–179 (cit. on p. 8).
- [51] B. Jang et al. "Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism". In: *Applied Sciences* 10.17 (2020), p. 5841 (cit. on pp. 9, 13, 14).
- [52] K. Kowsari et al. "Text Classification Algorithms: A Survey". In: *Information* 10.4 (2019). issn: 2078-2489. doi: [10.3390/info10040150](https://doi.org/10.3390/info10040150). url: <https://www.mdpi.com/2078-2489/10/4/150> (cit. on p. 9).
- [53] P. Li et al. "Bag-of-Concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base". In: *Knowledge-Based Systems* 193 (2020), p. 105436. issn: 0950-7051. doi: <https://doi.org/10.1016/j.knosys.2019.105436>. url: <https://www.sciencedirect.com/science/article/pii/S0950705119306604> (cit. on p. 9).
- [54] K. Wu et al. "A fuzzy logic-based text classification method for social media data". In: *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2017, pp. 1942–1947 (cit. on p. 9).
- [55] Z. Zhang et al. "A deep learning approach for detecting traffic accidents from social media data". In: *Transportation research part C: emerging technologies* 86 (2018), pp. 580–596 (cit. on p. 9).
- [56] Y. Huang et al. "Hierarchical multi-attention networks for document classification". In: *International Journal of Machine Learning and Cybernetics* 12 (June 2021). doi: [10.1007/s13042-020-01260-x](https://doi.org/10.1007/s13042-020-01260-x) (cit. on pp. 9, 10).

- [57] J. X. Koh and T. M. Liew. "How loneliness is talked about in social media during COVID-19 pandemic: Text mining of 4,492 Twitter feeds". In: *Journal of Psychiatric Research* (2020). issn: 0022-3956. doi: <https://doi.org/10.1016/j.jpsychires.2020.11.015>. url: <https://www.sciencedirect.com/science/article/pii/S0022395620310748> (cit. on p. 10).
- [58] A. Hotho, A. Nürnberger, and G. Paass. "A Brief Survey of Text Mining". In: *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology* 20 (Jan. 2005), pp. 19–62 (cit. on p. 10).
- [59] S. A. Curiskis et al. "An evaluation of document clustering and topic modelling in two online social networks: Twitter and Reddit". In: *Information Processing & Management* 57.2 (2020), p. 102034 (cit. on p. 10).
- [60] P. Otero, J. Gago, and P. Quintas. "Twitter data analysis to assess the interest of citizens on the impact of marine plastic pollution". In: *Marine Pollution Bulletin* 170 (2021), p. 112620 (cit. on pp. 10, 17).
- [61] W. Wang et al. "Twin labeled LDA: a supervised topic model for document classification". In: *Applied Intelligence* 50.12 (2020), pp. 4602–4615 (cit. on p. 10).
- [62] J. P. Carvalho, H. Rosa, and F. Batista. "Detecting relevant tweets in very large tweet collections: The London Riots case study". In: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2017, pp. 1–6 (cit. on p. 10).
- [63] J. P. Carvalho et al. "MISNIS: An intelligent platform for twitter topic mining". In: *Expert Systems with Applications* 89 (2017), pp. 374–388 (cit. on pp. 10, 20, 51).
- [64] J. T. *Text Categorization: Approaches*. 2019. doi: [10.1007/978-3-319-91815-0_6](https://doi.org/10.1007/978-3-319-91815-0_6) (cit. on p. 10).
- [65] X. Luo. "Efficient english text classification using selected machine learning techniques". In: *Alexandria Engineering Journal* 60.3 (2021), pp. 3401–3409 (cit. on pp. 10, 11).
- [66] A. I. Kadhim. "Survey on supervised machine learning techniques for automatic text classification". In: *Artificial Intelligence Review* 52.1 (2019), pp. 273–292 (cit. on p. 11).
- [67] H. T. Sueno, B. D. Gerardo, and R. P. Medina. "Multi-class document classification using support vector machine (SVM) based on improved Naïve bayes vectorization technique". In: *International Journal of Advanced Trends in Computer Science and Engineering* 9.3 (2020) (cit. on p. 11).
- [68] A. Moldagulova and R. B. Sulaiman. "Using KNN algorithm for classification of textual documents". In: *2017 8th International Conference on Information Technology (ICIT)*. IEEE. 2017, pp. 665–671 (cit. on p. 11).
- [69] S. Yilmaz and S. Toklu. "A deep learning analysis on question classification task using Word2vec representations". In: *Neural Computing and Applications* (2020), pp. 1–20 (cit. on p. 12).

