TÉCNICO LISBOA

Collection and classification of telecom-related fast-text news and events

Artur Francisco Filipe Simões

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. João Paulo Baptista de Carvalho
Eng. Cristina João Pires

Examination Committee

Chairperson: Prof. Pedro Filipe Zeferino Aidos Tomás
Supervisor: Prof. João Paulo Baptista de Carvalho
Member of the Committee: Prof. Fernando Manuel Marques Batista

July 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

I would like to thank my parents and family for their encouragement and caring over all these years, for always being there for me through thick and thin, and without whom this project would not be possible.

I would also like to thank Carlota for her understanding, support, friendship, and caring.

To all my friends and colleagues Gaspar, Tavares, Isi, Malcata, Botto, and Edu that helped me grow as a person and were always there for me during the good and bad times for the past 7 years together. Thank you.

I would also like to acknowledge my dissertation supervisors Professor João Carvalho and Cristina for their insight, support, and sharing of knowledge that has made this Thesis possible.

Abstract

The quality of Telecom companies' mobile networks can be seriously compromised by the occurrence of different types of events, whether they are expected or not. Operations Support Systems (OSS) team helps to manage the resilience and performance of the network, including Service Provisioning, Problem Management, and Quality of Service processes. Collecting probes, diagnoses, and alarms might be useful to detect the causes of the network problem. But what might have been the external cause? The cause of the cause? The purpose of the project is to automatically identify and classify events contained in short text news and civil protection occurrences from their official websites, as soon as they are known, in order to correlate them with network issues reported from the OSS team alarms. From this perspective, this thesis created an architecture that periodically collects information from the aforementioned events, in order to process it and, then, classify them using the following frameworks *Docker*, *Maven*, *Quarkus* and *Kafka*. Finally, data is stored using *PostgreSQL*. We also perform an exhaustive comparison between different lightweight models for text classification using information collected from several Portuguese online newspapers: Support Vector Machines (SVM), Fuzzy Fingerprints (FFP), and K-Nearest Neighbours (KNN). More complex deep models, such as Bidirectional Encoder Representations from Transformers (BERT) or Robustly Optimized BERT Pretraining Approach (RoBERTa), are dismissed due to the requirement of large amounts of training data. The proposed models predict the categories based entirely on the title and the short news snippets that are freely available. The initial findings show F1 scores exceeding 0.68 for every class, despite the dataset being unbalanced with around 14,000 samples and 4 classes.

Keywords

Text classification; Short Texts classification; Support Vector Machines; Fuzzy Fingerprints; K-nearest

neighbors; Operation Support Systems, Telecommunications

Resumo

A qualidade da rede fornecida pelas operadoras de telecomunicações é fortemente afetada pela ocorrência de diferentes tipos de eventos externos, quer estes sejam esperados ou não. A equipa de Sistema de Suporte às Operações (SSO) monotoriza o desempenho e resiliência da rede, através de provisionamento de serviços, gestão de problemas e monotorização da qualidade de serviço dos processos. A coleta de amostras, diagnósticos e alarmes permite detetar causas de falha na rede, como uma antena que deixou de emitir sinal. Mas qual poderá ter sido a causa externa? A causa da causa? O objetivo deste trabalho é idenfificar e classificar automaticamente eventos presentes em textos de notícias e ocorrências da proteção civil através dos seus sites oficiais, assim que estes estejam disponíveis, para que sejam correlacionados com problemas da rede reportados pela equipa de SSO. O desenvolvimento de uma arquitetura que coleta periodicamente a informação contida nos eventos acima referidos para que seja, posteriormente, processada, classificada usando as frameworks *Docker*, *Maven*, *Quarkus* e *Kafka* e armazenada em base de dados usando *PostgreSQL*. A escolha do classificador de texto é feita através de uma comparação exaustiva de diferentes classificadores usando as notícias portuguesas coletadas: Support Vector Machines (SVM), Fuzzy Fingerprints (FFP)), and K-Nearest Neighbours (KNN). Modelos mais complexos, como Bidirectional Encoder Representations from Transformers (BERT) ou Robustly Optimized BERT Pretraining Approach (RoBERTa), não foram considerados porque exige um vasto data set de treino. Os modelos propostos classificam notícias com base no título e breve descrição disponíveis gratuitamente. Os primeiros resultados indicam F1-scores acima de 0.68 para cada uma das classes, considerando um conjunto de dados desequilibrado, com 14000 amostras e 4 classes.

Palavras Chave

Classificação de texto; Support Vector Machines; Fuzzy Fingerprints; K-nearest neighbors; Sistemas

de Suporte ás Operações

Contents

List of Figures

List of Tables

Listings

Acronyms

ANACOM	Autoridade Nacional de Comunicações
ANCP	Autoridade Nacional de Proteção Civil
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
BOW	Bag of Words
FFP	Fuzzy Fingerprints
IPMU	Information Processing and Management of Uncertainty in Knowledge-Based Systems
JAR	Java ARchive
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbours
LSTM	Long Short-term Memory
ML	Machine Learning
NLP	Natural Language Processing
NB	Naïve Bayes
NER	Named Entity Recognition
NN	Neural Networks
OSS	Operations Support Systems
PK	Primary Key
POM	Project Object Model
RF	Random Forests
RDBMS	Relational Databases Management System
REST	Representational State Transfer
RF	Random Forests

RoBERTa	Robustly Optimized BERT Pretraining Approach
RSS	Really Simple Syndication
SADO	Sistema de Apoio à Decisão Operacional
SVM	Support Vector Machines
SQL	Structured Query Language
TC	Text Classification
TF-IDF	Term frequency Inverse Document Frequency
XML	Extensible Markup Language

1

Introduction

Contents

Chapter 1 gives a foretaste of the scientific and business relevance of this work developed at Altice Labs, SA in collaboration with Instituto Superior Técnico. This chapter starts by motivating the problem addressed in this thesis from a general approach (Subsection 1.1) to a more specific (Subsection 1.3) architecture perspective. The remaining sections are dedicated to data sources analysis and thesis organization.

1.1 Motivation

Telecommunications service providers use Operations Support Systems (OSS) teams to develop computer systems to manage their networks. They support management and alarm functions such as network inventory, service provisioning, network configuration, and fault management. Therefore, OSS receives large amounts of alarms when monitoring events and detecting anomalous situations. Besides those alarms, there are also external factors to this monitoring system that could also be important to prevent or mitigate issues when providing telecommunications services. Namely, customer complaints about the network on social media, any type of event that jeopardizes the network service in a specific zone (such as a huge public gathering), and any type of occurrences that might compromise the good functioning of the equipment (like fires or storms). By signaling and filtering these relevant external events, the motorization of the network would be richer and would contribute to a faster fault resolution. In order to get customer perceptions of service quality and experience about the network, it is necessary to identify specific channels from which they would express their satisfaction or dissatisfaction and try to identify relevant external events. Gunarathne, P., Rui, H. et al [5] point out the importance of monitoring costumers complaints on social media. Empowered by the popularization of social media and mobile technologies, consumers nowadays easily distribute their complaints to brands publicly in real time, expanding the boundaries of traditional customer service. Such a public approach may work out for consumers in the digital age, rather than spending hours on the phone to contact the brand's dedicated customer service. As a result, more and more customers are turning to social media platforms such as Twitter and Facebook to express their complaints about brands on social media. In response, companies are striving to monitor and respond to their customers, before complaints go viral and cause reputational damage to the organization. Previous studies indicate that failures themselves do not necessarily lead to customer dissatisfaction, since most customers accept that things may sometimes go wrong [6]. Instead, the service provider's response or lack of response to the failure is the most likely cause of dissatisfaction [7]. Davidow [8] examined how customers assess the organizational responses to complaints and the impact of those assessments on future consumer behavior. They find attentiveness as the most important organizational response dimension, affecting both word-of-mouth activity and repurchase intentions. Telecom companies are especially exposed to this effect as they are re-

sponsible for television, internet, and mobile communications - something that people do not tolerate failing.

When collecting external events that can have an impact on the network service, there are some considered unique to the Internet, besides social media customer complaints, that are easily collected. Really Simple Syndication (RSS) feeds, news aggregators, news blogs, news wikis [9], public forums, and newspapers on social media. Studies suggest that the shift towards online news sources stems from the attraction of the Internet as a news medium [1] has happened anywhere in developed countries.

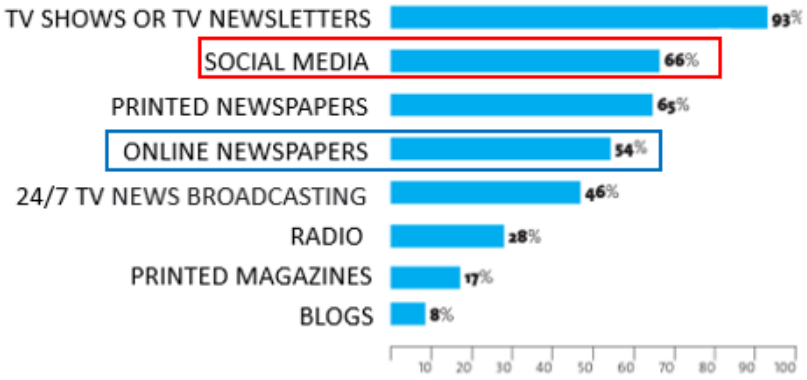


Figure 1.1: Percentages of news sources consumed in Portugal. Social media and newspaper websites, referred to in the discussion, are highlighted in red and grey, respectively. The survey was conducted in 2015 and this figure was extracted from [1]

.

Access to abundant information has become the key advantage of online news media. In addition to news supplied by mainstream news organizations, people also have online access to alternative news and views from social media. Online news media has some important attributes considering the goal of this project, such as the 24/7 delivery of breaking news (news immediacy), the free access to news snippets, and the capacity to find information faster than using offline media. In addition, online news stories can be “linked” to more in-depth information, giving news and information a detailed and richer historical background.

All these studies provide important insights into customer complaint management procedures and their consequences to organizations. They also enhance that the digital era has raised customer expectations about the range and quality of services delivered by any company. Traditional networks and customer experience management are no longer enough to keep your customers satisfied. Today, you need to manage service quality proactively, and improve troubleshooting, so that you can prevent any customer issues before (if possible) they even arise.

1.2 Problem Formulation

In this section, we compare data sources considered relevant to detect events that compromise the network service, along with the main challenges encountered by the company in that specific context. Later we analyze data information and ways to collect, extract and classify it using Natural Language Processing (NLP) and Machine Learning (ML).

1.2.1 Data sources

Before signaling specific sources of information to the context of this project, some characteristics are required to analyze and compare data sources available in Portugal: social media, online newspapers, and other report issues that could jeopardize the network service. Data sources containing the characteristics in table 1.1 are considered relevant to this project:

Table 1.1: Important data sources characteristics for this project

Free access - data retrieving is free
Unlimited access - all information should be collectible
Real-time data - data must be instantly available
Representative - must be used by a large number of people
Traceable - events must have a location

1.2.1.A Social Media

Figure 1.2 shows the most visited mobile social media websites in Portugal in March 2022. They represent a great opportunity for data collection. However, each of them has different characteristics and limitations.

From all social media platforms in Figure 1.2, only Twitter has all the characteristics mentioned in Table 1.1 but after a first analysis two problems were identified:

- Even though it is used by a large number of people, the amount of customer complaints mentioning the company service provider is not enough to produce consistent results.
- Using the free Twitter Application Programming Interface (API), only 1% of the total amount of tweets are collectable.

Secondly, it was necessary to establish the first problem analyzed on Twitter - are there complaints expressed on social media scaled enough to produce relevant information? For this purpose, we created a survey asking what kind of channels customers typically use when complaining about the network service (social media or operator customer support).

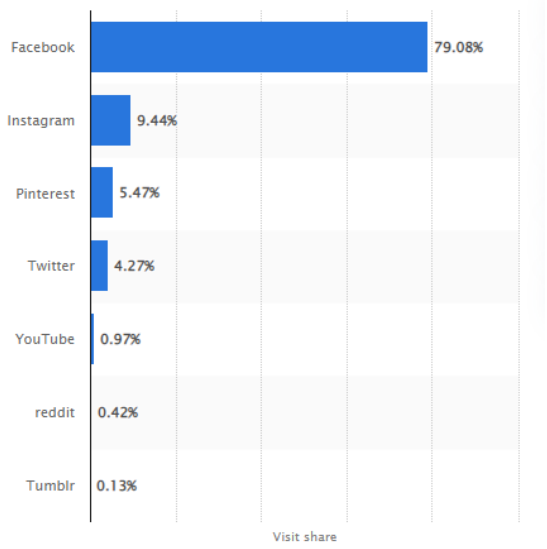


Figure 1.2: Leading mobile social media websites in Portugal in March 2022, based on the share of visits. This figure was extracted from [2]

.

Our survey received a total of 266 responses from individuals across Portugal, the majority from the center of Portugal (75%). 45% of the respondents were younger than 39 years old, 47.4% of the respondents were between the ages of 40-59 and 7.5% were older than 59. From that population, 50.4% identified as female, 49.2% identified as male, and 0.4% preferred not to say. Each sample record corresponds to one person, and reporting units were selected to correspond to all the interviewees. The goal of the electronic survey is to collect data from a portion of a population regarding its behavior when complaining about network performance. When usually decides to complain, which kind of channels typically use, which is the preferred network provider, and which kind of channel would use to complain about specific problems (WiFi connection, mobile telecommunication, mobile data, fixed network)?

The results from the survey are very clear, in figure 1.3, customer dissatisfaction is not expressed on social media scaled enough to produce relevant information.

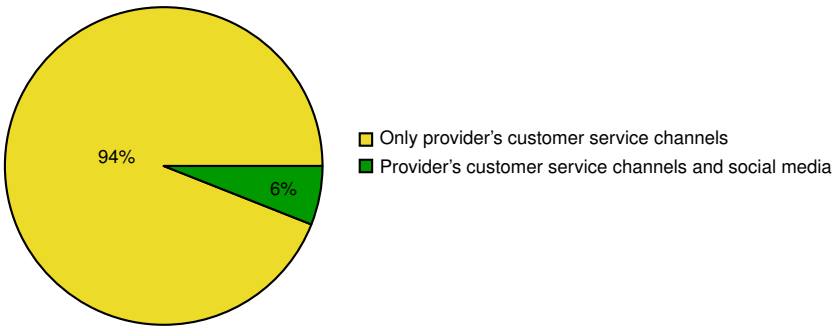


Figure 1.3: "What type of channels do you typically use when complaining about your network service"? n= 266 (sample)

1.2.1.B Websites of Costumers Complaints

Livro de Reclamações online [10] is the Portuguese Official book of complaints. It is a Portuguese website where customers can express their complaints about a brand or a service. Ideally, after a complaint, the problem should be solved in a friendly way between the customer and the company. If that does not happen, an appointing authority would analyze and judge the case. This website has an extremely important role when complaining, however, its information is not accessible nor collectible.

Portal da Queixa [11] is a Portuguese website where customers can express their complaints about a brand. The website does not intervene in the troubleshooting between the customers and the brand. When a complaint is validated the brand is notified via email. Although in some specific cases, this website information could help, it is not collectible and the general complaint is not technical.

Official MEO Costumer Portal [12] issues are similar to Portal da Queixa. In this portal, customers can express their complaints about the company's services. However, they are not collectible due to the company's policies and the general complaint is not technical but related to bureaucracy and invoicing.

Downdetector [13] collects posts from Twitter and reports submitted on their websites and mobile apps. The system validates and analyzes these reports in real time, allowing for automatically detecting outages and service disruptions. The website's downsides are: data status reports from Twitter have exactly the same Twitter API limitations (discussed at the beginning of this subsection) and the amount of customers' complaints is only relevant regarding national wide problems or failures in big cities, therefore it is not representative in most cases.

1.2.1.C Online newspapers APIs

Most Portuguese newspapers offer an online version where on average the latest 15 daily news are available. Each news contains several attributes as shown in figure 1.4. Being the title, description, link, and publication date the most important ones.

```
▼<item>
   <title>Matosinhos. Incêndio urbano fere uma mulher</title>
   <description>Fogo começou no primeiro andar de um prédio com sete andares.</description>
   <enclosure url="https://images.rr.sapo.pt/01051443f62_base.jpg" length="40665" type="image/jpeg"/>
   <pubDate>Wed, 22 Jun 2022 07:55:40 GMT+1</pubDate>
   <link>https://rr.sapo.pt/noticia/pais/2022/06/22/matosinhos-incendio-urbano-fere-uma-mulher/289236/?utm_medium=rss</link>
   <guid isPermaLink="false">822a5748-faf1-ec11-b47a-281878139228</guid>
 </item>
```

Figure 1.4: Example of a news snippet body in Extensible Markup Language (XML). This figure was extracted from `https://www.rtp.pt/noticias/rss/pais`

.

Some news APIs collectors already exist taking advantage of these worldwide newspapers' online versions but they come with limitations. For example, `https://newsapi.ai/` and `newsapi.org`, when running on free mode, limit the number of requests per day and for that reason, these collectors are not

relevant data sources. However, the news is still a strong source of data, so the guideline is to build a collector that consumes the maximum Portuguese online newspapers' news in table 1.2.

Table 1.2: Available Portuguese online newspapers' feeds to extract news

Online newspaper	URL
Diário de Notcias	http://feeds.dn.pt/DN-Ultimas
Jornal de Notícias	http://feeds.jn.pt/JN-Nacional
Público	https://www.publico.pt/api/list/ultimas
Rádio e Televisão de Portugal	https://www.rtp.pt/noticias/rss/pais
TSF - Rádio Notícias	http://feeds.tsf.pt/TSF-Portugal
Rádio Renascença	https://rr.sapo.pt/rss/rssfeed.aspx
Correio da Manhã	https://www.cmjornal.pt/portugal

In this work, online news is used as a tool to detect possible telecom network problems: analyzing their location and classifying using NLP and ML guarantees every characteristic in 1.1. One of the goals of this project is, based on the nature of some relevant news categories, to identify events that could compromise the quality of service of a network provider. Each piece of news should be labeled with one category from a set of possible different ones that may vary during the experimental analysis.

1.2.1.D Portuguese civil protection reports

Autoridade Nacional de Proteção Civil (ANCP), the Portuguese official civil protection authority provides public reports from all its operational data on the official website since 2007. It is possible to analyze all the occurrences close to real time on Portugal's mainland. The reported occurrences result from warnings from different sources. From private calls to the national emergency number, calls to firefighters' headquarters, etc. Every mobilization that results from warnings is reported in the portal Sistema de Apoio à Decisão Operacional (SADO), instantly. Analyzing Figure 1.5 we can notice that every characteristic in 1.1 is respected.

DataOcorrencia	Natureza	EstadoOcorre	Distrito	Concelho	Freguesia	Localidade	Latitude	Longitude	NumeroMeios
22/06/2022 13:15	Riscos Tecnológicos / Acidentes / Colisão rodoviária	Despacho	FARO	LOULÉ	Almancil	ALMANSIL	37,086336	-8,028466	0
22/06/2022 13:14	Protecção e Assistência a Pessoas e Bens / Assistência e Preve	Despacho de 1º	BRAGA	CELORICO DE BASTO	Veade, Gagos e Molares	VEADE, GAGOS E MOLARES	41,4194089	-7,991275342	1
22/06/2022 13:13	Riscos Mistos / Incêndios Rurais / Mato	Despacho de 1º	SANTARÉM	ALCANENA	Minde	MINDE	39,511095	-8,691237	5
22/06/2022 13:11	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Despacho de 1º	BRAGA	VILA NOVA DE FAM	Vermoim	Vermoim	41,42160203	-8,451027457	1
22/06/2022 13:11	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Em Curso	GUARDA	SEIA	Seia, São Romão e Lapa dos	Av. Luis Vaz Camões EDF. Jardim 1 - Seia	40,39931742	-7,686215787	1
22/06/2022 13:11	Protecção e Assistência a Pessoas e Bens / Assistência e Preve	Despacho de 1º	SANTARÉM	SANTARÉM	Azoia de Cima e Tremês	TREMÊS	39,358492	-8,766593	1
22/06/2022 13:10	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Despacho de 1º	BRAGA	ESPOSENDE	Fonte Boa e Rio Tinto	FONTE BOA E RIO TINTO	41,50009075	-8,730779861	1
22/06/2022 13:10	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Em Curso	PORTALEGRE	GAVIÃO	Belver	Vale Pedro Dias	39,51483359	-7,93068306	1
22/06/2022 13:07	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Em Curso	PORTO	AMARANTE	Aboadela, Sanche e Várzea	Aboadela	41,28055642	-7,972611397	1
22/06/2022 13:05	Riscos Tecnológicos / Incêndios Urbanos ou em Área Urbaniza	Em Resolução	LISBOA	LISBOA	Marvila	Marvila	38,739104	-9,11071	1
22/06/2022 13:05	Riscos Tecnológicos / Acidentes / Despiste	Despacho de 1º	LISBOA	LISBOA	Avenidas Novas	Avenidas Novas	38,73746	-9,152329	2
22/06/2022 13:04	Protecção e Assistência a Pessoas e Bens / Assistência em Saú	Em Curso	LISBOA	LOURES	Sacavém e Prior Velho	Sacavem	38,79276261	-9,114283738	0
22/06/2022 13:02	Riscos Tecnológicos / Incêndios em Transportes / Rodoviário	Em Resolução	BRAGANÇA	TORRE DE MONCOR	Adeganha e Cardanha	ADEGANHA	41,276943	-7,050589	3
22/06/2022 13:01	Riscos Mistos / Comprometimento total ou parcial de seguran	Despacho de 1º	AVEIRO	AROUCA	Alvarenga	ALVARENGA	40,966861	-8,172332	1
22/06/2022 13:01	Protecção e Assistência a Pessoas e Bens / Assistência e Preve	Em Curso	VISEU	TAROUCA	Tarouca e Dálvares	Tarouca	41,01519432	-7,784095549	1
22/06/2022 13:00	Riscos Tecnológicos / Incêndios Urbanos ou em Área Urbaniza	Conclusão	LISBOA	VILA FRANCA DE XI	Alverca do Ribatejo e Sobra	Sobralinho	38,918068	-9,034761	1
22/06/2022 13:00	Riscos Mistos / Incêndios Rurais / Mato	Despacho de 1º	PORTO	TROFA	Bougado (São Martinho e S	BOUGADO (São Martinho)	41,337087	-8,550072	1

Figure 1.5: Example of occurrences collected from SADO. This figure was extracted from http://www.prociv.pt/pt-pt/SITUACAOOPERACIONAL/Paginas/default.aspx

The portal may contain information from the past few days. That happens because every day dozens of occurrences from all natures are uploaded and some of them take more than one day to solve. For the

sake of this work, only the natures that compromise the proper functioning of the antennas are filtered. Like urban and rural fires, adverse weather conditions, the fall of buildings, trees, etc.

1.3 Objectives

By comparing and analyzing different data sources (Subsection 1.2.1) and considering characteristics in Table 1.1, we decided to focus on occurrences posted by Portuguese civil protection and online newspapers as a telecom-related rich source of information.

Part of this project consists in building a software architecture that collects, in a short period of time, data from the aforementioned sources. Then the data is processed and classified using NLP and ML. Finally, the architecture should store the processed and classified data in a database.

The architecture, fully explained in Chapter 3 of this thesis, will be integrated into Altice's Assurance module to signal events that could jeopardize the quality of the network and try to reduce the complaints number of the company as stated in 1.1.

Every trimester, Autoridade Nacional de Comunicações (ANACOM), publishes customer complaints analysis reported on the Portuguese official Book of Complaints either physical or digital. The report from the first trimester of 2022 [4] stated that there are 27,5 thousand complaints and 70 % of them refer to electronic communications. Another interesting fact is that 63 % of the customers used a digital official Book of complaints to express their issues. Table 1.3 summarizes most complained topics reported to ANACOM mentioning the company network provider - MEO.

Table 1.3: Most complained topics reported to ANACOM mentioning the company network provider in the first trimester of 2022. This data was extracted from [4]

Most complained topics	**Weight (%)**
Long or deficient services repair	15
Long or nonexistent complaint resolution	11
Failures accessing fixed Internet services	11
Failures accessing subscribed TV services	9

This project has two big goals. The first one is to understand which is the best ML model to classify fast text of news snippets. The second one is to signal relevant events using the designed architecture and try to reduce the number of complaints of the first two topics in table 1.3. Although the latest report indicates that MEO has fewer complaints per 1000 clients than its main competitors [4], there is still space for improvement in the Assurance module policies. Even if the events were not identified on time, they are still valuable to evaluate if the problem was local or if it could happen in the surrounding areas.

1.4 Achievements

The Portuguese news database used to train the classifiers was built by us because there was none available online. It is made up of 15000 thousand news and will be available to enrich the research of other colleagues. In addition, we successfully built an architecture that consumes, classifies, and stores the events relevant to the network performance. At this point, it processes on average 200 news snippets and 270 occurrences per day. Although it was not possible to compare our results to company reports, the architecture is integrated in the company's Assurance module and it is being used officially as the company considers this information relevant and useful for analysis. In the sequence of this work, we published our paper called "Fast Text-based Classification of News Snippets for Telecom Assurance" [14], where we expose preliminary results of our classification predictions using K-Nearest Neighbours (KNN), Support Vector Machines (SVM) and Fuzzy Fingerprints (FFP) and later participated in the Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU) 2022 conference from the 11th of July until the 15th in Milan, Italy.

1.5 Organization of the Document

The remaining chapters are organized as follows. Chapter 2 analysis how Text Classification (TC) can be integrated into the telecom assurance module. It highlights the previous work related to TC, major concerns that hamper the adoption of TC solutions in real-world scenarios, the selected classifiers are analyzed and performance evaluation metrics are explained. Also, frameworks used in the architecture are described. Chapter 3 analysis and details the architecture of the presented solution and the step-by-step procedure. Chapter 4 provides a review of the data set. Later the performance of the classifiers and a comparison between them is assessed in parallel. Chapter 5 concludes the thesis, summarizing the findings on TC, and pointing out directions for future improvements.

2

Text Classification in Telecom Assurance

Contents

This section starts by analyzing and comparing the previous studies on Text Classification with a special focus on SVM, KNN, and FFP. Then, evaluation metrics are described. Ends with a brief explanation of some of the frameworks used when building the system architecture.

2.1 Text Classification

For years, a wide range of methods has been applied to analyze the content and information of texts. Text classification is a vital task in NLP that involves categorizing text data into predefined classes or categories. This task has various applications in areas such as information retrieval, sentiment analysis, and spam filtering. Text classification involves building models that learn to automatically classify text based on various features, such as word frequencies, semantic meaning, and contextual information. This process enables the automation of tasks that require sifting through large amounts of unstructured text data, saving time and improving efficiency.

One of the most common and important use cases of text classification is topic classification, which involves assigning a topic or category to a given text. Topic classification is used in various domains, including news categorization, document classification, and topic modeling. In news categorization, for instance, news articles are sorted and organized based on their topics, such as politics, business, sports, and entertainment. Document classification involves categorizing documents, such as emails or legal documents, into different categories based on their topics or purposes. Topic modeling is used to identify topics and subtopics in a corpus of text data, enabling the exploration and analysis of large textual datasets.

Topic Classification problems define a short and generic set of categories, and the documents will often belong to at least one of those categories. The difficulty of a classification task varies across tasks and becomes substantially high as the number of categories or classes exponentially increases. Moreover, in multi-class text classification tasks, an increased number of classes demand larger sets of training data, and some of those classes will be more difficult than others to classify if classes are unbalanced. Reasons for that may be: i) few positive training examples for the class, and/or ii) lack of good predictive features for that class. [15].

Some of the most well-known and commonly applied methods for text classification tasks include: Naïve Bayes (NB) variants, KNN [16] [17] [18], Random Forests (RF) were used develop to a lexical resource used for sentiment analysis and opinion mining tasks [19], Neural Networks (NN) including Long Short-term Memory (LSTM) used by *Google* for speech recognition [20] and generating suggested replies to messages [21], SVM [18] [22] [23]. In this project, we work also with a relatively new classifier FFP that was previously used to identify the author of several documents [24] and to identify Twitter topics [17].

Recent works in text classification are mostly based on very large language models such as Bidirectional Encoder Representations from Transformers (BERT) or its evolutions such as Robustly Optimized BERT Pretraining Approach (RoBERTa). BERT (Bidirectional Encoder Representations from Transformers) is a deep learning algorithm for NLP pre-training developed by Google. BERT is trained on a large corpus of unlabeled text data, which allows the algorithm to learn general-purpose "language understanding" features that can be fine-tuned for a variety of NLP tasks, such as question answering and sentiment analysis. One of the key innovations of BERT is its ability to process words in the context of the entire sentence, rather than just the left or right context, which was a limitation of previous algorithms. This bidirectional training allows BERT to understand the full meaning of a word based on the context in which it is used. BERT has been shown to achieve state-of-the-art results on a wide range of NLP benchmarks and has been widely adopted in the industry for various NLP tasks. The algorithm has been used to improve the relevance of search results on platforms such as Google [25], to improve the understanding of natural language input in conversational systems with chatbots and virtual assistants, making them more efficient in understanding the user's query [26]. Other interesting applications of BERT consist sentiment analysis to classify text into positive, negative, or neutral categories [27], and text classification [25], question answering [28] and name entity recognition [29].

RoBERTa (Robustly Optimized BERT Pre-training Approach) is a variant of the BERT algorithm that was developed by *Facebook AI*. RoBERTa is based on the BERT architecture, but with several modifications that were designed to improve performance even further. RoBERTa's authors found that BERT's pre-training was not fully optimal and proposed several changes to improve its performance. Some of the main differences between RoBERTa and BERT include training on a larger dataset and longer training times. RoBERTa has been shown to achieve state-of-the-art performance on a wide range of natural language understanding tasks, such as text classification [30], question answering [31], and named entity recognition [32].

Even though such models outperform most other simpler techniques in most NLP tasks, in the context of our project, these models may not be the best fit for two key reasons. Firstly, at the beginning of our project, we did not have enough data to perform fine-tuning. These models require a significant amount of data to achieve optimal performance, and training them on a small dataset would likely result in overfitting. Secondly, both BERT and RoBERTa were pre-trained on large English datasets, and fine-tuning them on our Portuguese language data would not necessarily improve their performance on our specific task. Therefore, considering the limited availability of training data and the additional effort required for fine-tuning, it may be more efficient to explore other pre-trained language models that are already trained in Portuguese and require fewer data for fine-tuning or to use simpler classification models.

In what concerns text classification, KNN and the SVM are amongst the most widely used and best-performing text classifiers when lightweight models are a must. In [33], Yang and Liu performed several

tests in a controlled study and reported that SVM and KNN are at least comparable to other well-known classification methods, including NN and NB, and that significantly outperform the other methods when the number of positive training instances per topic is small. When dealing with a text classification problem with a few unbalanced classes and a relatively small dataset, SVM and KNN classifiers may be a better choice than random forests. This is because SVM and KNN have certain advantages over random forests in the following scenarios. SVM can handle unbalanced classes effectively by maximizing the margin between the classes, which reduces the impact of the majority class on the decision boundary. This is particularly useful when the dataset has few instances for one or more of the classes. Secondly, KNN is a simple and intuitive algorithm that can also work well in the presence of unbalanced classes. It can be effective in identifying rare classes when the dataset is small. On the other hand, random forests may not be the best choice for text classification problems with few unbalanced classes and a small dataset. Random forests work best when the dataset is large and diverse, with many instances and features. With a small dataset, random forests may overfit the training data, resulting in poor generalization to new instances.

In this project, we also use an adaptation of the FFP introduced in [24] [34], since they proved to be simple and fast. Previous works on FFP proved that this classifier outperforms SVM and KNN when there are a large number of categories. In [34], an attempt is made to classify Twitter Trending Topics into 18 broad categories, such as sports, politics, technology, etc, and their experiments on a database of randomly selected 768 trending topics (over 18 classes) show that using text-based and using FFP classification modeling, a classification accuracy up to 65% and 70% can be achieved, respectively.

The text contained in each document, independent of its size, is the most relevant source of information for text classification. However, the text is an unstructured form of data that classifiers and learning algorithms cannot directly process [35]. For that reason, our piece of news must be converted into a more manageable form.

A general text analytics framework consists of three consecutive phases: Text Preprocessing, Text Representation, and Knowledge Discovery, shown in Fig.2.1. We use three news titles as an example of Document corpus to illustrate these methods in each step:

- *"Estradas na Serra da Estrela cortadas"*

- *"Fogo volta na serra Estrela"*

- *"Aldeias da serra sem telecomunicações"*

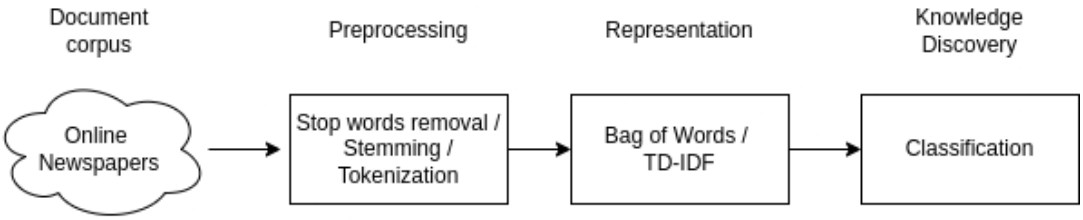


Figure 2.1: General Framework for Text Classification

2.1.1 Preprocessing

Text preprocessing aims to make the input documents more consistent to facilitate text representation, which is necessary for most text classification tasks. Traditional text preprocessing methods include tokenization, stop word removal, stemming, and lemmatization.

Tokenization is the process of breaking down raw text into smaller units called tokens. The goal of tokenization is to segment the text into meaningful units that can be processed by machine learning algorithms. Tokenization involves separating the text into words, punctuation, and other meaningful units such as hashtags, mentions, and URLs. The output of tokenization is a sequence of tokens that can be used as input for further processing. Tokenization can be done using various libraries such as *NLTK*, *spaCy*, or the built-in string library in Python.

Stop word removal eliminates words using a stop word list, in which the words are considered more general and meaningless like pronouns, adverbs, and common verbs. The removal of stop words can help reduce the dimensionality of the data and improve the performance of the model. The process of stop word removal involves creating a list of stop words and removing them from the tokenized text. Those words are retrieved from *Natural Language Toolkit* (NLTK) [36] Python library. The following sentences represent preprocessing with stop words removal:

- *”estradas serra estrela cortadas”*
- *”fogo serra estrela”*
- *”aldeias serra telecomunicações”*

Stemming and lemmatization are techniques used to reduce the inflectional forms of words to their base or root form. The goal of these techniques is to reduce the number of unique words in the text representation and capture the underlying meaning of the words. Stemming involves removing suffixes from words to obtain their root form, while lemmatization involves converting the word into its base form by considering its context in the sentence. For example, the words ”serrando”, ”serras”, and ”serrado” can be stemmed to ”serra”, while lemmatization would convert ”serrado” to ”serra” and ”serrando” to ”serra”. In this project, lemmatization was not used due to poor results regarding Portuguese text. The output of text preprocessing for the three news are:

- *"estrad serr estrel cort"*
- *"fog serr estrel"*
- *"alde serr telecomunic"*

2.1.2 Representation

Text representation in NLP refers to the process of transforming text data into a numerical format that can be easily processed by machine learning algorithms. It is an important step in NLP because most machine learning algorithms require numerical data to perform classification, clustering, and other natural language processing tasks. Text representation helps to convert raw text data into a format that can be easily analyzed and processed by these algorithms.

There are several techniques used for text representation in NLP, including one-hot encoding, bag-of-words, word embeddings, and transformer-based models. Each technique has its own advantages and disadvantages, and the choice of technique depends on the specific NLP task at hand.

2.1.2.A One-hot encoding

One-hot involves representing each word in a text document as a binary vector, where each dimension in the vector corresponds to a unique word in the vocabulary. The vector has a value of 1 in the dimension that corresponds to the word and a value of 0 in all other dimensions.

For example, considering the three preprocessed news in subsection 2.1.1, their one-hot encoding would be:

$$\begin{bmatrix} \textit{estrad} \\ \textit{serr} \\ \textit{estrel} \\ \textit{cort} \\ \textit{fog} \\ \textit{alde} \\ \textit{telecomunic} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

One-hot encoding has the advantage of being a simple and easy-to-understand technique for text representation. It is also very interpretable, as the binary vector for each word corresponds directly to the word in the vocabulary. However, one-hot encoding can be very inefficient for large vocabularies, as the dimensionality of the resulting vector grows with the size of the vocabulary. Additionally, one-hot encoding does not capture any semantic relationships between words, and cannot handle out-of-vocabulary words.

2.1.2.B Bag-of-Words (BOW)

In this technique, a document or piece of text is represented as a bag or collection of its words, without any information about their order or context [23] [34] [37].