- [70] S. Minaee et al. "Deep Learning–Based Text Classification: A Comprehensive Review". In: *ACM Comput. Surv.* 54.3 (Apr. 2021). issn: 0360-0300. doi: [10 . 1145 / 3439726](https://doi.org/10.1145/3439726). url: [https : //doi .org/10.1145/3439726](https://doi.org/10.1145/3439726) (cit. on pp. 12, 13).
- [71] M.-Y. Cheng, D. Kusoemo, and R. A. Gosno. "Text mining-based construction site accident classification using hybrid supervised machine learning". In: *Automation in Construction* 118 (2020), p. 103265. issn: 0926-5805. doi: <https://doi.org/10.1016/j.autcon.2020.103265>. url: <https://www.sciencedirect.com/science/article/pii/S092658051931341X> (cit. on p. 12).
- [72] B. Jiménez Gutiérrez et al. "Document Classification for COVID-19 Literature". In: *arXiv e-prints*, arXiv:2006.13816 (June 2020), arXiv:2006.13816. arXiv: [2006.13816 \[cs.IR\]](https://arxiv.org/abs/2006.13816) (cit. on p. 12).
- [73] V. Kumar et al. "A Comprehensive Analysis of Deep Learning Techniques for Documentation Classification". In: *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*. 2021, pp. 228–235. doi: [10 . 1109 / ICAIS50930 . 2021 . 9395861](https://doi.org/10.1109/ICAIS50930.2021.9395861) (cit. on pp. 12, 15, 16, 20).
- [74] F. E. Ayo et al. "Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions". In: *Computer Science Review* 38 (2020), p. 100311 (cit. on pp. 12, 19).
- [75] K. Winter and R. Kern. "Know-center at SemEval-2019 task 5: multilingual hate speech detection on Twitter using CNNs". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 431–435 (cit. on p. 12).
- [76] A. Ribeiro and N. Silva. "INF-HatEval at SemEval-2019 Task 5: Convolutional neural networks for hate speech detection against women and immigrants on twitter". In: *Proceedings of the 13th International Workshop on Semantic Evaluation*. 2019, pp. 420–425 (cit. on p. 12).
- [77] Y. Zhang and B. Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification". In: *arXiv preprint arXiv:1510.03820* (2015) (cit. on p. 12).
- [78] M. Tezgider, B. Yildiz, and G. Aydin. "Text classification using improved bidirectional transformer". In: *Concurrency and Computation: Practice and Experience* n/a.n/a (), e6486. doi: [https : //doi .org/10.1002/cpe.6486](https://doi.org/10.1002/cpe.6486). eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.6486>. url: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.6486> (cit. on p. 13).
- [79] *A Beginner's Guide to Attention Mechanisms and Memory Networks | Pathmind*. (Visited on 01/01/2022) (cit. on p. 13).
- [80] A. Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008 (cit. on p. 14).