The working of BOW can be broken down into the following steps:

1. A vocabulary of unique words is created from all the words in the text.
2. Each word in the vocabulary is assigned a unique index or position.
3. A vector is created for each document, where each key of the vector corresponds to a word in the vocabulary. The value of each key is the count of the number of times the corresponding word appears in the document.

The advantages of BOW are being simple to implement and understand. It can scale easily. It can capture the frequency and distribution of words in a document, which is useful for text classification.

The disadvantages of BOW are it does not capture any information about the order or context of the words in the text. It treats all words as equally important, regardless of their relevance or significance in the text. It can be sensitive to noise or irrelevant words, which can affect the accuracy of the representation.

Using TF-IDF with BOW:

BOW can be used with Term frequency Inverse Document Frequency (TF-IDF) for text representation. [23] [17] [38]. TF-IDF is a normalization technique that adjusts the weights of words in the BOW representation based on their frequency in the document and the frequency of their occurrence in the entire corpus. In TF-IDF, the weight of a word is proportional to the number of times it appears in the document and inversely proportional to the number of documents in the corpus that contain the word. This helps in giving more importance to the words that are more relevant to the document and less importance to the words that are common across the corpus.

Using TF-IDF with BOW has several advantages, such as better representation of documents. TF-IDF helps in better representation of documents by giving more importance to the words that are more relevant to the document. Improved document classification as TF-IDF helps in improving the accuracy of document classification tasks, such as topic modeling and sentiment analysis, by giving more importance to the words that are more indicative of the topic or sentiment [22]. Lastly, reduces the impact of common words that occur frequently across the corpus but are not relevant to the document.

Here is a simple example of how to represent text as a BOW with TF-IDF:

After using the same 3 steps described in the BOW workflow, we end up with a vector created for each document. A feature vector is a numerical representation of the text that captures the frequency of words in the vocabulary. It is common to apply normalization techniques, such as TF-IDF to the features

vectors generated. This normalization method re-weights the count features into floating point values suitable for usage by a classifier.

$$tfidf(w) = tf * idf \tag{2.2}$$

$$idf = \log \frac{N}{df(w)} \tag{2.3}$$

where:

– tf(w) is term frequency (the number of word occurrences in a document) – df(w) is document frequency (the number of documents containing the word)

– N is the number of documents in the corpus

– tfidf(w) is the relative weight of the feature in the vector

As succinctly explained in [39], TF-IDF assigns a weight to a term in a document that is: 1) highest when the term occurs many times within a small number of documents; 2) lower when the term occurs fewer times in a document or occurs in many documents; 3) lowest when the term occurs in virtually all documents, which is the case of stop words, that tend to 0 [40];

Using Bag of Words (BOW) to model the three messages with a TF-IDF weight, the corpus can be represented as a words * documents matrix. Each row represents a word (7 distinct words in total) and each column represents a message, as shown below:

$$\begin{bmatrix} \textit{estrad} \\ \textit{serr} \\ \textit{estrel} \\ \textit{cort} \\ \textit{fog} \\ \textit{alde} \\ \textit{telecomunic} \end{bmatrix} = \begin{bmatrix} 0.477 & 0 & 0 \\ 0 & 0 & 0 \\ 0.176 & 0.176 & 0 \\ 0.477 & 0 & 0 \\ 0 & 0.477 & 0 \\ 0 & 0 & 0.477 \\ 0 & 0 & 0.477 \end{bmatrix} \tag{2.4}$$

Finally, after the feature vector is created, it can be used to represent the text as a BOW. The feature vector can be passed as input to a machine-learning model or used to compute the similarity between different pieces of text.

2.1.2.C Word Embedding

Word Embedding is a text representation technique that maps words to vectors of real numbers in a high-dimensional space. The goal of word embedding is to capture the meaning and context of words in a way that allows them to be compared and manipulated mathematically [41].

The working of Word Embedding can be broken down into the following steps:

1. A large corpus of text is used to train a neural network model, where each word in the text is

treated as a separate input.

2. The model learns to predict the probability of each word in the text given its surrounding words.

3. The weights of the neural network model are used to represent each word as a vector of real numbers in a high-dimensional space.

The advantages of using Word Embedding are capturing the semantic and contextual relationships between words, which can be useful for tasks like natural language understanding, language translation, and text summarization. It can handle out-of-vocabulary words, which are words that are not present in the training data. It can reduce the dimensionality of the text representation, making it easier to process and analyze.

On the other side, the disadvantages of Word Embedding are requiring a large corpus of text to train an accurate model. It can be sensitive to the quality and diversity of the training data, which can affect the accuracy of the representation. It can be computationally expensive to train and use.

Popular Word Embedding techniques like *word2vec* and *word2vec* use different approaches to learning word embeddings from text data. *Word2vec* uses a neural network with a skip-gram or continuous bag-of-words architecture to predict the probability of each word given its context. *GloVe*, on the other hand, uses a co-occurrence matrix to capture the frequency and distribution of words in

2.1.2.D Transformer-based models

Transformer-based models are a type of neural network architecture that uses the self-attention mechanism to process sequential data, such as text. The self-attention mechanism allows the model to focus on different parts of the input sequence at different times, without the need for recurrent connections or convolutional filters. This makes transformer-based models highly parallelizable and computationally efficient. The most popular transformer-based models for NLP include BERT [42], RoBERTa, and GPT-2.

BERT is a pre-trained transformer-based model developed by Google. It is trained on large amounts of text data using a masked language modeling task, which requires the model to predict missing words in a sentence. RoBERTa is a variant of BERT that is pre-trained on even larger amounts of data and uses different training strategies to further improve its performance. GPT-2 (Generative Pre-trained Transformer 2) is a transformer-based language model developed by OpenAI that generates coherent and grammatical text based on a given prompt.

The advantages of transformer-based models include their ability to handle long sequences of text, their high accuracy and state-of-the-art performance on a wide range of NLP tasks, and their ability to transfer knowledge across tasks through pre-training on large amounts of data. Additionally, transformer-based models can be fine-tuned on specific tasks with only a small amount of task-specific data, making

them highly adaptable to new tasks.

While transformer-based models have shown remarkable performance in a wide range of NLP tasks, including text classification, there are also some disadvantages to using them for text classification. Transformer-based models are computationally intensive and require high-end hardware, making them difficult to deploy on low-end devices or in low-resource settings. They can have a large memory footprint, which can be a problem when working with large datasets or in memory-constrained environments. Transformer-based models are often considered ”black boxes” due to their complex architecture, which can make it difficult to understand how they make predictions. They can be prone to overfitting if they are trained on small datasets or if the model architecture is not optimized properly. So, transformer-based models require large amounts of training data to achieve state-of-the-art performance, which can be a challenge in low-resource settings.

2.1.2.E Comparison of different text representation techniques

One-hot encoding, BOW, word embedding, and transformer-based models have all commonly used techniques for representing text in NLP. The suitability of each technique depends on the specific NLP task. One-hot encoding and BOW are suitable for simpler tasks like text classification or sentiment analysis, while word embedding and transformer-based models are more suitable for more complex tasks like language modeling, question answering, and machine translation. The choice of technique also depends on the size of the dataset and the computational resources available for training and inference.

Regarding the text representation techniques the BOW with TF-IDF model has several advantages when compared to other text representation techniques, such as one-hot encoding, word embeddings, and transformer-based models:

- **Simplicity**: The BOW model is simple to understand and implement. It represents text as a ”bag” of its words, disregarding grammar and even word order but keeping track of the frequency of words.

- **Sparsity**: BOW models are not as sparse as one-hot encoding, as they keep track of the frequency of words, this means the vectors produced by BOW tend to be dense, which is generally better when working with large vocabularies.

- **Understanding the frequency of word**: BOW models give an idea of the importance of a word in the text. Since it is the representation is based on the frequency of words, it can give an understanding of how much a word occurs in a text which can be used for some specific NLP tasks, for example, analyzing the topic of a text.

- **Handling OOV words**: BOW model can handle OOV(out of vocabulary) words efficiently, since it is based on counting, it can handle unseen words or words that are not present in the vocabulary.

It is important to note that the BOW representation discards much of the context and meaning of

the words and is not suitable for tasks that require an understanding of the meaning of the text like text generation, Language Translation, etc. This type of approach assumes that words are independent, and do not consider the context where the word was used, losing the syntactic structure and semantic meaning of the sentence. For certain tasks, such as text classification and information retrieval, the bag-of-words representation can be a good choice due to its simplicity and efficiency. However, for other tasks, such as sentiment analysis and machine translation, it might not be the best choice, and other methods like word embeddings or transformer-based models can be more suitable.

In the context of this project containing a relatively small dataset, aims to classify news, and it is important to keep track of the frequency of words, the model BOW seems the most suitable.

2.1.3 Knowledge Discovery:

When we successfully transform the text corpus into numeric vectors, we can apply the existing machine learning or data mining methods like classification or clustering.

When using SVMs the similarity measure is called kernel trick to split data, KNNs uses Euclidean distance to get the *k* nearest neighbors, and FFPs computes a similarity score.

2.1.3.A Support Vector Machine

SVMs are commonly used for Topic Classification tasks [22] [23] [43]. This classification is based on a set of hyperplanes in a high-dimensional space. In a *p* dimensions space the hyperplanes are characterized by $p-1$ dimensions. equation 2.5 describes the hyperplanes defined in Figure 2.2.

$$H(\vec{x}) = \vec{w}^T * \vec{x} + w_0 \tag{2.5}$$

Figure 2.2, represents three different hyperplanes and functional margins: the hyperplane H_1 does not separate the positive from the negative instances.H_2 does, but it does not guarantee the maximum distance between them. Finally, H_3 offers the necessary solution as *z3* represents the largest functional margin.

A good separation is achieved by the hyperplane that has the largest functional margin, the distance to the nearest training data point of any class (represented by *z*),

$$z = \frac{H(\vec{x})}{|\vec{w}|} \tag{2.6}$$

When building a SVM classifier, the goal is to minimize $|\vec{w}|$, in order to obtain the largest functional margin. This optimization task can be nonlinear, so its general equation, using Lagrange multipliers λ_i, is:

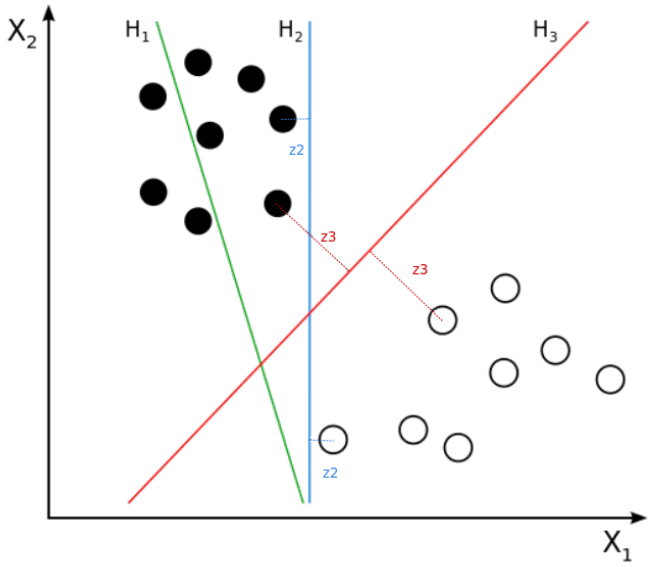


Figure 2.2: 2D Support Vector Machines classifier with three hyperplanes

$$|\vec{w}| = \sum_{i=0}^{N} \lambda_i y_i \vec{x_i} \tag{2.7}$$

In general, a larger margin means a lower classifier generalization error. SVMs can efficiently perform linear and non-linear classifications using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. SVM are also very fast and effective text classifiers when used as binary relevance classifiers. According to [35] "every category has a separate classifier and documents are individually matched against each category".

If it is impossible to find an optimal decision boundary using the aforementioned kernel trick, the SVM algorithm will still try to find a decision boundary that separates the data as well as possible. However, the resulting decision boundary may not be optimal and could potentially misclassify some data points.

In this scenario, the SVM algorithm will try to find a balance between maximizing the margin (i.e., the distance between the decision boundary and the closest data points) and minimizing the number of misclassified data points. This is often achieved by introducing a regularization parameter, which controls the trade-off between these two objectives.

The regularization parameter allows the SVM to accept some misclassifications in exchange for a wider margin, which can help to improve the overall performance of the classifier. However, it is important to note that this approach does not guarantee optimal performance and may still result in misclassifications for some data points.

An interesting property of SVMs is that the decision surface is determined only by support vectors, which are the only effective elements in the training set; if all other points were removed, the algorithm will

learn the same decision function. This characteristic makes SVMs theoretically unique and different from other methods where all the data points in the training set are used to optimize the decision function [33].

2.1.4 k-Nearest Neighbours

KNN is an example-based classifier, commonly referred in the literature [15] [43] [44]. The representations of training data (usually TF-IDF representations) are simply stored together with their category labels. In the figure 2.3, represents a decision whether a new document represented by the yellow dot belongs to a category blue, KNN checks if the k training documents closest to the new document belong to blue. If the answer is positive for a sufficiently large proportion of documents then the result is positive.

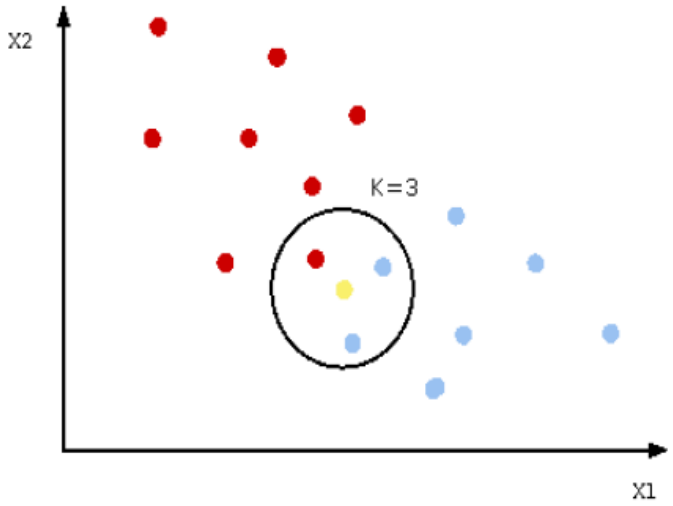


Figure 2.3: 2D k-Nearest Neighbours classifier

An appropriate value of k is of the utmost importance. While k=1 can be too simplistic, as the decision is made according only to the nearest neighbor, a high value of k can lead to too much noise and favor dominant categories. Usually, when testing the value of k, we could start with the square root of the total number of data points. It is also a good practice to choose an odd value of k to avoid ties.

In order to find the k new document's nearest neighbors, the classifier computes the distance between the new document and each one of the training documents using Minkowski Distance, given by the following equation

$$distance = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} \tag{2.8}$$

Minkowski distance is the generalized distance metric. Here generalized means that we can manipulate the above formula to calculate the distance between two data points in different ways. In the

aforementioned example, p=2, then the distance correspond to Euclidean Distance.
In the context of text classification using KNN with Minkowski distance, each document is represented as a vector of features, where each feature represents a term or word. The dimensionality of the vector is equal to the total number of unique terms across all documents in the corpus. The resulting distance value represents the similarity between the two documents, with smaller distances indicating greater similarity. In text classification using KNN with Minkowski distance, the K nearest neighbors are selected based on their Minkowski distance from the test document, and the majority class among the K neighbors is assigned to the test document as its predicted class.

KNN is known to be affected by noisy data, still, it is considered one of the simplest and best-performing text classifiers, whose main drawback is the relatively high computational cost of classification, because, for each test document, it must compute its similarity to all of the training documents. The training is fast, but the classification is slow because it implies computing all the similarities between a document that has not been categorized and the existing collection of training documents. However, given that in our work we are using short text snippets, KNN should be adequate at least as a baseline while the dataset is in its earliest stages.

2.1.5 Fuzzy Fingerprints

The Fuzzy Fingerprint Classifier is a machine-learning model designed to classify text documents based on their content. In [24], the author applies it to the task of authorship identification, which involves identifying the author of a document based on its content. The model generates ”fuzzy fingerprints” for each author by combining the fingerprints of all their documents. These fingerprints capture the author's distinctive writing style, such as their choice of vocabulary, grammar, and sentence structure.

To classify a new document, the model generates a fuzzy fingerprint for the document and compares it to the fingerprints of each author using a similarity measure. The document is then assigned to the author with the highest similarity score. The results showed that the model was able to correctly identify the author of a document with an accuracy of 57.5%. Overall, the study demonstrates the effectiveness of the Fuzzy Fingerprint Classifier model for authorship identification tasks and highlights its potential applications in areas such as forensic linguistics and plagiarism detection.

In this paper, we suggest using the FFP classification method outlined in [17]. The study extends the Fuzzy Fingerprint Classifier model described in [24] to the task of topic identification in Twitter. The goal is to identify the topic of a tweet based on its content, which is often short and noisy.

The model generates ”fuzzy fingerprints” for each topic by combining the fingerprints of a set of the k-most frequent words in tweets that are representative of the topic and in addition also account for their TF-IDF. To classify a new tweet, the model compares the words present in a tweet and the words present in the fingerprint of each topic using the similarity measure in 2.10. The tweet is then assigned

to the topic with the highest similarity score.

The results showed that the model was able to correctly identify the topic of a tweet with an accuracy of 71%, which outperformed other baseline methods such as Naive Bayes and Support Vector Machines. The study demonstrates the effectiveness of the Fuzzy Fingerprint Classifier model for small text classification.

The approach to data text length is where the two studies diverge most from one another, because our project has the same text length as in [17] we suggest using the FFP classification method. Although specific strategies are used to deal with this difference, the approach taken in both classes is comparable. The model uses two hyper-parameters, *K* and *threshold*. *K* represents the size of the fingerprint, and *threshold* serves as the minimal classification score required for a text to be classified. The influence of *threshold* is described in section 2.1.5.B in greater detail. This is how the algorithm itself operates:

- The classifier is initialized with a list of topics, and a dictionary of empty fingerprints are created for each topic.
- Text data is input into the classifier, which adds the words in the input text to the appropriate fingerprint for the corresponding topic.
- Gather the frequencies of the top-k words in all known texts of each news topic and then computes the TF-IDF for each word
- Build the fuzzy fingerprint for each topic by applying the Pareto distribution on the frequencies of the top-k words
- Predict the news topic as the most similar topic fingerprint between the input text and each topic fingerprint

2.1.5.A Building the fingerprint library

In order to build the fingerprint library, the proposed method goes over the training set, which, in this situation, is news from different topics. For each piece of news, it adds each word in the news to a topic table alongside its counter of occurrences. Only the top-k most frequent words are considered. The main difference between the original method and ours is that due to the small size of each news, its words should be as unique as possible in order to make the fingerprints distinguishable amongst the various topics. Therefore, in addition to counting each word occurrence, we also account for its TF-IDF.