-
- [81] F. Souza, R. Nogueira, and R. Lotufo. “BERTimbau: pretrained BERT models for Brazilian Portuguese”. In: *Brazilian Conference on Intelligent Systems*. Springer. 2020, pp. 403–417 (cit. on pp. 14, 15).
 - [82] G. Wiedemann, S. M. Yimam, and C. Biemann. *UHH-LT at SemEval-2020 Task 12: Fine-Tuning of Pre-Trained Transformer Networks for Offensive Language Detection*. 2020. arXiv: 2004.11493 [cs.CL] (cit. on p. 14).
 - [83] T. D. Salma, G. A. P. Saptawati, and Y. Rusmawati. “Text Classification Using XLNet with Infomap Automatic Labeling Process”. In: *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*. 2021, pp. 1–6. doi: 10.1109/ICAICTA53211.2021.9640255 (cit. on p. 14).
 - [84] J. Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423. url: <https://aclanthology.org/N19-1423> (cit. on p. 14).
 - [85] M. S. Z. Rizvi. *Demystifying BERT: A Comprehensive Guide to the Groundbreaking NLP Framework*. 2019. url: <https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/> (visited on 11/23/2021) (cit. on p. 14).
 - [86] A. Adhikari et al. *DocBERT: BERT for Document Classification*. 2019. arXiv: 1904.08398 [cs.CL] (cit. on p. 14).
 - [87] R. Anggrainingsih, G. M. Hassan, and A. Datta. “BERT based classification system for detecting rumours on Twitter”. In: *arXiv preprint arXiv:2109.02975* (2021) (cit. on pp. 14, 17, 20).
 - [88] B. Lutkevich. *What is Bert (language model) and how does it work?* Jan. 2020. url: <https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model> (cit. on pp. 14, 15).
 - [89] Y. Guo et al. “Benchmarking of Transformer-Based Pre-Trained Models on Social Media Text Classification Datasets”. In: *Proceedings of the The 18th Annual Workshop of the Australasian Language Technology Association*. 2020, pp. 86–91 (cit. on p. 15).
 - [90] D. Kumar, N. Kumar, and S. Mishra. “NLP@NISER: Classification of COVID19 tweets containing symptoms”. In: *Proceedings of the Sixth Social Media Mining for Health (#SMM4H) Workshop and Shared Task*. Mexico City, Mexico: Association for Computational Linguistics, June 2021, pp. 102–104. doi: 10.18653/v1/2021.smm4h-1.19. url: <https://aclanthology.org/2021.smm4h-1.19> (cit. on p. 15).

- [91] S. S. Ahmed and A. Kumar M. "Classification of Censored Tweets in Chinese Language using XLNet". In: *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*. Online: Association for Computational Linguistics, June 2021, pp. 136–139. doi: [10.18653/v1/2021.nlp4if-1.21](https://doi.org/10.18653/v1/2021.nlp4if-1.21). url: <https://aclanthology.org/2021.nlp4if-1.21> (cit. on pp. 15, 17, 20).
- [92] A. Sharma and H. Pandey. "LRG at TREC 2020: Document Ranking with XLNet-Based Models". In: *ArXiv abs/2103.00380* (2020) (cit. on pp. 15, 20).
- [93] P. by Statista Research Department and N. 1. *Twitter global mdau 2021*. Nov. 2021. url: <https://www.statista.com/statistics/970920/monetizable-daily-active-twitter-users-worldwide/> (visited on 01/01/2022) (cit. on p. 16).
- [94] R. Behzadidoost et al. *A framework for text mining on Twitter: a case study on joint comprehensive plan of action (JCPOA)- between 2015 and 2019*. 2021. doi: <https://doi.org/10.1007/s11135-021-01239-y> (cit. on pp. 16, 17, 19).
- [95] A. Karami et al. "Twitter and Research: A Systematic Literature Review Through Text Mining". In: *IEEE Access* 8 (2020), pp. 67698–67717. doi: [10.1109/ACCESS.2020.2983656](https://doi.org/10.1109/ACCESS.2020.2983656) (cit. on pp. 16, 17).
- [96] B. de Sousa Pereira Amorim et al. "Using Supervised Classification to Detect Political Tweets with Political Content". In: *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web* (2018) (cit. on p. 16).
- [97] E. Aramaki, S. Maskawa, and M. Morita. "Twitter catches the flu: detecting influenza epidemics using Twitter". In: *Proceedings of the 2011 Conference on empirical methods in natural language processing*. 2011, pp. 1568–1576 (cit. on p. 17).
- [98] *Social Media Strategy for Twitter - 2018 Research on 100 Million Tweets | Vicinitas*. en. url: <https://www.vicinitas.io/blog/twitter-social-media-strategy-2018-research-100-million-tweets> (visited on 01/01/2022) (cit. on p. 17).
- [99] E. P. Souza et al. "Characterising Text Mining: a Systematic Mapping Study of the Portuguese Language". In: *IET Software* 12 (July 2017). doi: [10.1049/iet-sen.2016.0226](https://doi.org/10.1049/iet-sen.2016.0226) (cit. on p. 19).
- [100] Babbel.com and L. N. GmbH. *The 10 most spoken languages in the world*. url: <https://www.babbel.com/en/magazine/the-10-most-spoken-languages-in-the-world> (visited on 01/01/2022) (cit. on p. 19).
- [101] M. S. Ramalho. "High-level Approaches to Detect Malicious Political Activity on Twitter". In: *arXiv preprint arXiv:2102.04293* (2021) (cit. on pp. 19, 21, 35).
- [102] G. P. M. Paiva et al. "COVID 19: O que sentem os brasileiros de acordo com o Twitter?" In: *Journal of Health Informatics* 12 (2021) (cit. on p. 19).