After obtaining the top-k list for a given topic, we take the same approach as the original method and use a custom implementation of the *Pareto* distribution as the membership equation to build the fingerprint equation 2.9. Where *k* is the size of the top-k fingerprint words and *i* represents the membership

index. The constants *a* and *b*, can be used to control the slope of the tail of the distribution, with larger values of those constants resulting in a steeper tail.

$$\mu_{ab}(i) = \begin{cases} 1 - (1-b)\frac{i}{kb}, i < a \\ \frac{a(1-\frac{i-a}{k-a})}{k}, i \geq a \end{cases} \tag{2.9}$$

The fingerprint is a *k* sized bi-dimensional array containing in the first column the list of the top-k words, and in the second column its membership value $\mu_{ab}(i)$. The *Pareto* distribution is often used as a model for the frequency of words in natural language because it has a long tail, which means that a small number of words have a much higher frequency than the majority of words. Where roughly 80 percent of the membership value is assigned to the first 20 percent of elements in the ranking.

2.1.5.B Similarity Score

In the original method, equation, in order to check the authorship of a given document, a fingerprint would be built for the document, and then the document fingerprint would be compared with each fingerprint present in the library. Within the news snippets and Twitter context, such an approach would not work due to the very small number of words contained in one example. Therefore we used the similarity score, developed in [17], Tweet-Topic Similarity Score (T2S2) that tests how much a news snippet fits a given topic. The T2S2 function equation 2.10 provides a normalized value ranging between 0 and 1. In equation 2.10, Φ is the topic fingerprint, *T* is the set of words of the (preprocessed) news snippet, $\mu_\Phi(v)$ is the membership degree of the word *v* in the topic fingerprint, and *j* is the number of features of the news. Essentially, T2S2 divides the sum of the membership values $\mu_\Phi(v)$ of every word *v* that is common between the news and the topic fingerprint, by the sum of the top *j* membership values in $\mu_\Phi(w_i)$ where $w \in \Phi$.

$$T2S2(\Phi, T) = \frac{\sum_v \mu_\Phi(v) : v \in (\Phi \cap T)}{\sum_{i=0}^{j} \mu_\Phi(w_i)} \tag{2.10}$$

Equation 2.10 will tend to 1.0 when most to all features of the news snippet belong to the top words of the fingerprint, and tend to 0.0 when none or very few features of the news snippet belong to the bottom words of the fingerprint. At this point, it is possible to define a value for the hyper-parameter *threshold* from 0 to 1. Meaning that news snippets achieving a similarity score below the threshold value would be considered irrelevant.

2.2 Evaluation metrics

It is important to adopt adequate evaluation frameworks with appropriate and diversified metrics, intended to cover all requirements needed to be integrated in Text Classification. We use confusion matri-

ces to evaluate the performance of each classification algorithm. It is used to determine the number of correct and incorrect predictions made by each model.

The following table shows a sample format of a confusion matrix with *n* classes:

		Predicted topic			
		Class 1	Class 2	…	Class ***n***
True topic	Class 1	x_{11}	x_{12}	…	x_{1n}
	Class 2	x_{21}	x_{22}	…	x_{2n}
	. . .	. . .	. . .	. . .	. . .
	Class ***n***	x_{n1}	x_{n2}	…	x_{nn}

Figure 2.4: Confusion matrix for *n* classes (image adapted from [3])

The number of false negatives (FN), false positives (FP), and true negatives (TN) for each class i will be calculated based on the equations 2.11, 2.12, and 2.13, respectively. The true positives in the system will be obtained through equation 2.14.

$$FN_i = \sum_{j=1, j\neq i}^{n} x_{ij} \tag{2.11}$$

$$FP_i = \sum_{i=1, j\neq i}^{n} x_{ij} \tag{2.12}$$

$$TN_i = \sum_{j=1, j\neq i}^{n} \sum_{k=1, k\neq i}^{n} x_{jk} \tag{2.13}$$

$$TP_i = \sum_{i=1}^{n} x_{ii} \tag{2.14}$$

In binary classification, there are only two classes (e.g., *meteorology* and *others*). If the true topic and also the predicted topic are the same correspond to a TP. FN means the true topic is *meteorology* but the model has predicted it as *others*. FP means the true value is *others* but the model has predicted it as *meteorology*. TN means the actual value and also the predicted values are the same, the true value is *others* and the model predicted it the same way. The confusion matrix for a binary classifier contains four entries. Where, cell x_{11} corresponds to TP, cell x_{12} corresponds to FN, cell x_{21} corresponds to FP and cell x_{22} corresponds to FP.

In multi-class classification, there are more than two classes (e.g., *public gatherings*, *fires*, *meteorology*, *others*). The confusion matrix in this case will have dimensions 4 x 4. Each row of the matrix

represents the instances in a predicted class, while each column represents the instances in an actual class (or vice versa). When using multi-class classifiers, the confusion matrix is slightly more complex, following the format of the image 2.4;

The literature review conducted in the present work shows that the most known evaluation metrics used are Accuracy equation 2.15, Precision equation 2.16, Recall equation 2.17, and F1-Score equation 2.18.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.15}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.16}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.17}$$

$$F1_{score} = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{2.18}$$

By analyzing equation 2.15, we can notice that accuracy when there is a dominant class can be largely biased by a large number of TN. Thus F1 Score might be a better measure to use if we need to seek a balance between precision and recall in the presence of uneven class distribution.

The metrics used to evaluate our classification results were: Precision, Recall, and F1-Measure, computed based on the calculation of a confusion matrix using Figure 2.4 and *sklearn.metrics* Python library.

- **Precision** is a measure of the accuracy of the positive predictions made by the classifier. It is defined as the number of true positive predictions divided by the total number of positive predictions made by the classifier. A high precision means that the classifier is making very few false positive predictions.

- **Recall** is a measure of the ability of the classifier to detect all positive instances. It is defined as the number of true positive predictions divided by the total number of actual positive instances in the data. A high recall means that the classifier is detecting a high percentage of all positive instances.

- **F1-score** is the harmonic mean of precision and recall and provides a balanced evaluation of the performance of the classifier. It takes into account both the false positive and false negative rates.

In the context of the previous classification metrics, micro-averaged and macro-averaged scores are two ways to calculate the overall performance of a model.

Micro-averaged scores are calculated by aggregating the contributions of each instance in the dataset individually and then computing the metric of interest. This means that each instance is given equal weight in the calculation of the final score, regardless of its class.

Macro-averaged scores, on the other hand, are calculated by computing the metric of interest separately for each class, and then taking the average across all classes. This means that each class is given equal weight in the calculation of the final score, regardless of its frequency in the dataset.

The micro-averaged scores (recall, precision, and F1) tend to be dominated by the classifier's performance on common categories, and the macro-averaged scores are more influenced by the performance on rare categories [33]. In order to compute the overall metrics, we consider the macro-averaging version over the micro-averaging. A macro-average computes the metric independently for each category and then takes the average (hence treating all categories equally), the objective of this choice is to give more relevance to small categories.

2.3 Frameworks used in the Architecture

The present section describes the five most important framework characteristics of our architecture. Furthermore, information about their influence on the project can be found in Section 3.1.

2.3.1 Docker and Kubernetes

Docker and Kubernetes are mostly complementary technologies. In a nutshell, Docker is a suite of software development tools for creating, sharing, and running individual containers; Kubernetes is a system for operating containerized applications at scale. Organizations use Kubernetes to automate the deployment and management of containerized applications [45].

Containers act as standardized packaging for microservices with all the needed application codes and requirements inside. In the scope of our architecture, the Classification Module was deployed in a container, alongside *CentOS* operating system, *Python* 3.9.8, and libraries and packages were downloaded to the container. Creating, monitoring, and managing these containers is the domain of Docker. A container can run anywhere, on a local machine or in the cloud. In this case, it runs in the cloud on a virtual machine.

The image of the Classifier explained in detail in 3.1.3, was deployed with all Python files and libraries needed to ensure its proper functioning. In our approach, we decided to deploy the Classifier module in one container because containers are easy to replicate and can auto-scale: expand or contract processing capacities to match user demands.

2.3.2 Maven

Maven is a tool that can be used for building and managing any Java-based project. It is a standard way to build projects, a clear definition of what the project consisted of, an easy way to publish project information, and a way to share Java ARchives (JARs), which are compacted Java classes, across several projects. To attain this goal, Maven deals with several areas of concern: Making the build process easy and shielding developers from many details although it does not eliminate the need to know about the underlying mechanisms. Providing a uniform build style Maven builds a project using its Project Object Model (POM) and a set of plugins. All Maven projects are built similarly. This saves time when navigating many projects. Maven provides useful project information that is in part taken from your POM and in part generated from your project's sources (Change log created directly from source control or unit creating test reports). Providing quality project information and encouraging better development practices [46]. The part of our project coded in Java has been made using Maven. The goal of using this framework as project architecture is to allow a developer to comprehend the complete state of a project in the shortest time. It encourages better development practices that were important when dealing with more than one developer when developing the project.

2.3.3 Quarkus

Traditional Java stacks were engineered for monolithic applications with long startup times and large memory requirements in a world where the cloud, containers, and Kubernetes did not exist. Java frameworks needed to evolve to meet the needs of this new world [47]. Quarkus was created to enable Java developers to create applications for a modern, cloud-native world. Quarkus is a Kubernetes-native Java framework adapted with the most used Java libraries and standards, like the ones used in this project RESTEasy (API management), Hibernate (relational database management from which we can create *Entities* like 3.6 and 3.7) and SmallRye (Kafka management). It was created to make Java the leading platform in Kubernetes and serverless environments while offering developers a framework to address a wider range of distributed application architectures. All the modules but the Classification Module run in these Quarkus packages similar to Kubernetes, also deployed in virtual machines. Partially our project is coded in Java using Eclipse Java Integrated Development Environment. This framework was chosen because it contains a base workspace and an extensible plug-in system for customizing the environment. For example, it is simple to create a Quarkus image or to commit our project to the cloud repository.

2.3.4 Kafka

Kafka was developed by Apache, it consists of a storage layer and a compute layer that combines efficient, real-time data ingestion, streaming data pipelines, and storage across distributed systems. In short, this enables simplified, data streaming between Kafka and external systems, so you can easily manage real-time data and scale within any type of infrastructure [48]. Its architecture is represented in figure 2.5

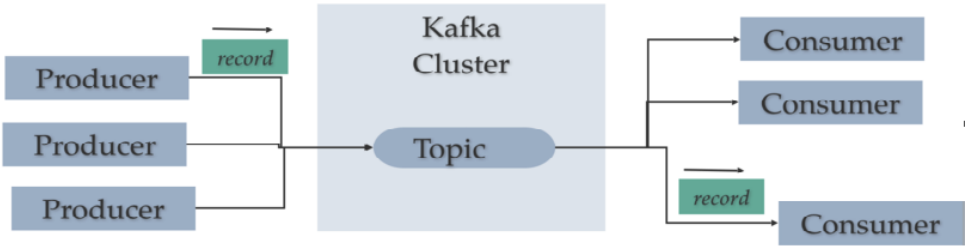


Figure 2.5: Kafka Architecture: Topics, Producers, and Consumers. Image from http://cloudurable.com/blog/kafka-architecture/index.html

Each Kafka data stream is called a Topic. Which is a particular stream of data, like a table in a database (but without all the constraints such as datatype verification). There can exist as many topics as wanted, a topic is identified by its name, is data type agnostic, and a sequence of messages is called a data stream. It is not possible to query topics, instead, Kafka Producers send data and Kafka Consumers read data. The utilization of Kafka within architectural designs offers a level of robustness through its caching functionality, which allows for the retention of data within its memory for a period of one week. This feature enables the ability to access any data that may have been unsuccessful in reaching the database.

The integration of Kafka brings several advantages to this project due to its main characteristics. Kafka is distributed and has resilient architecture. It is fault tolerant and data is kept for a limited time (default is one week), so if data transfer between modules randomly fails it will try to finish it afterward. It has horizontal scalability (can scale to millions of messages per second). We use it as data stream processing between the modules written in different programming languages.

2.3.5 PostgreSQL

PostgreSQL is an open-source relational database that supports both SQL (relational) and JSON (non-relational) querying. It is a highly stable database management system. It is highly scalable both in the quantity of data it can manage and in the number of concurrent users it can accommodate. It includes most SQL:2008 data types, including INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR,

DATE, INTERVAL, and TIMESTAMP. PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications [49]. All databases used in this project were stored on top of this framework.

3

System Architecture

Contents

3.1 Architecture

This architecture was designed to fulfill the following processes: data collection, data processing, and data insertion in the database. It consists of two pipelines, each per data source type, online newspapers news, and civil protection occurrences. These data sources already exist, so no web scraping is needed. After being retrieved, news data need standardization (because data come from different data sources), filtering, text processing, classification, location identification, and storage in the database. The civil protection data pipeline is less demanding than online newspapers. Civil protection data, after being retrieved, go threw filtering, processing, and storage. The goal of this architecture is to enrich a database with news or civil protection occurrences. Those should be well-defined, located, and classified so can be viewed and compared with network metrics and alarms resulting from other companies' products. Our model aims to be fast, robust, and fault-tolerant (due to Kafka implementation).

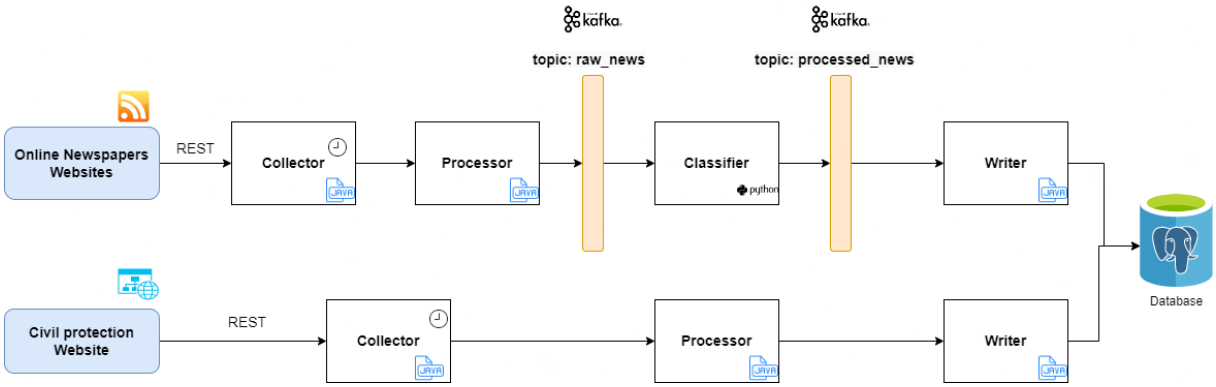


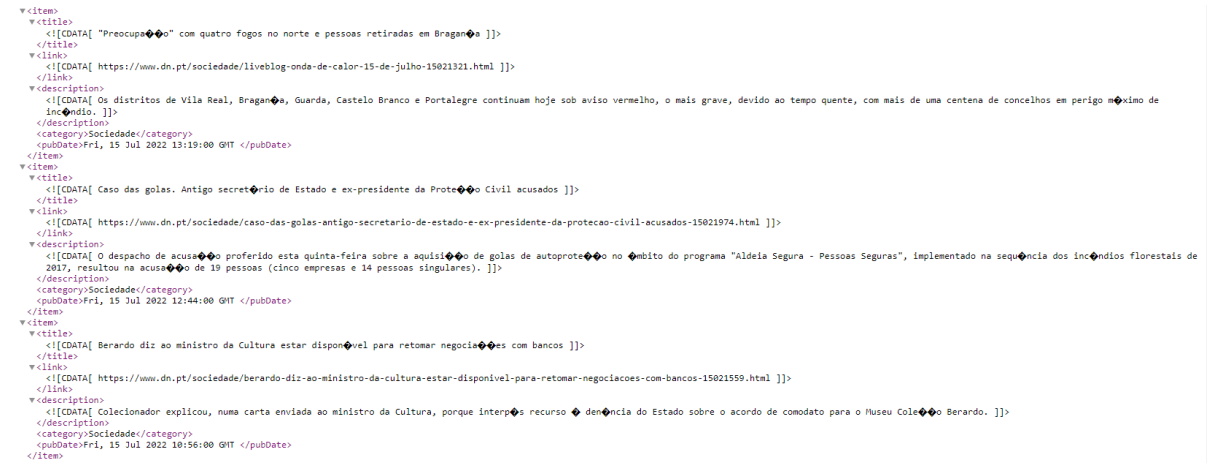
Figure 3.1: Proposed architecture

3.1.1 Collector

3.1.1.A News Collector

The online news collector module collects data from seven different online Portuguese newspapers APIs: Diário de Notícias, Jornal de Noticias, Público, Rádio Renascença, Diário de Notícias, TSF and Correio da Manhã. All sources are real-time sources. Every source but Público and Correio da Manhã, use RSS technology. RSS is a web feed that allows users and applications to have access to updates to websites in a standardized computer-readable format. This method of data extraction is used to get access to approximately the latest 15 news from each online newspaper in XML format, as shown in figure 3.2. Data is retrieved by requesting a Representational State Transfer (REST) GET call every 25 minutes. The maximum collection is 105 news per hour and each news processing time is 10 seconds. So, the maximum collection limit time is around 1050 seconds, which corresponds to 17 minutes and

30 seconds. Even though this is the worst-case scenario we consider a 25 minutes periodic request the perfect trade-off between prevention and maximum information per hour.

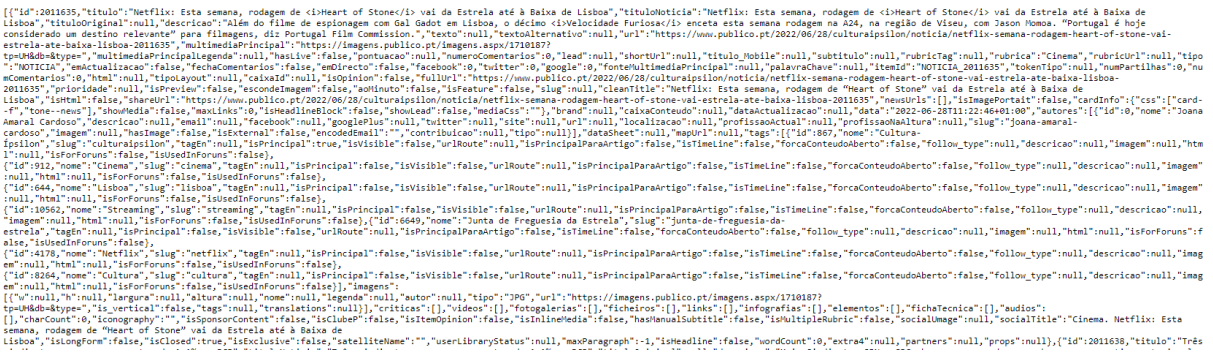


```
<item>
  <title>
    <![CDATA[ "Preocupa��o" com quatro fogos no norte e pessoas retiradas em Bragan�a ]]>
  </title>
  <link>
    <![CDATA[ https://www.dn.pt/sociedade/liveblog-onda-de-calor-15-de-julho-15021321.html ]]>
  </link>
  <description>
    <![CDATA[ Os distritos de Vila Real, Bragan�a, Guarda, Castelo Branco e Portalegre continuam hoje sob aviso vermelho, o mais grave, devido ao tempo quente, com mais de uma centena de concelhos em perigo m�ximo de
    inc�ndio. ]]>
  </description>
  <category>Sociedade</category>
  <pubDate>Fri, 15 Jul 2022 13:19:00 GMT </pubDate>
</item>
<item>
  <title>
    <![CDATA[ Caso das golas. Antigo secret�rio de Estado e ex-presidente da Prote��o Civil acusados ]]>
  </title>
  <link>
    <![CDATA[ https://www.dn.pt/sociedade/caso-das-golas-antigo-secretario-de-estado-e-ex-presidente-da-protecao-civil-acusados-15021974.html ]]>
  </link>
  <description>
    <![CDATA[ O despacho de acusa��o proferido esta quinta-feira sobre a aquisi��o de golas de autoprote��o no �mbito do programa "Aldeia Segura - Pessoas Seguras", implementado na sequ�ncia dos inc�ndios florestais de
    2017, resultou na acusa��o de 19 pessoas (cinco empresas e 14 pessoas singulares). ]]>
  </description>
  <category>Sociedade</category>
  <pubDate>Fri, 15 Jul 2022 12:44:00 GMT </pubDate>
</item>
<item>
  <title>
    <![CDATA[ Berardo diz ao ministro da Cultura estar dispon�vel para retomar negocia��es com bancos ]]>
  </title>
  <link>
    <![CDATA[ https://www.dn.pt/sociedade/berardo-diz-ao-ministro-da-cultura-estar-disponivel-para-retomar-negociacoes-com-bancos-15021559.html ]]>
  </link>
  <description>
    <![CDATA[ Colecionador explicou, numa carta enviada ao ministro da Cultura, porque interp�s recurso  den�ncia do Estado sobre o acordo de comodato para o Museu Cole��o Berardo. ]]>
  </description>
  <category>Sociedade</category>
  <pubDate>Fri, 15 Jul 2022 10:56:00 GMT </pubDate>
</item>
```


Figure 3.2: Example of a news snippet body in XML. This figure was extracted from `http://feeds.dn.pt/DN-Ultimas`

.

Público API GET call, on the other hand, returns a list of JavaScript Object Notation (JSON) objects, as shown in figure 3.3, and compared to news snippets retrieved from RSS it has more and irrelevant attributes.



```
[{"id":2011635,"titulo":"Netflix: Esta semana, rodagem de <i>Heart of Stone</i> vai da Estrela até à Baixa de Lisboa","tituloNoticia":"Netflix: Esta semana, rodagem de <i>Heart of Stone</i> vai da Estrela até à Baixa de
Lisboa","tituloOriginal":null,"descricao":"Além do filme de espionagem com Gal Gadot em Lisboa, o décimo <i>Velocidade Furiosa</i> enceta esta semana rodagem na A24, na região de Viseu, com Jason Momoa. “Portugal é hoje
considerado um destino relevante” para filmagens, diz Portugal Film Commission.","texto":null,"textoAlternativo":null,"url":"https://www.publico.pt/2022/06/28/culturaipsilon/noticia/netflix-semana-rodagem-heart-of-stone-vai-
estrela-ate-baixa-lisboa-2011635","multimediaPrincipal":"https://imagens.publico.pt/imagens.aspx/1710187?
tp=UH&db=&type=","multimediaPrincipalLegenda":null,"hasLive":false,"pontuacao":null,"numeroComentarios":0,"lead":null,"shortUrl":null,"titulo_Mobile":null,"subtitulo":null,"rubricTag":null,"rubrica":"Cinema","rubricUrl":null,"tipo
":"NOTICIA","emActualizacao":false,"fechaComentarios":false,"emDirecto":false,"facebook":0,"twitter":0,"google":0,"fonteMultimediaPrincipal":null,"palavraChave":null,"itemId":"NOTICIA_2011635","tokenTipo":null,"numPartilhas":0,"nu
mComentarios":0,"html":null,"tipoLayout":null,"caixaId":null,"isOpinion":false,"fullUrl":"https://www.publico.pt/2022/06/28/culturaipsilon/noticia/netflix-semana-rodagem-heart-of-stone-vai-estrela-ate-baixa-lisboa-
2011635","prioridade":null,"isPreview":false,"escondeImagem":false,"aoMinuto":false,"isFeature":false,"slug":null,"cleanTitle":"Netflix: Esta semana, rodagem de “Heart of Stone” vai da Estrela até à Baixa de
Lisboa","isHtml":false,"shareUrl":"https://www.publico.pt/2022/06/28/culturaipsilon/noticia/netflix-semana-rodagem-heart-of-stone-vai-estrela-ate-baixa-lisboa-2011635","newsUrls":[],"isImagePortait":false,"cardInfo":{"css":["card-
-f","tone--news"],"showMedia":false,"maxLinks":0,"isHeadlineBlock":false,"showLead":false,"mediaCss":""},"brand":null,"caixaConteudo":null,"dataActualizacao":null,"data":"2022-06-28T11:22:46+01:00","autores":[{"id":0,"nome":"Joana
Amaral Cardoso","descricao":null,"email":null,"facebook":null,"googlePlus":null,"twitter":null,"site":null,"url":null,"localizacao":null,"profissaoActual":null,"profissaoNaAltura":null,"slug":"joana-amaral-
cardoso","imagem":null,"hasImage":false,"isExternal":false,"encodedEmail":"","contribuicao":null,"tipo":null}],"dataSheet":null,"mapUrl":null,"tags":[{"id":867,"nome":"Cultura-
Ipsilon","slug":"culturaipsilon","tagEn":null,"isPrincipal":true,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imagem":null,"htm
l":null,"isForForuns":false,"isUsedInForuns":false},
{"id":912,"nome":"Cinema","slug":"cinema","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imagem"
:null,"html":null,"isForForuns":false,"isUsedInForuns":false},
{"id":644,"nome":"Lisboa","slug":"lisboa","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imagem"
:null,"html":null,"isForForuns":false,"isUsedInForuns":false},
{"id":10562,"nome":"Streaming","slug":"streaming","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,
"imagem":null,"html":null,"isForForuns":false,"isUsedInForuns":false},{"id":6649,"nome":"Junta de Freguesia da Estrela","slug":"junta-de-freguesia-da-
estrela","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imagem":null,"html":null,"isForForuns":f
alse,"isUsedInForuns":false},
{"id":4178,"nome":"Netflix","slug":"netflix","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imag
em":null,"html":null,"isForForuns":false,"isUsedInForuns":false},
{"id":8264,"nome":"Cultura","slug":"cultura","tagEn":null,"isPrincipal":false,"isVisible":false,"urlRoute":null,"isPrincipalParaArtigo":false,"isTimeLine":false,"forcaConteudoAberto":false,"follow_type":null,"descricao":null,"imag
em":null,"html":null,"isForForuns":false,"isUsedInForuns":false}],"imagens":
[{"w":null,"h":null,"largura":null,"altura":null,"nome":null,"legenda":null,"autor":null,"tipo":"JPG","url":"https://imagens.publico.pt/imagens.aspx/1710187?
tp=UH&db=&type=","is_vertical":false,"tags":null,"translations":null}],"criticas":[],"videos":[],"fotogalerias":[],"ficheiros":[],"links":[],"infografias":[],"elementos":[],"ficheTecnica":[],"audios":
[],"charCount":0,"iconography":"","isSponsorContent":false,"isClubeP":false,"isItemOpinion":false,"isInlineMedia":false,"hasManualSubtitle":false,"isMultipleRubric":false,"socialUmage":null,"socialTitle":"Cinema. Netflix: Esta
semana, rodagem de “Heart of Stone” vai da Estrela até à Baixa de
Lisboa","isLongForm":false,"isClosed":true,"isExclusive":false,"satelliteName":"","userLibraryStatus":null,"maxParagraph":-1,"isHeadline":false,"wordCount":0,"extra4":null,"partners":null,"props":null},{"id":2011638,"titulo":"Três
```


Figure 3.3: Example of a news snippet body in JSON. This figure was extracted from `https://www.publico.pt/api/list/ultimas`

.

Correio da Manhã, as Público API, returns a list of JSON objects from a 3rd party file after a GET call. When comparing simple syndication of RSS with API, the last is user dynamic. As an interface that can be programmed, API distribution of content can carry instructions that go beyond merely handing over a file. Those instructions can track how the file is consumed, and send the information back to the source. This specific API returns web scraping content from the website and the limitation of this source is that it is highly unstable due to changes in the website architecture which obliges constant maintenance of the code. However, this data source is extremely important as it is used by some telecom Network

operations center teams. Which is a centralized location where IT teams can continuously monitor the performance and health of a network.

The extracted list of news either in XML or JSON contains a lot of heterogeneous information but only a portion of that information is needed to our goal. Online news raw data examples correspond to figures 3.2 and 3.3. The next step is to filter and process the information.

3.1.1.B Civil protection occurrences Collector

Civil protection website provides data about national wide occurrences. This information (shown in 1.5) can be retrieved using a REST GET call and it is stored as a list of *bytes* for ease of further processing. Data is retrieved requesting a REST GET call every 5 minutes. The average collection is 9 occurrences per hour and each occurrence processing time is approximately instantaneous. The maximum collection limit time could be less than 1 minute, however, we consider that 1 minute is not enough for new data to be available. So we consider a 5 minutes periodic request the perfect trade-off between availability and maximum information per hour.

3.1.2 Processor

This module is programmed using Java. It consumes directly data from the collector module. The following actions depend on the type of data consumed.

3.1.2.A News Processor

Before this stage, raw data has multiple irrelevant attributes for example *enclousure*, *guid*, or *id*, data in this stage is called raw data. Different sources produce, in general, different data attributes as it can be compared between figure 3.2 and figure 3.3. In this module, raw data is filtered and standardized. At the end of this stage, each news snippet element has the following attributes: Title, Description, Link, and Publication date, they are stored in a Java class called *News Data* represented in listing 3.1. Unfortunately using this technology the body of the news is not available to retrieve, so the title and the description are the only used text in our approach.

Listing 3.1: Java class where raw news data is stored

```
1 public class NewsData {
2     private String title;
3     private String description;
4     private String link;
5     private String pubDate;
```

```
6
7       public NewsData(String link, String title, String description, String pubDate) {
8           this.link = link;
9           this.title = title;
10          this.description = description;
11          this.pubDate = pubDate;
12      }
```

After this process, each instance of this Java class is sent to the Classifier module using a Kafka channel (explained in subsection 2.3.4) in JSON serialization format with the topic *raw news*.

3.1.2.B Civil protection occurrences Processor

In this case, due to a significantly less complexity of processing compared to news, the civil protection processor module handles all the processing logic before sending data to the database.

In its raw stage, each occurrence has the following attributes: *Id*, *Date*, *Category*, *Occurrence*, *State*, *District*, *Municipality*, *Parish*, *Local*, *Latitude*, *Longitude*, *Number of firefighters trucks involved*, *Number of firefighters involved*, *Number of air resources involved* and *Number of pilots involved* as we can see in figure 1.5. For further use and easier analysis, each occurrence needs filtering and processing.

Filtering consists of ignoring occurrences of a *Category* that is not considered relevant to the goal of this project. Among others, the relevant categories are Urban fires, Rural fires, Earthquakes, Collapses of buildings, etc. For our platform, relevant occurrences are the ones that can affect antennas' proper functioning.

Processing means converting each occurrence information from *bytes* to *String* and it is mapped to a new Java class object for further use. When creating the object two new attributes are added *Relevance* and *Ops* and some are removed from the original retrieval. The Java class is called *ProcivData* and can be analyzed in listing 3.2.

Listing 3.2: Java class where occurrences data is stored

```
1
2  public class ProcivData {
3      private String id;
4      private Timestamp datetime;
5      private String nature;
6      private String stateOccurrence;
7      private String district;
8      private String municipality;
```

```
9       private String parish;
10      private String local;
11      private Float latitude;
12      private Float longitude;
13      private String opsNumber;
14      private String relevance;
15      }
```

Each municipality has one value of relevance along with its district, latitude, and longitude. These attributes are stored in the table *Relevance* represented in figure 3.4. The *relevance* attribute varies from 1 to 6 and it is proportional to the normalized average of the municipality's total number of inhabitants and the municipality's population density.

The methodology employed involved the normalization of data for each municipality's absolute population and population density. Specifically, we divided both variables by the highest value on their respective lists, resulting in values between 0 and 1 that accurately represent the population in each municipality. To achieve this normalization, we divided the absolute population and population density of each municipality by the highest absolute population and population density values on the list, respectively. We then calculated the average of these normalized values to arrive at a representative population value for each municipality. Furthermore, to facilitate analysis, the resulting list of population values was divided into six equally-sized bins. To perform this task we used a discretization method called frequency-based binning. This division allowed for easier comparison and identification of patterns in the data. Relevance equals 6 is given to less populated regions and on the other hand, relevance equals 1 is given to most populated regions.

municipality	relevance	district	latitude	longitude
Albufeira	2	Faro	37,0889	-8,2511
Ansião	5	Leiria	39,9167	-8,4333
Aguiar da Beira	6	Guarda	40,818	-7,5414
Albergaria-a-Velha	3	Aveiro	40,6936	-8,4806
Alcobaça	2	Leiria	39,5522	-8,9775
Alenquer	2	Lisboa	39,0531	-9,0092
Aljezur	6	Faro	37,3178	-8,8
Almeida	6	Guarda	40,7259	-6,9056
Almodôvar	5	Beja	37,5114	-8,0603
Alvaiázere	6	Leiria	39,8333	-8,3833
Alvito	6	Beja	38,25	-7,9833
Amadora	1	Lisboa	38,75	-9,2333
Amares	4	Braga	41,6333	-8,35
Arganil	5	Coimbra	40,2183	-8,0542
Arouca	3	Aveiro	40,9289	-8,2436
Arronches	6	Portalegre	39,1167	-7,2833
Aveiro	1	Aveiro	40,6389	-8,6553
Avis	6	Portalegre	39,05	-7,8833
Azambuja	3	Lisboa	39,0667	-8,8667
Barcelos	1	Braga	41,5347	-8,615
Batalha	4	Leiria	39,6603	-8,8247
Beja	3	Beja	38,0333	-7,8833
Bombarral	5	Leiria	39,2672	-9,1581
Borba	5	Évora	38,8056	-7,4547
Braga	1	Braga	41,5333	-8,4167

Figure 3.4: Prociv relevance table with *Relevance*, *Latitude*, *Longitude* and *District* by municipality. PostgreSQL system was used and the table was retrieved using DBeaver.

The computation of the average is necessary to prevent less populated municipalities with high density to be considered as relevant as most populated municipalities with lower density. For example, Amadora municipality has 7375,5 inhabitants per km^2 and 171500 total inhabitants, and Lisboa municipality has 6656,8 inhabitants per km^2 and 545923 total inhabitants. Both should be considered equally relevant as we can see in figure 3.5. The data used to compute relevance was retrieved from https://www.pordata.pt/Municipios/Densidade+populacional-452 and https://www.pordata.pt/Municipios.

This value expresses the impact of an occurrence on the network and the customer experience in a municipality. It is assumed that the number of customers is proportional to the number of inhabitants. This means the same occurrence would be associated with a different relevance value depending on where it happened.

municipality	relevance	district	latitude	longitude
Amadora	1	Lisboa	38,75	-9,2333
Lisboa	1	Lisboa	38,7452	-9,1604

Figure 3.5: Prociv relevance table with attributes *Relevance*, *Latitude*, *Longitude* and *District* of Amadora and Lisboa. PostgreSQL system was used and the table was retrieved using DBeaver.

Ops corresponds to the sum of the number of firefighters trucks involved, number of firefighters involved, Number of air resources involved, and Number of pilots involved (the last attributes are not con-

sidered after the processing stage). It is added to reduce the number of attributes and will be important later for visual analysis.

After this processing stage, occurrences from the Civil Protection website represented in the listing 3.2 are sent to the Writer module to be inserted in the database.

3.1.3 Classifier

The classifier module only exists in the news pipeline and its goal is to analyze the news snippets text. This module consumes Kafka streams from the topic *raw news* produced in the module Processor (3.1.2.A). Each of the consumed messages represented in the listing 3.3 from the stream is JSON deserialized, then the news title and the news description are concatenated creating the text to be analyzed. The text analysis should bring the following information: the news category - predicted using the best performer classifier from the classifiers described in section 2 and extract the location or locations where the news refers to.

Listing 3.3: NewsData JSON object from the topic stream *raw news* consumed by the module Classifier

```
raw_news_item={
 "title":"Estradas na Serra da Estrela cortadas devido a queda de neve",
 "description": "Concelhos de Belmonte e Penamacor afetados.",
 "pubDate": "Thu, 03 Mar 2022 11:02:00 Z",
 "link": "https://www.tsf.pt/portugal/sociedade/estradas-na-serra-da-estrela-
     cortadas-devido-a-queda-de-neve-14643913.html"
}
```

The category prediction using SVM, KNN, and FFP was already fully explained in section 2.1.3. The extraction of the new location has two steps: First, search for location names in the text using Named Entity Recognition (NER) applied to Portuguese. This method will help us computationally identify people, places, and things (of various kinds) in a text or collection of texts. Comparing the results to the English example, the author notices that the Portuguese NER is much less good at recognizing entities, and is especially bad at distinguishing different kinds of entities, like *LOC* (Locations) vs *PER* (Persons) [50]. However, in our experiments, the method does a good job when identifying locations in the text.