-
- [103] P. Fortuna et al. “A hierarchically-labeled portuguese hate speech dataset”. In: *Proceedings of the Third Workshop on Abusive Language Online*. 2019, pp. 94–104 (cit. on p. 19).
 - [104] F. Souza, R. Nogueira, and R. Lotufo. “Portuguese named entity recognition using BERT-CRF”. In: *arXiv preprint arXiv:1909.10649* (2019) (cit. on p. 19).
 - [105] M. Stilpen Junior and L. H. C. Merschmann. “A methodology to handle social media posts in brazilian portuguese for text mining applications”. In: *Proceedings of the 22nd Brazilian Symposium on Multimedia and the Web*. 2016, pp. 239–246 (cit. on p. 19).
 - [106] E. Souza et al. “Characterising text mining: a systematic mapping review of the Portuguese language”. In: *IET Software* 12.2 (2018), pp. 49–75 (cit. on p. 19).
 - [107] S. Piscitelli, E. Arnaudo, and C. Rossi. “Multilingual Text Classification from Twitter during Emergencies”. In: *2021 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2021, pp. 1–6 (cit. on p. 19).
 - [108] D. Cirqueira et al. “A literature review in preprocessing for sentiment analysis for Brazilian Portuguese social media”. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE. 2018, pp. 746–749 (cit. on p. 21).
 - [109] W. Hersh. “Evaluation of biomedical text-mining systems: lessons learned from information retrieval”. In: *Briefings in bioinformatics* 6.4 (2005), pp. 344–356 (cit. on p. 21).
 - [110] C. Baziotis, N. Pelekis, and C. Doulkeridis. “DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis”. In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 747–754 (cit. on p. 36).
 - [111] J. Ferreira, H. Gonalo Oliveira, and R. Rodrigues. “Improving NLTK for Processing Portuguese”. In: *Symposium on Languages, Applications and Technologies (SLATE 2019)*. In press. June 2019 (cit. on p. 37).
 - [112] M. Imran, U. Qazi, and F. Ofli. “TBCOV: Two Billion Multilingual COVID-19 Tweets with Sentiment, Entity, Geo, and Gender Labels”. In: *Data* 7.1 (2022). issn: 2306-5729. doi: [10.3390/data7010008](https://doi.org/10.3390/data7010008). url: <https://www.mdpi.com/2306-5729/7/1/8> (cit. on pp. 42, 44).
 - [113] E. Chen and E. Ferrara. *Tweets in Time of Conflict: A Public Dataset Tracking the Twitter Discourse on the War Between Ukraine and Russia*. 2022. doi: [10.48550/ARXIV.2203.07488](https://doi.org/10.48550/ARXIV.2203.07488). url: <https://arxiv.org/abs/2203.07488> (cit. on pp. 42, 47).
 - [114] L. Marujo et al. “BP2EP-adaptation of Brazilian Portuguese texts to European Portuguese”. In: *Proceedings of the 15th Annual conference of the European Association for Machine Translation*. 2011 (cit. on p. 42).

- [115] T. de Melo and C. M. Figueiredo. "A first public dataset from Brazilian twitter and news on COVID-19 in Portuguese". In: *Data in Brief* 32 (2020), p. 106179. issn: 2352-3409. doi: <https://doi.org/10.1016/j.dib.2020.106179>. url: <https://www.sciencedirect.com/science/article/pii/S2352340920310738> (cit. on p. 42).
- [116] U. Qazi, M. Imran, and F. Ofli. "GeoCoV19: A Dataset of Hundreds of Millions of Multilingual COVID-19 Tweets with Location Information". In: *SIGSPATIAL Special* 12.1 (June 2020), pp. 6–15. doi: [10.1145/3404111.3404114](https://doi.org/10.1145/3404111.3404114). url: <https://doi.org/10.1145/3404111.3404114> (cit. on p. 44).
- [117] N. S. Agarwal, N. S. Pun, and S. K. Sonbhadra. "Exploring Public Opinion Dynamics on the Verge of World War III using Russia-Ukraine war-Tweets Dataset". In: (2022) (cit. on p. 47).