The available text in the news snippet represented in listing 3.3 corresponds to the concatenation of the fields *Title* and *Description*. Applying NER method we get:

- *Serra da Estrela*
- *Concelhos*

- *Belmonte*
- *Penamacor ´*

Secondly, each *LOC* entity found in the previous step is matched against our database to extract the location information - district, municipality, relevance, and latitude and longitude (of the middle point of the municipality).

- *'municipality': 'Belmonte', 'district': 'Castelo Branco', 'relevance': '6', 'latitude': '40,3583', 'longitude': '-7,3514'*
- *'municipality': 'Penamacor', 'district': 'Castelo Branco', 'relevance': '6', 'latitude': '40,1667', 'longitude': '-7,1667' ´*

If the news refers to two or more locations, a new instance of the news is created with information matching each location. If no location is found in the text, the fields aforementioned appear as "Not found". As *Belmonte* and *Penamacor* municipalities matched against our relevance table (figure), then two processed news instances are created. If the news refers to two or more locations, a new instance of the news is created with information matching each location. If no location is found in the text, the fields aforementioned appear as "Not found". The list of the processed news corresponds to the next listing 3.4.

Listing 3.4: Processed news list example produced by the module Classifier as a result of the news snippet in 3.3 processing

```
processed_news_list=[{'title': 'Estradas na Serra da Estrela cortadas devido
    a queda de neve',
  'description': 'Concelhos de Belmonte e Penamacor afetados.',
  'pubDate': '2022-03-03 11:02:00',
  'link': 'https://www.tsf.pt/portugal/sociedade/estradas-na-serra-da-estrela
     -cortadas-devido-a-queda-de-neve-14643913.html',
  'municipality': 'Belmonte',
  'district': 'Castelo Branco',
  'relevance': '6',
  'lat': '40.3583',
  'lon': '-7.3514',
  'topic': 'meteorologic'},
 {'title': 'Estradas na Serra da Estrela cortadas devido a queda de neve',
  'description': 'Concelhos de Belmonte e Penamacor afetados.',
  'pubDate': '2022-03-03 11:02:00',
```

```
14   'link': 'https://www.tsf.pt/portugal/sociedade/estradas-na-serra-da-estrela
        -cortadas-devido-a-queda-de-neve-14643913.html',
15   'municipality': 'Penamacor',
16   'district': 'Castelo Branco',
17   'relevance': '6',
18   'lat': '40.1667',
19   'lon': '-7.1667',
20   'topic': 'meteorologic'}]
```

Only after going through this process, each message is now sent to the Writer module using a Kafka producer and a new topic called *processed news*.

3.1.4 Writer

3.1.4.A News Writer

This module has a Kafka Consumer implemented that consumes JSON streams with the topic *processed news* produced in the Classifier module. It then creates an instance of a Java class *NewsProcessedData* to store each object from the list consumed by Kafka containing all the attributes present in 3.4.

Although the class in listing 3.5 is almost ready to database insertion, we first need to create a java class called *NewsProcessed* listed in listing 3.6 that represents the table in the database and to import *javax.persistence* library that allows us to do that.

Listing 3.5: Java class where processed news data is stored

```
1  public class NewsProcessedData {
2      private String title;
3      private String description;
4      private String pubDate;
5      private String link;
6      private String district;
7      private String municipality;
8      private String relevance;
9      private String lat;
10     private String lon;
11     private String topic;
12 }
```

A new attribute is added to the previous class called *insertDate* which is important for debugging purposes. Now, we can add the new instances in *news processed* table 3.8 in the database using the *MERGE* statement. The *MERGE* statement in SQL is a very popular clause that can handle inserts, updates, and deletes all in a single transaction without having to write separate logic for each of these. It could happen that news instances to add were already added because of previous Kafka consumption. So if only the *INSERT* statement was used, their unique identifier would clash and throw the error *duplicate key violates unique constraint*.

Listing 3.6: Java class Entity where processed news data is stored

```
1  @IdClass(NewsProcessedId.class)
2  @Table(name ="news_processed")
3  @Entity
4  public class NewsProcessed {
5      @Id
6      private String title;
7      private String description;
8      @Id
9      @Column(name = "pubDate")
10     private Timestamp pubDate;
11     private String link;
12     private String district;
13     @Id
14     private String municipality;
15     private String relevance;
16     private Float lat;
17     private Float lon;
18     private String topic;
19     private Timestamp insertDate;
20     }
```

This module has extreme importance, as it prevents data losses because Kafka stores the streams temporarily before their insertion into the database. Even if the database insertion fails, data would still be available in Kafka's temporary memory by accessing the topic *processed news* manually.

3.1.4.B Civil protection occurrences Writer

Before the insertion in the database, the occurrences represented in the listing 3.2 must be mapped to an Entity Java class called *ProcivOccurrence* that represents the table in the database. This Java class

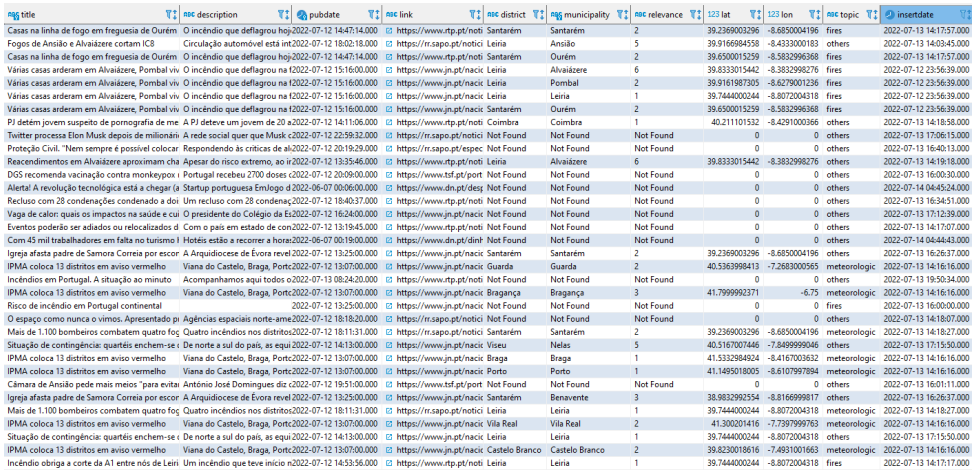
can be analyzed in listing 3.7 and it has one more attribute called *InsertionDate*. This is a *datetime* type attribute that represents the date and time that the occurrence was added to the database, it is helpful for debugging purposes.

Now, we can add create occurrences instances represented by the Entity Java class in the listing 3.7. And then send to *prociv occurrences* table 3.6 in the database using the *MERGE* statement as used in the module news Writer 3.1.4.A.

Listing 3.7: Java class Entity where occurrence data is stored

```
1  @Entity
2  @Table(name = "prociv_occurrences")
3  public class ProcivOccurrence {
4      @Id
5      private String id;
6      private Timestamp datetime;
7      private String nature;
8      private String district;
9      private String municipality;
10     private Float lat;
11     private Float lon;
12     private String ops;
13     private String relevance;
14     private Timestamp insertDate;
15     }
```

The final table can be seen in figure 3.6.



title	description	pubdate	link	district	municipality	relevance	lat	lon	topic	insertdate
Casas na linha de fogo em freguesia de Ourém	O incêndio que deflagrou hoj	2022-07-12 14:47:14.000	https://www.rtp.pt/noti	Santarém	Santarém	2	39.2369003296	-8.6850004196	fires	2022-07-13 14:17:57.000
Fogos de Ansião e Alvaiázere cortam IC8	Circulação automóvel está int	2022-07-12 18:02:18.000	https://rr.sapo.pt/notici	Leiria	Ansião	5	39.9166984558	-8.4333000183	others	2022-07-13 14:03:45.000
Casas na linha de fogo em freguesia de Ourém	O incêndio que deflagrou hoj	2022-07-12 14:47:14.000	https://www.rtp.pt/noti	Santarém	Ourém	2	39.6500015259	-8.5832996368	fires	2022-07-13 14:17:57.000
Várias casas arderam em Alvaiázere, Pombal viv	O incêndio que deflagrou na	2022-07-12 15:16:00.000	https://www.jn.pt/nacio	Leiria	Alvaiázere	6	39.8333015442	-8.3832998276	fires	2022-07-12 23:56:39.000
Várias casas arderam em Alvaiázere, Pombal viv	O incêndio que deflagrou na	2022-07-12 15:16:00.000	https://www.jn.pt/nacio	Leiria	Pombal	2	39.9161987305	-8.6279001236	fires	2022-07-12 23:56:39.000
Várias casas arderam em Alvaiázere, Pombal viv	O incêndio que deflagrou na	2022-07-12 15:16:00.000	https://www.jn.pt/nacio	Leiria	Leiria	1	39.7444000244	-8.8072004318	fires	2022-07-12 23:56:39.000
Várias casas arderam em Alvaiázere, Pombal viv	O incêndio que deflagrou na	2022-07-12 15:16:00.000	https://www.jn.pt/nacio	Santarém	Ourém	2	39.6500015259	-8.5832996368	fires	2022-07-12 23:56:39.000
PJ detém jovem suspeito de pornografia de me	A PJ deteve um jovem de 20 a	2022-07-12 14:11:06.000	https://www.rtp.pt/noti	Coimbra	Coimbra	1	40.211101532	-8.4291000366	others	2022-07-13 14:18:58.000
Twitter processa Elon Musk depois de milionári	A rede social quer que Musk c	2022-07-12 22:59:32.000	https://rr.sapo.pt/notici	Not Found	Not Found	Not Found	0	0	others	2022-07-13 17:06:15.000
Proteção Civil, "Nem sempre é possível colocar	Respondendo às críticas de al	2022-07-12 20:19:29.000	https://rr.sapo.pt/espec	Not Found	Not Found	Not Found	0	0	others	2022-07-13 16:40:13.000
Reacendimentos em Alvaiázere aproximam cha	Apesar do risco extremo, ao ir	2022-07-12 13:35:46.000	https://www.rtp.pt/noti	Leiria	Alvaiázere	6	39.8333015442	-8.3832998276	others	2022-07-13 14:19:18.000
DGS recomenda vacinação contra monkeypox	Portugal recebeu 2700 doses	2022-07-12 20:09:00.000	https://www.tsf.pt/port	Not Found	Not Found	Not Found	0	0	others	2022-07-13 16:00:30.000
Alerta! A revolução tecnológica está a chegar (a	Startup portuguesa Emslogo d	2022-06-07 00:06:00.000	https://www.dn.pt/desp	Not Found	Not Found	Not Found	0	0	others	2022-07-14 04:45:24.000
Recluso com 28 condenações condenado a doi	Um recluso com 28 condenaç	2022-07-12 18:40:37.000	https://www.rtp.pt/noti	Not Found	Not Found	Not Found	0	0	others	2022-07-13 16:34:51.000
Vaga de calor: quais os impactos na saúde e cui	O presidente do Colégio da Es	2022-07-12 16:24:00.000	https://www.jn.pt/nacio	Not Found	Not Found	Not Found	0	0	others	2022-07-13 17:12:39.000
Eventos poderão ser adiados ou relocalizados d	Com o país em estado de con	2022-07-12 13:19:45.000	https://www.rtp.pt/noti	Not Found	Not Found	Not Found	0	0	others	2022-07-13 14:17:07.000
Com 45 mil trabalhadores em falta no turismo I	Hotéis estão a recorrer a hora	2022-06-07 00:19:00.000	https://www.dn.pt/dinh	Not Found	Not Found	Not Found	0	0	others	2022-07-14 04:44:43.000
Igreja afasta padre de Samora Correia por escor	A Arquidiocese de Évora revel	2022-07-12 13:25:00.000	https://www.jn.pt/nacio	Santarém	Santarém	2	39.2369003296	-8.6850004196	others	2022-07-13 16:26:37.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Guarda	Guarda	2	40.5363998413	-7.2683000565	meteorologic	2022-07-13 14:16:16.000
Incêndios em Portugal. A situação ao minuto	Acompanhamos aqui todos o	2022-07-13 08:24:20.000	https://www.rtp.pt/noti	Not Found	Not Found	Not Found	0	0	others	2022-07-13 19:50:34.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Bragança	Bragança	3	41.7999992371	-6.75	meteorologic	2022-07-13 14:16:16.000
Risco de incêndio em Portugal continental		2022-07-12 13:25:00.000	https://www.jn.pt/nacio	Not Found	Not Found	Not Found	0	0	fires	2022-07-13 16:00:00.000
O espaço como nunca o vimos. Apresentado pe	Agências espaciais norte-ame	2022-07-12 18:18:20.000	https://rr.sapo.pt/notici	Not Found	Not Found	Not Found	0	0	others	2022-07-13 14:18:07.000
Mais de 1.100 bombeiros combatem quatro foç	Quatro incêndios nos distritos	2022-07-12 18:11:31.000	https://rr.sapo.pt/notici	Santarém	Santarém	2	39.2369003296	-8.6850004196	meteorologic	2022-07-13 14:18:27.000
Situação de contingência: quartéis enchem-se (	De norte a sul do país, as equi	2022-07-12 14:13:00.000	https://www.jn.pt/nacio	Viseu	Nelas	5	40.5167007446	-7.8499999046	others	2022-07-13 17:15:50.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Braga	Braga	1	41.5332984924	-8.4167003632	meteorologic	2022-07-13 14:16:16.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Porto	Porto	1	41.1495018005	-8.6107997894	meteorologic	2022-07-13 14:16:16.000
Câmara de Ansião pede mais meios "para evita	António Iocé Domingues diz c	2022-07-12 19:51:00.000	https://www.tsf.pt/port	Not Found	Not Found	Not Found	0	0	others	2022-07-13 16:01:11.000
Igreja afasta padre de Samora Correia por escor	A Arquidiocese de Évora revel	2022-07-12 13:25:00.000	https://www.jn.pt/nacio	Santarém	Benavente	3	38.9832992554	-8.8166998817	others	2022-07-13 16:26:37.000
Mais de 1.100 bombeiros combatem quatro foç	Quatro incêndios nos distritos	2022-07-12 18:11:31.000	https://rr.sapo.pt/notici	Leiria	Leiria	1	39.7444000244	-8.8072004318	meteorologic	2022-07-13 14:18:27.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Vila Real	Vila Real	2	41.300201416	-7.7397999763	meteorologic	2022-07-13 14:16:16.000
Situação de contingência: quartéis enchem-se (	De norte a sul do país, as equi	2022-07-12 14:13:00.000	https://www.jn.pt/nacio	Leiria	Leiria	1	39.7444000244	-8.8072004318	others	2022-07-13 17:15:50.000
IPMA coloca 13 distritos em aviso vermelho	Viana do Castelo, Braga, Portc	2022-07-12 13:07:00.000	https://www.jn.pt/nacio	Castelo Branco	Castelo Branco	2	39.8230018616	-7.4931001663	meteorologic	2022-07-13 14:16:16.000
Incêndio obriga a corte da A1 entre nós de Leiri	Um incêndio que teve início n	2022-07-12 14:53:56.000	https://www.rtp.pt/noti	Leiria	Leiria	1	39.7444000244	-8.8072004318	fires	2022-07-13 14:17:17.000

Figure 3.6: Civil protection occurrences table. PostgreSQL system was used and the table was retrieved using DBeaver.

3.1.5 Database

A database is an organized collection of data that can be accessed and manipulated. PostgreSQL is a Relational Databases Management System (RDBMS). RDBMS allows interacting with the database using Structured Query Language (SQL) commands. Although the relational database capabilities were not used in this project, we decided to work with PostgreSQL because it is the most used open-source database in the world. SQL is the language for talking to relational databases. It is used to create tables, insert data, and retrieve data. SQL queries are very similar across different database systems.

The created database has three tables:

- Table *prociv-occurrences*:
  In this table each row represents a unique record of a processed occurrence, containing the same attributes found in the Java class *ProvicOccurrence* in the listing 3.7. Represented in the figure 3.7 (a).

- Table *processed-news*:
  In this table each row represents a unique record of a processed news snippet, containing the same attributes found in the Java class *NewsProcessed* in the listing 3.6. Represented in 3.7 (b).

- Table *prociv-relevance*:
  In this table each row represents a unique record of each municipality information. It is used to map relevance and coordinates values to both aforementioned tables. Represented in 3.7 (c).

Attribute data types are chosen based on their use and interaction. The most common types are *varchar* which corresponds to small strings, *timestamp* which is used to work with dates and times, *float* corresponds to a number that includes decimal fractions and *text* is used to represent bigger strings.

Column Name	Data type	Not Null
id	varchar(255)	[v]
datetime	timestamp	[ ]
nature	varchar(255)	[ ]
district	varchar(255)	[ ]
municipality	varchar(255)	[ ]
lat	float8	[ ]
lon	float8	[ ]
ops	varchar(255)	[ ]
relevance	varchar(255)	[ ]
insertdate	timestamp	[ ]

(a)

Column Name	Data type	Not Null
title	text	[v]
description	text	[ ]
pubdate	timestamp	[v]
link	varchar(255)	[ ]
district	varchar(255)	[ ]
municipality	varchar(255)	[v]
relevance	varchar(255)	[ ]
lat	float8	[ ]
lon	float8	[ ]
topic	varchar(255)	[ ]
insertdate	timestamp	[ ]

(b)

Column Name	Data type	Not Null
municipality	varchar(255)	[v]
relevance	varchar(255)	[ ]
district	varchar(255)	[ ]
latitude	varchar(255)	[ ]
longitude	varchar(255)	[ ]

(c)

Figure 3.7: (a) Table prociv-occurrences (b) Table processed-news (c) Table prociv-relevance

Every table has at least one field with the constraint Primary Key (PK) that uniquely identifies each record in a table. PKs must contain unique values, and cannot contain *null* values. A table can have only one PK. The PK can consist of single or multiple columns (fields). Table prociv-occurrences 3.7 (a) has as PK the field *id* and its value is automatically retrieved by the Collector. Table prociv-relevance 3.7 (b) has as PK the field *Municipality* because each municipality name is unique. Finally, the PK of Table processed-news 3.7 (c) is a composite PK formed by the fields *title*, *pubdate* and *municipality*. In the beginning, the idea was using the *link* field as single PK but when more than one location is found in the news snippet, the news is replicated and inserted in the database so, the *link* field would not represent only one row of the database thus would be unique.

title	description	pubdate	link	distr	municipal	r	lat	lc	tc	insertdate
Hospital de Portalegre sem urg	Durante os próxim	-06-15 11:25:50.000	https://ww	Portalegre	Portalegre	3	166999817	66998863	others	-06-15 11:31:04.000
PSP registou 41 crimes por dia	A Polícia de Segur	-06-15 07:50:57.000	https://ww	Not Found	Not Found	Not Four	0	0	others	-06-15 11:31:04.000
Cavaco Silva diz que falta conci	Cavaco Silva foi ur	-06-14 22:09:04.000	https://ww	Braga	Fafe	2	500007629	67003632	others	-06-15 11:31:04.000
Crise nas urgências. SIM diz qu	Para os sindicatos	-06-14 20:43:45.000	https://ww	Not Found	Not Found	Not Four	0	0	others	-06-15 11:31:04.000
Maioria dos ginecologistas e ob	Mais de metade dc	-06-14 20:41:45.000	https://ww	Not Found	Not Found	Not Four	0	0	others	-06-15 11:31:04.000
Ministério Público investiga mc	O Ministério Públic	-06-14 20:39:45.000	https://ww	Leiria	Caldas da Rainha	2	068984985	36300087	others	-06-15 11:31:04.000
Ginecologia e obstetrícia. Urgê	Por falta de pediati	-06-14 20:38:45.000	https://ww	Faro	Portimão	2	333007812	33003998	others	-06-15 11:31:04.000
Ginecologia e obstetrícia. Urgê	Por falta de pediati	-06-14 20:38:45.000	https://ww	Faro	Faro	2	161018372	49999428	others	-06-15 11:31:04.000
Urgências obstetrícias. Hospital	Na passada madru	-06-14 20:36:45.000	https://ww	Not Found	Not Found	Not Four	0	0	others	-06-15 11:31:04.000
Urgências obstétricas fechadas	Mais urgências ob	-06-14 20:12:45.000	https://ww	Not Found	Not Found	Not Four	0	0	others	-06-15 11:31:04.000
Ministério Público do Seixal ani	O Ministério Públic	-06-14 20:03:53.000	https://ww	Setúbal	Seixal	1	427993774	61000824	others	-06-15 11:31:04.000
Resgatado corpo de jovem des	O jovem de 15 ano	-06-14 20:03:30.000	https://ww	Aveiro	Vagos	3	553016663	14002991	others	-06-15 11:31:04.000
Resgatado corpo de jovem des	O jovem de 15 ano	-06-14 20:03:30.000	https://ww	Porto	Porto	1	495018005	07997894	others	-06-15 11:31:04.000
Resgatado corpo de jovem des	O jovem de 15 ano	-06-14 20:03:30.000	https://ww	Aveiro	Aveiro	1	389007568	53001404	others	-06-15 11:31:04.000
Autarcas de Vila do Conde e Pó	Os autarcas de Vila	-06-14 19:28:22.000	https://ww	Porto	Porto	1	495018005	07997894	others	-06-15 11:31:04.000
Autarcas de Vila do Conde e Pó	Os autarcas de Vila	-06-14 19:28:22.000	https://ww	Porto	Vila do Conde	1	499984741	-8.75	others	-06-15 11:31:04.000
Homem morre eletrocutado nc	Um homem morre	-06-14 19:27:56.000	https://ww	Coimbra	Mira	5	285011292	63004684	others	-06-15 11:31:04.000
Homem morre eletrocutado nc	Um homem morre	-06-14 19:27:56.000	https://ww	Coimbra	Coimbra	1	211101532	91000366	others	-06-15 11:31:04.000
Saúde. A pressão está a aument	O diretor do serviç	-06-14 18:31:10.000	https://ww	Lisboa	Lisboa	1	452011108	04003906	others	-06-15 11:31:04.000

Figure 3.8: Processed news table

3.2 Implementation

At first, the proposed architecture was built and deployed in a local machine using several Windows Terminals to run the different modules and to monitor the several phases of data processing. Afterward,

this architecture was deployed in the AlticeLabs virtual machines. To accomplish that goal the project was automated.

Implementing the project in virtual machines consists of deploying the different modules in the cloud creating Kubernetes containers managed with Docker and creating Quarkus images.

All the other used frameworks are explained in section 2.3 with their advantages to the project. Specific libraries are enumerated below:

- **NLTK** - NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum [36]

  - **Tokenization**: Breaking text into words, phrases, symbols or other meaningful elements is a fundamental step in text classification. NLTK provides various tokenization methods. The chosen one was wordTokenize() to help split the text into individual tokens.
  - **Vectorization**: Converting text data into numerical vectors is a crucial step in machine learning-based text classification. NLTK provides several vectorization methods, the one used was TfidfVectorizer() to convert text data into numerical vectors.
  - **Classification algorithms**: NLTK provides several classification algorithms. SVMClassifier was used to build and train text classification models.
  - **Evaluation**: NLTK provides a range of evaluation metrics such as precision, recall, F1-score, and accuracy that can be used to evaluate the performance of text classification models.

- **spaCy** - Designed specifically for production use and helps to build applications that process large volumes of text. It can be used to build information extraction or natural language understanding systems or to pre-process text for deep learning [51]. Spacy was used with NLTK because it contains a Portuguese (Portugal) model vocabulary trained with *Wikipedia* articles.

  - **Stopword Removal**: spaCy provides a built-in stopwords corpus and a method that can be used to remove stopwords from the text. Token objects (words) can be filtered using a list comprehension to remove stopwords.
  - **Stemming and Lemmatization**: Stemming and lemmatization are techniques used to reduce words to their base form, in order to reduce the number of unique features in a text dataset. spaCy provides stemming and lemmatization algorithms, such as lemmatize(), PorterStemmer(), that can be used to transform words to their root form.

  - **Part-of-Speech (POS) Tagging**: POS involves assigning a part-of-speech tag to each word in a text, such as ”noun”, ”verb”, ”adjective”, and so on. This is done using the pos attribute of each Token (word) object in a Doc (text) object. It was used to extract information about the location of each piece of news.

- **scikit-learn**: The KNN algorithm was implemented using the scikit-learn library, which provides a KNeighborsClassifier class for classification tasks.

4

Experimental Analysis

Contents

Chapter 4 starts with an analysis of the data set used to train and test the models. Each classifier performance is presented and then the performance across topics is compared. The chapter ends with a final comparison and choice between the results of the three models used.

4.1 Dataset information

All the news used to train and test the different classifiers was retrieved daily from several Portuguese online newspapers from December 29^{th}, 2021 until July 25^{th}, 2022, containing a total of 14185 up-to-date news. RSS and API technologies were used as explained in subsection 3.1.1.A.

For classifier training purposes, each record contains text and a topic. Commonly, each text consists of only a couple of sentences and has on average 37.2 words, including stopwords, and has on average 22.7 words, excluding stopwords. The reduced number of words makes the text based classification task a difficult problem. Figure 4.1 shows a histogram of the frequency of news per topic. Revealing that the data set is highly unbalanced. There are 14185 records with the following distribution *fires*: 818, *meteorologic*: 429, *public gatherings*: 369 and *other*: 12570. The minor category *public gatherings* is assigned to only 2.6% of the records and the largest category *others* is assigned to 88.6% of the records. It is important to note that it was decided to keep the real-world dataset unbalanced in order not to bias the models. This allows us to expect a similar model performance when under real use cases.

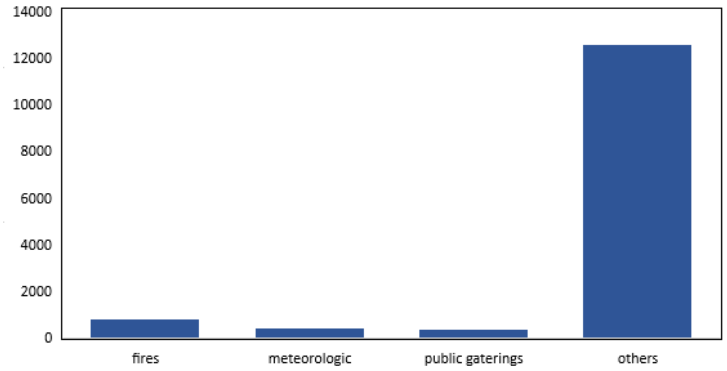


Figure 4.1: Number of news labeled for a given category.

Text classification works usually demand higher amounts of data to achieve better and more reliable results. Moreover, data was collected over half a year and it is biased by the conditions of that period. During the first times of collection, the Covid pandemic was in every piece of news, later the Ukrainian war and the latest the huge wildfires in Portugal. Certainly, no two years are exactly the same, but it is expected that the data collected does not represent the generality of the Portuguese national news. Nevertheless, the chosen classification model is expected to improve its performance over time as the data set increases.

4.2 Classifiers performance

This section aims to evaluate the performance of text classifiers on a dataset of 14185 news snippets. The dataset was divided into a training set and a test set, with 80% and 20% of the records respectively. The training set was used to tune the hyperparameters and test various text-processing techniques, while the test set was reserved for a blind evaluation of the classifiers.

The evaluation process involved the following steps: First, the hyperparameters were optimized on the validation set. Then, the best combination of hyperparameters and text-processing techniques was applied to the test set, and the results were recorded in a confusion matrix. Finally, the confusion matrix was analyzed to assess the overall performance of the classifiers and to identify any potential areas for improvement.

The results of this study provide valuable insights into the performance of text classifiers and the impact of different text processing techniques on the score of the classifier. The findings can be used to inform future research on text classification and to guide the development of more effective text classification systems.

For each scenario, we tested the influence of the removal of *stopwords* and *stemming* on the results. The first was implemented using *nltk.corpus* Portuguese library and consists in removing from text Portuguese words that do not provide any useful information to decide in which topic a text should be classified. The latter is the process of reducing inflected (or sometimes derived) words to their word stem, base, or root. This technique was implemented using *SnowballStemmer* from *NLTK* framework Portuguese version and simplifies category's *bag-of-words*.

In our models, the preprocessing stage, implemented in Python, consists of converting to lowercase and removing punctuation and characters, and then stripping, removing stopwords, and stemming. In all classifiers, we use TF-IDF.

SVM and KNN were modeled using a Python library *scikit-learn*. The FFP method was modeled using the same approach as [17]. The exact same training data sets and test data sets were used for all methods.

Several test scenarios were built to find each algorithm's optimal performance setting. The SVM model was applied as a binary classifier and as a multi-class classifier. KNN e FFP were applied as multi-class classifiers.

In this project, a bigger dataset was needed to achieve good results. In order to overpass this problem, Cross Validation was used. Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample. The method involves dividing the original sample into complementary subsets, with each subset used as a test set for evaluating the model fit on the remaining training data. In this specific case, 5-fold cross-validation was employed. The data was split into a training set (80%) and a test set (20%) in a way that meets the 5 cross-validation proportions. The purpose

of the procedure was to tune hyperparameters and prevent overfitting.

4.2.1 SVM performance

In this subsection, the performance of SVM classifiers is evaluated on a dataset of 14185 news snippets. Both multiclass and binary class SVM models were trained using a bag-of-words representation with TF-IDF weighting, and the soft-margin parameter C that was optimized during the training phase. The SVM models were tested with both linear and non-linear kernels, and it was found that the linear kernel provided better results. Word embeddings were tested but produced worse results, probably due to the lack of data to create a good model.

4.2.1.A SVM multi-class

For the multi-class model, we tried to predict all classes from the same SVM model. During the training phase, hyper-parameters C and kernel were tuned to get the best results. The algorithm was trained with a C value in the range of 0.001 to 1000, the kernels tested were linear, poly, rbf, and sigmoid, and different text processing techniques were tested. Table 4.1 resumes the algorithm's performance with the influence of stemming and removing stopwords when tuning hyperparameters on the training set. The hyperparameters that produced the best results are linear *kernel*, *C* equals to 1, and stemming and removing stopwords as text processing techniques.

Table 4.1: Hyper parameters tuning and various text processing techniques influence the training set on multi-class SVM.

Kernel	C	Text Processing	Precision	Recall	F1-Score
linear	2	None	0.818	0.668	0.735
linear	2	Remove stopwords	0.834	0.642	0.726
linear	1	Stemming	0.818	0.666	0.734
linear	1	Stemming, Remove stopwords	0.795	0.685	**0.736**

In Table 4.2, the results of SVM multi-class classification are presented, using the hyperparameters and text processing techniques that had the best F1 scores during the training phase. The performance of the classifier was evaluated on a separate test set and the results are reported in terms of various metrics, including precision, recall, and F1-score. These metrics provide insight into the effectiveness of the classifier in correctly identifying each class and its overall performance. The results in this table can be used to compare the performance of SVM with other multi-class classification methods and to determine the suitability of SVM for a specific application.

The precision of the fires class is 0.87, which indicates that the classifier is making very few false positive predictions for this class. The recall of 0.86 indicates that the classifier is detecting 86% of all

fires instances in the data. The F1-score of 0.865 is a balanced evaluation of the precision and recall and indicates that the classifier is performing well overall on the fires class.

The precision of the meteorologic class is 0.80, which indicates that the classifier is making a low number of false positive predictions for this class. The recall of 0.73 indicates that the classifier is detecting 73% of all meteorologic instances in the data. The F1-score of 0.763 is a balanced evaluation of the precision and recall and indicates that the classifier is performing well overall in the meteorologic class.

The precision of the public gatherings class is 0.80, which indicates that the classifier is making a low number of false positive predictions for this class. However, the recall of 0.49 indicates that the classifier is detecting only 49% of all public gatherings instances in the data. The F1-score of 0.608 is a balanced evaluation of the precision and recall and indicates that the classifier is performing poorly in the public gatherings class.

In conclusion, the classifier is performing well on the fires and meteorologic classes, with a high precision, recall, and F1-score. However, the classifier is performing poorly on the public gatherings class, with a low recall, which means that it is not detecting a high percentage of all public gatherings instances in the data.

Table 4.2: SVM multi-class scores on the test set for linear Kernel, C = 1, text with no stopwords and stemming for each topic

Category	Precision	Recall	F1-Score	Support
Fires	0.87	0.86	0.865	164
Meteorologic	0.80	0.73	0.763	86
Public gatherings	0.80	0.49	0.608	74
Macro avg	0.823	0.693	0.745	324

Based on the confusion matrix in figure 4.2, we can see that the classifier performed well on the class *fires* with 141 instances being correctly classified out of 163. The class *meteorologic* had 63 out of 86 instances correctly classified, which is a decent performance. However, there were still 23 instances of this class that were classified as a different class, which could potentially impact the results of an application. The model performed poorly on the class *public gatherings* having 36 out of 74 instances correctly classified. The biggest issue with this class classification is the large quantity of false negatives instances. This could indicate that the model is having difficulty capturing the characteristics of this class.

True topic \ Predicted topic	Fires	Meteorologic	Public Gatherings	Others
Fires	141	1	0	22
Meteorologic	3	63	0	20
Public Gatherings	0	0	36	38
Others	18	15	9	2472

Figure 4.2: Confusion matrix results for multi-class SVM with Stemming, Remove stopwords and C=1

4.2.1.B SVM binary

The SVM binary model was constructed from four binary models corresponding to one of our classes: *fires*, *meteorologic*, *public gatherings*, and *others*. In order to train each model we selected news labeled with the corresponding category as positive samples and all the other news as negative samples. At the end of the classification, each piece of news is assigned to the category with the highest score.

For the binary-class model, we used the same tuning as multi-class SVM. During the training phase, hyper-parameter C and the kernel were tuned to get the best results. The algorithm was trained with a C value in the range of 0.001 to 1000, the kernels tested were linear, poly, rbf, and sigmoid, and different text processing techniques were tested. Table 4.3 resumes the binary SVM performance. The results obtained for SVM binary classifier were similar to the SVM multi-class. The hyperparameters that produced the best results are linear *kernel*, *C* equals to 1, and removing stopwords as text processing techniques.

Table 4.3: Hyper parameters tuning and various text processing techniques influence the training set on binary-class SVM.

Kernel	C	Text Processing	Precision	Recall	F1-Score
linear	0.001	None	0.833	0.629	0.717
linear	1	Remove stopwords	0.799	0.681	**0.735**
linear	1	Stemming	0.774	0.657	0.711
linear	1	Stemming, Remove stopwords	0.778	0.693	0.733

In Table 4.4, the results of SVM binary-class classification are presented, using the same approach as SVM multi-class.

The results show that the model has good performance for the *fires* class, good but not excellent performance for the *meteorologic* class, and poor performance for the *public gatherings* class. In general, this model brings no advantages when compared to multi-class SVM regarding classification scores.

Table 4.4: SVM binary-class scores on the test set for linear Kernel, C = 1, text with no stopwords for each topic

Category	Precision	Recall	F1-Score	Support
Fires	0.870	0.86	0.865	164
Meteorologic	0.797	0.690	0.745	86
Public gatherings	0.760	0.530	0.620	74
Macro avg	0.810	0.693	0.743	324

Based on the confusion matrix in figure 4.3, we can see that the classifier performed well on the classes *fires* and *meteorologic* with 141 and 59 instances being correctly classified out of 163 and 86, respectively. The model performed poorly on the class *public gatherings* having 39 out of 74 instances correctly classified. Overall, the precision values for each class were good, but the recall for *meteorologic* and *public gatherings* classes is low. Mainly because a high number of instances from these two classes are being classified as instances of the class *others*. This could indicate that the model is having difficulty capturing the characteristics of these classes.

		Predicted topic			
		Fires	Meteorologic	Public Gatherings	Others
True topic	Fires	141	2	0	21
	Meteorologic	3	59	0	24
	Public Gatherings	0	0	39	35
	Others	18	13	12	2471

Figure 4.3: Confusion matrix results for multi-class SVM with Remove stopwords and C=1

4.2.2 FFP performance

In our study, we employed two distinct methodologies for classifying the FFP. The first methodology involved training the model using four news topics, whereby each news instance was assigned a similarity score by each fingerprint corresponding to each topic. If the similarity score was below the predefined threshold, the instance was classified as *others*. In contrast, the second methodology entailed testing the model with only three topics, excluding the *others* category. Any news instance with a similarity score below the threshold was attributed to the *others* topic. Subsequent to conducting tests, it was revealed that the first approach yielded a 3% superior F1-score, thus rendering it the approach that is expounded upon in the ensuing text.

For the FFP model, we tried to predict all classes from the same model. During the training phase, hyper-parameters *K* and *threshold* were tuned to get the best results. The algorithm was trained with a

K value in the range of 150 to 800, the *threshold* in the range of 0.08 to 0.24 and different text processing techniques were tested. Table 4.5 resumes the algorithm's performance with the influence of stemming and removing stopwords when tuning hyperparameters on the training set. The hyperparameters that produced the best results are *K* equal to 500, *threshold* equal to 0.13, and removing stopwords as text processing techniques. We

Table 4.5: Hyper parameters tuning and various text processing techniques influence the training set on FFP.