COVID-19 Portuguese Twitter Accounts

- [19_portugal](#)
- [actamedport](#)
- [ciencia_pt](#)
- [covidometroPT](#)
- [COVID19PTBot](#)
- [covid19Portugal](#)
- [DGSaude](#)
- [euronewspt](#)
- [govpt](#)
- [gripenet_pt](#)
- [guiadasaudept](#)
- [inemtwtitting](#)
- [infarmed_ip](#)
- [irj_pt](#)
- [itwtitting](#)
- [jornalpoligrafo](#)
- [lusa_noticias](#)
- [medscapept](#)
- [mentesaudavelmz](#)
- [observadorpt](#)
- [onuportugal](#)
- [saude_pt](#)
- [sns_portugal](#)
- [solonline](#)
- [sos_covid19](#)
- [theblindspot8](#)
- [vacinacaocovid1](#)
- [vacinacaoCOVID](#)
- [visao_pt](#)
- [vostaz](#)
- [vostpt](#)
- [voaportugues](#)

COVID-19 Dataset - Seed For Extraction

```
{
  "db_name": "covid-19_pt_br",
  "seed_hashtags": {
    "mandatory": [],
    "optional": ["#vaificartudobem", "#omicron", "#vacinasocovid",
      "#aquinaocovid", "#distanciasocial", "#covid", "#wuhantviro",
      "#wuhantpneumonia", "#chinacoronavir", "#coronovirus", "#coronoviruschina",
      "#coronovirus", "#covid", "#covid19", "#vacinacovid", "#virochines",
      "#sarscov2", "#vacinacaocovid", "#covid19brasil", "#sars-cov-2",
      "#coronovirus", "#pandemia", "#quarentena", "#fiqueemcasa", "#2019ncov",
      "#ncov2019", "#coronovirusindia", "#sars", "#coronoavirus",
      "#coronaravirus", "#coronoviruswuhant", "#covid19", "#coronovirusportugal",
      "#coronovirusbrasil"]
  },
  "seed_keywords": {
    "mandatory": [],
    "optional": ["covid19", "covid", "coronavirus", "pandemia", "'vacina
covid'", "'viro chines'"]
  },
  "since": "2019-12-01",
  "until": "2022-08-20",
  "seed_users": {
    "mandatory": [],
    "optional": []
  },
  "lang": "pt"
}
```

Figure 26: Extraction seed for extracting COVID-19 dataset using the Twitter Extractor

Russian Invasion of Ukraine Dataset - Seed For Extraction

```
{
  "db_name": "covid-19_pt_br",
  "seed_hashtags": {
    "mandatory": [],
    "optional": ["#vaificartudobem", "#omicron", "#vacinasocovid",
      "#aquinaocovid", "#distanciasocial", "#covd", "#wuhanvirus",
      "#wuhanpneumonia", "#chinacoronavirus", "#coronavirus", "#coronaviruschina",
      "#coronavirus", "#covid", "#covid19", "#vacinacovid", "#viruschines",
      "#sarscov2", "#vacinacaocovid", "#covid19brasil", "#sars-cov-2",
      "#coronavirus", "#pandemia", "#quarentena", "#fiqueemcasa", "#2019ncov",
      "#ncov2019", "#coronavirusindia", "#sars", "#coronoavirus",
      "#coronaravirus", "#coronaviruswuhan", "#covid19", "#coronavirusportugal",
      "#coronavirusbrasil"]
  },
  "seed_keywords": {
    "mandatory": [],
    "optional": ["covid19", "covid", "coronavirus", "pandemia", "'vacina covid'", "'virus chines'"]
  },
  "since": "2019-12-01",
  "until": "2022-08-20",
  "seed_users": {
    "mandatory": [],
    "optional": []
  },
  "lang": "pt"
}
```

Figure 27: Extraction seed for extracting Russian Invasion of Ukraine dataset using the Twitter Extractor