K	Threshold	Text Processing	Precision	Recall	F1-Score
600	0.09	None	0.510	0.831	0.632
500	0.13	Remove stopwords	0.606	0.743	**0.668**
500	0.12	Stemming	0.51	0.789	0.620
300	0.14	Stemming, Remove stopwords	0.567	0.732	0.639

In Table 4.6, the results of FFP model classification are presented, using the hyperparameters and text processing techniques that had the best F1 scores during the previous phase. The performance of the classifier was evaluated on a separate test set and the results are reported in terms of various metrics, including precision, recall, and F1-score.

The performance is acceptable, but the results are below the SVM models and are heavily penalized by the performance in the minor public gatherings class. The results are not totally unexpected, since FFP usually demand longer texts to perform well and FFP tend to outperform better other classifiers when the number of categories is much larger [17]. Overall results with a threshold equal to 0.13, fingerprint size equal to 500, with stopwords removed and no stemming, are shown in Table 4.6.

Table 4.6: FFP scores on the test set for threshold=0.13, K=500, text with no stopwords for each topic

Category	Precision	Recall	F1-Score	Support
Fires	0.740	0.830	0.782	164
Meteorologic	0.650	0.850	0.737	86
Public gatherings	0.460	0.680	0.550	74
Macro avg	0.617	0.787	0.690	324

Based on the confusion matrix in figure 4.4, we can see that the classifier performed relatively well on the classes *fires* with 136 instances being correctly classified out of 163 (even though this metric is worse than the previous results). The model performed poorly on the class *meteorologic* and *public gatherings*. The most significant value is 0.46 for precision regarding *public gatherings* class. This means, that the model is not very precise in identifying positive instances, and it has a high rate of false positives. This could be due to a variety of factors such as the complexity of the data, the quality of the features used in the model, or the specific parameters chosen for the FFP algorithm.

True topic \ Predicted topic	Fires	Meteorologic	Public Gatherings	Others
Fires	136	2	0	26
Meteorologic	2	73	0	11
Public Gatherings	1	0	50	23
Others	45	37	59	2373

Figure 4.4: Confusion matrix results for FFP threshold=0.13, K=500, text with no stopwords

4.2.3 KNN performance

During the training phase, hyper-parameter K was tuned to get the best results. The algorithm was trained with a K value in the range of 1 to 201 and different text processing techniques were tested. Table 4.7 resumes the algorithm's performance with the influence of stemming and removing stopwords when tuning hyperparameters on the training set. The hyper-parameter that produced the best results is K equals 11 and removing stopwords as text processing techniques.

Table 4.7: Hyper parameters tuning and various text processing techniques influence the training set on KNN.

K	Text Processing	Precision	Recall	F1-Score
17	None	0.783	0.622	0.693
11	Remove stopwords	0.774	0.657	**0.711**
51	Stemming	0.867	0.5	0.634
41	Stemming, Remove stopwords	0.865	0.551	0.673

Overall results with K equal to 11, with stopwords removed, are shown in Table 4.8.

For the three classes fires, meteorologic, and public gatherings, the following results are obtained.

Classes fires and meteorologic have a value F1-score of 0.83 and 0.759, respectively. This suggests that the classifier is doing well for both classes. Regarding the public gatherings class, its metrics values are precision of 0.760, recall of 0.350, and F1-score of 0.480. This indicates that the classifier is not doing well for this class, with relatively low precision and recall, and a low F1 score. This suggests that the classifier is making a high number of false positive and false negative predictions for this class.

Table 4.8: KNN scores on the test set for K=11, text with no stopwords for each topic

Category	Precision	Recall	F1-Score	Support
Fires	0.830	0.830	0.830	164
Meteorologic	0.790	0.730	0.759	86
Public gatherings	0.760	0.350	0.480	74
Macro avg	0.793	0.637	0.689	324

Based on the confusion matrix in figure 4.5, we can see that the classifier performed relatively well on the classes *fires* and *meteorologic* with 136 and 63 instances being correctly classified out of 163 and 86, respectively. The model performed poorly on the class *public gatherings*. The most significant value is the low value of 0.35 for recall in *public gatherings* class. This happens because a high number of instances from this class is being classified as instances of the class *others*. This could indicate that the model is having difficulty capturing the characteristics of this class.

		Predicted topic			
		Fires	Meteorologic	Public Gatherings	Others
True topic	Fires	136	2	0	26
	Meteorologic	6	63	0	17
	Public Gatherings	0	0	26	48
	Others	22	15	8	2469

Figure 4.5: Confusion matrix results for KNN, K=11, text with no stopwords

4.3 Classifiers comparison

We have performed experiments using four classification approaches, multi-class and binary SVM, KNN, and FFP. Our real-world dataset is highly unbalanced where the frequency of the largest class *others* is almost 8 times greater than the other 3 categories combined. Our results reveal that despite the short text size (title and description of a piece of news) it is possible to predict its category with an overall performance of about 74.5% F1-Score when using a binary SVM approach on Figure 4.9. When comparing all methods, it is evident that the SVM models outperform the others. Regarding the referenced evaluation metrics, the SVM binary model is slightly better than the SVM multi-class. The last model achieves the best values for precision and f-score value. The FFP strategy, while achieving the best recall from all the classifiers achieves 21% lower precision in comparison to the SVM multi-class. One reason could be the small number of words per piece of news, which jeopardizes heavily fingerprints' performance. Classifiers KNN and FFP have the same F1 score, although both precision and recall values are contrasting. FPP performance is superior to others when dealing with a large number of classes which is not the situation. One reason that can explain why SVM performs better than KNN is that it can be better fine-tuned than the latter. SVM models are often good at handling imbalanced datasets, where the number of samples in each class is not equal. This is because the SVM algorithm is not affected by the imbalance of the data and instead focuses on finding the best separation boundary between the classes. Overall, the SVM model's ability to handle imbalanced data and the flexibility in hyperparameter tuning may have contributed to its high F1 score in this classification.

Table 4.9: Comparison of all models using Precision, Recall, and F1-Score

Model	Precision	Recall	F1-Score
SVM multi-class	**0.823**	0.693	**0.745**
SVM binary	0.810	0.693	0.743
FFP	0.617	**0.787**	0.690
KNN	0.793	0.637	0.689

When analyzing Table 4.10, it is clear that *public gatherings* class achieves the worst result in every model, having a best F1-score value of 0.62 and worst 0.48, which represents an amplitude of 0.14. *Fires* class is the best performer in every model having a best F1-score value of 0.865 and worst of 0.782, which represents an amplitude of 0.083. Finally, *meteorologic* performs well in every model having a best F1-score value of 0.763 and worst of 0.737, which represents an amplitude of 0.026.

The lowest F1-score amplitude value of *meteorologic* class can be related to the homogeneous instances of text from this class. That is why its evaluation metrics are similar along different classifiers. On the other hand, the highest F1-score amplitude value belongs to *public gatherings* class. This category is not only the one with fewer training examples, but also the most heterogeneous one, which justifies why it has the worst score in every model in Table 4.10. There could be several reasons for this poor performance. Lack of training data for this class, leading to poor representation and low model performance. Overlap or confusion with other classes makes it difficult for the classifier to accurately distinguish between different classes. The features used in the model may not be sufficient to distinguish between the different classes, leading to poor performance. Possible solutions to improve its score could be to split the category into more specific ones or increase data training records.

Table 4.10: Performance across topics

Topic	SVM multi-class	SVM binary	FFP	KNN
Fires	**0.865**	**0.865**	0.782	0.830
Meteorologic	**0.763**	0.745	0.737	0.759
Public gatherings	0.608	**0.620**	0.550	0.480

5

Conclusion

Contents

Chapter 5 outlines the main findings of the conducted experiments and suggests possible extensions and future research directions.

5.1 Summary of Findings

This study presents a comprehensive evaluation of lightweight ML models for text classification on Portuguese news snippets. In addition, an ETL pipeline was developed to store and analyze the news data and civil protection events that could be correlated with network jeopardizing. The study examines the impact of training data on the performance of these models.

Our project adopts a heuristic approach informed by an understanding of the needs of assurance applications. By providing a detailed overview of requirements and data characteristics, along with statistical analysis, each model is evaluated regarding its strengths and weaknesses across text classification. F1-score is used as an evaluation metric to gain a better assessment of topic properties and distribution. The study reveals that the quality and quantity of training data can significantly impact classification performance.

The public gatherings topic, in particular, exhibits lower performance across all models. This can be attributed to several factors:

- **Fewer samples**: The public gatherings topic has the fewest samples, which might not be enough to train a robust model.
- **Heterogeneity**: The public gatherings class is also a heterogeneous class, which means that it encompasses a diverse range of events and scenarios. This can make it difficult for the model to accurately identify and classify these events. A more homogeneous class with similar examples would be easier for the model to learn.
- **Complexity**: Public gatherings can also be complex events that involve a lot of different factors, such as the number of people, location, and purpose. Capturing these nuances can be challenging, especially with limited data.

Overall, the poor performance of the public gatherings class is likely due to a combination of these factors. Including confusion matrices in the analysis enables realize how frequently this topic is misclassified as the topic ”Others” due to its complexity and different purposes. To improve this topic’s performance, one could consider collecting more data, creating a more homogeneous class, or using a more complex model to better capture the nuances of public gatherings.

In addition, this thesis demonstrates that despite the challenges posed by short text length, it is possible to achieve an overall performance of approximately 74.5% F1-score for predicting news categories

using a binary SVM approach. Overall, this research provides valuable insights into the challenges and opportunities of text classification for news data in Portuguese.

5.2 Limitations and Future Work

This study employed a specific methodology to collect data, which allowed for the reflection of potential issues faced by an operator in a particular region. However, it is important to acknowledge the limitations and areas for improvement in the current approach.

One limitation is the possibility that the collected data may not fully capture all problems encountered by the operator. The methodology used in this study focused on proactive and reactive characteristics of the platform, aiming to anticipate events that could affect network quality and prioritize network repairs in identified areas of degradation. Nevertheless, there may be other factors and events that were not considered in the data collection process, which could impact the overall performance of the network.

To address this limitation and improve the classifier's performance, future work could explore the collection of additional data. Increasing the amount of data available for analysis could enhance the model's ability to identify and classify network-related issues. Furthermore, efforts should be made to ensure that the collected data is more homogeneous in terms of classes. By creating more balanced and representative classes, the classifier can better generalize and accurately classify network problems.

Additionally, a key aspect for future research is the analysis of the impact of the identified events on the reduction of service complaints. Although the objective of signaling useful events for network troubleshooting was achieved in this study, it was not possible to examine in a timely manner how these events influenced the decrease in service complaints. Future investigations should focus on analyzing this relationship and determining the extent to which the proactive and reactive characteristics of the platform contribute to customer satisfaction and the overall quality of service.

In conclusion, while this study successfully applied the methodology to identify and prioritize network issues, there are limitations that need to be addressed in future work. By collecting more data and creating more homogeneous classes, the classifier's performance can be improved, leading to more accurate identification and classification of network problems. Furthermore, exploring the impact of identified events on service complaints will provide valuable insights for enhancing customer satisfaction and meeting market demands.

Bibliography

[1] T. Gonçalves, "Públicos e consumos de média: o consumo de notícias e as plataformas digitais em portugal e em mais de dez países," in *Entidade Refuladora para a comunicação Social*, 2015, p. 36.

[2] S. Dixon, "Leading mobile social media websites in portugal in march 2022, based on share of visits," https://www.statista.com/statistics/1272959/portugal-share-social-mobile/#statisticContainer, journal=Statista 2022, 04 2022, accessed: 2022-06-21.

[3] C. Manliguez, "Generalized confusion matrix for multiple classes," 11 2016.

[4] ANACOM, "Estatísticas," 06 2022. [Online]. Available: https://www.anacom-consumidor.pt/reclamacoes-no-sector-das-comunicacoes

[5] R. H. e. a. Gunarathne, P., "Whose and what social media complaints have happier resolutions? evidence from twitter," *Journal of Management Information Systems*, vol. 34, pp. 314–340, 04 2017.

[6] V.-C. R. Del Río-Lanza, A. B. and A. M. Díaz-Martín, "Satisfaction with service recovery: Perceived justice and emotional responses," *Journal of Business Research*, pp. 775–781, 2009.

[7] B. T. W. J. Smith, A., "A model of customer satisfaction with service encounters involving failure and recovery," *Journal of Marketing Research*, vol. XXXVI, pp. 356–372, 09 1999.

[8] M. Davidow, "The bottom line impact of organizational responses to customer complaints," *Journal of Hospitality and Tourism Research - J Hospit Tourism Res*, vol. 24, pp. 473–490, 11 2000.

[9] R. J. Tewksbury, D., *News on the Internet: Information and Citizenship in the 21st Century*, 03 2012.

[10] "Livro de reclamações online," https://www.livroreclamacoes.pt/, accessed: 2022-06-21.

[11] "Portal da queixa - sobre nós," https://portaldaqueixa.com/about-us, accessed: 2022-06-21.

[12] "Meo fórum, pergunta, responde e contribui," https://forum.meo.pt/, accessed: 2022-06-21.

[13] "Real-time problem and outage monitoring," https://downdetector.com/, accessed: 2022-06-21.

[14] C. J. P. Simões, A., "Fast text based classification of news snippets for telecom assurance," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Cham: Springer International Publishing, 2022, pp. 69–81.

[15] K. S. T. V. Ikonomakis, E., "Text classification using machine learning techniques," *WSEAS transactions on computers*, vol. 4, pp. 966–974, 08 2005.

[16] K. D. A. M. DAha, D., "Instance-based learning algorithms," *Machine Learning*, vol. 6, no. 1, pp. 37–66, 1991. [Online]. Available: https://eprints.soton.ac.uk/18494/

[17] B. F. C. J. Rosa, H., "Twitter topic fuzzy fingerprints," 07 2014, pp. 776–783.

[18] e. a. Lee, K., "Twitter trending topic classification," 12 2011, pp. 251–258.

[19] E. A. . S.-F. Baccianella, S., "Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining," in *LREC*, 2010, pp. 2200–2204.

[20] C. G. W. X.-. Y. D. Li, X., "Acoustic modeling using deep neural networks for lvcsr," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4445–4449.

[21] . L. Q. Vinyals, O., "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[22] R. R. Batista, F., "Sentiment analysis and topic classification based on binary maximum entropy classifiers," *Procesamiento de Lenguaje Natural*, vol. 50, pp. 77–84, 03 2013.

[23] M. C. Wang, S., "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 90–94. [Online]. Available: https://aclanthology.org/P12-2018

[24] J. Homem N., Carvalho, "Authorship identification and author fuzzy "fingerprints"," in *Fuzzy Information Processing Society (NAFIPS), 2011 Annual Meeting of the North American*, 03 2011.

[25] C. M. L. K. Devlin, J. and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 10 2018.

[26] Z. Z. Chen, Q. and W. Wang, "Bert for joint intent classification and slot filling," 02 2019.

[27] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," 01 2018.

[28] J. R. Rajpurkar, P. and P. Liang, "Know what you don't know: Unanswerable questions for squad," 2018. [Online]. Available: https://arxiv.org/abs/1806.03822

[29] G. Becquin, “End-to-end NLP pipelines in rust,” in *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 20–25. [Online]. Available: https://aclanthology.org/2020.nlposs-1.4

[30] O. M. G. N.-D. J. e. a. Liu, Y., “Roberta: A robustly optimized bert pretraining approach,” 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[31] F. X. Q. B. Gong, H. and T. Liu, “Table-to-text generation with effective hierarchical encoder on three dimensions (row, column and time),” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3143–3152. [Online]. Available: https://aclanthology.org/D19-1310

[32] H. J. X. C.-e. a. Wu, Y., “Research on named entity recognition of electronic medical records based on roberta and radical-level feature,” *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, 06 2021.

[33] L. X. Yang, Y., “A re-examination of text categorization methods,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR ’99. New York, NY, USA: Association for Computing Machinery, 1999, p. 42–49. [Online]. Available: https://doi.org/10.1145/312624.312647

[34] C. J. e. a. Marujo, L., *Textual Event Detection Using Fuzzy Fingerprints. In: Angelov, P., et al.*, 2015.

[35] S. J. Feldman, R., *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*, 12 2006, vol. 34.

[36] NLTK, “Natural language toolkit,” 21 2022. [Online]. Available: https://www.nltk.org/

[37] B. C. Salton, G., “Term-weighting approaches in automatic text retrieval,” *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0306457388900210

[38] e. a. Sood, S., “Tagassist: Automatic tag suggestion for blog posts,” in *ICWSM*, 2007.

[39] U. J. Rajaraman, A., *Mining of Massive Datasets*. Cambridge University Press, 2011.

[40] S. P. . J. W. Rilo[U+FB00], E., “Feature subsumption for opinion analysis,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2006.

[41] C. K. C. G. Mikolov, T. and J. Dean, “Efficient estimation of word representations in vector space,” 2013. [Online]. Available: https://arxiv.org/abs/1301.3781

[42] S. N. P. N. U. J. J. e. a. Vaswani, As., “Attention is all you need,” 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[43] O. A. L. Cardoso-Cachopo, A., “An empirical comparison of text categorization methods,” in *String Processing and Information Retrieval*, M. A. Nascimento, E. S. de Moura, and A. L. Oliveira, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 183–196.

[44] H. S. Lim, “Improving knn based text classification with well estimated parameters,” in *Neural Information Processing*, N. R. e. a. Pal, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 516–523.

[45] “Kubernetes vs docker: What’s the difference?” https://www.dynatrace.com/news/blog/kubernetes-vs-docker/, accessed: 2022-07-18.

[46] “Apache maven project,” https://maven.apache.org/what-is-maven.html, accessed: 2022-07-18.

[47] Quarkus, “Quarkus,” 21 2022. [Online]. Available: https://quarkus.io/about/

[48] Apache, “Kafka,” 08 2022. [Online]. Available: https://www.confluent.io/what-is-apache-kafka/?utm_medium=sem&utm_source=google&utm_campaign=ch.sem_br.nonbrand_tp.prs_tgt.kafka_mt.xct_rgn.emea_lng.eng_dv.all_con.kafka-general&utm_term=kafka&creative=&device=c&placement=

[49] PostgreSQL, “Postgresql,” 08 2022. [Online]. Available: https://aws.amazon.com/pt/rds/postgresql/what-is-postgresql/

[50] “Named entity recognition for portuguese,” https://melaniewalsh.github.io/Intro-Cultural-Analytics/05-Text-Analysis/Multilingual/Portuguese/02-Named-Entity-Recognition-Portuguese.html, accessed: 2022-07-01.

[51] “Industrial-strength natural language processing,” https://spacy.io/, accessed: 2022-07-25.