FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Noise Reduction in Automatic Complaints Processing

Inês Alves Quarteu

Master in Informatics and Computing Engeneering

Supervisor: Henrique Lopes Cardoso

Second Supervisor: André Restivo

July 28, 2023

Noise Reduction in Automatic Complaints Processing

Inês Alves Quarteu

Master in Informatics and Computing Engeneering

Approved in oral examination by commitee:

President: Daniel Silva
Referee: Joaquim Gonçalves
Supervisor: Henrique Lopes Cardoso

July 28, 2023

Resumo

Reclamações submetidas por indivíduos são expressões de insatisfação que realçam problemas e procuram soluções em vários contextos. A necessidade de processar as queixas decorre dos conhecimentos valiosos que estas proporcionam. O tratamento eficaz das reclamações é essencial para resolver as queixas individuais, identificar problemas sistemáticos numa instituição e melhorar os serviços prestados por esta.

Embora todas as queixas devam ser processadas e tratadas de forma adequada e atempada, as queixas são frequentemente caracterizadas por conterem ruído, que pode incluir elementos como *templates* ou estruturas de *boilerplate*. Consequentemente, torna-se um desafio desenvolver um sistema que possa tratar automaticamente as novas reclamações recebidas, de modo a que possa ser implementada uma resposta adequada sem ter em conta a presença do ruído descrito. O sistema deve distinguir eficazmente entre informação relevante e elementos ruidosos para gerar respostas adequadas.

Esta tese explora o desenvolvimento de métodos de redução de ruído a aplicar especificamente a dados de reclamações extraídos de ficheiros de texto, alguns dos quais recorrem a técnicas de Reconhecimento Ótico de Caractéres. Para o efeito, utilizam-se como caso de estudo dados reais fornecidos por entidades reguladoras - mais precisamente, reclamações relacionadas com a saúde - e propõe-se uma pipeline como solução. São utilizadas diferentes abordagens em várias fases da solução proposta, e os resultados que estas produzem nos dados serão contrastados e comparados.

Finalmente, verifica-se que a solução desenvolvida resulta num melhor desempenho através da utilização do NLTK Punkt como método de segmentação de texto e de uma variante multilingue do SBERT como meio de representação de texto. A abordagem mencionada é capaz de identificar a maioria das instâncias de boilerplate num conjunto de dados criado para avaliação.

Abstract

Complaints are expressions of dissatisfaction that highlight individuals' issues and seek resolutions in various contexts. The need to process complaints arises from the valuable insights these provide. Effective complaint handling is essential to address personal grievances, identify systemic problems within an institution, and improve the services provided.

While all complaints should be processed and dealt with appropriately and timely, claims are often characterized as containing noise, which can include elements like templates or boilerplate structures. Consequently, it becomes challenging to develop a system that can automatically handle new incoming complaints so that a suitable response may be implemented without addressing the presence of such noise. The system must effectively distinguish between relevant information and extraneous elements to generate appropriate responses.

This thesis explores the development of noise reduction methods to be applied specifically to complaint data extracted from text files, some of which resort to Optical Character Recognition techniques. In order to do so, real data provided by regulatory bodies — health-related complaints, more precisely — is utilized as a study case, and a pipeline is proposed as a solution. Different approaches are employed in varied stages of the proposed solution, and the results these yield in the data are contrasted and compared.

In the end, the developed solution is found to result in better performance through the employment of NLTK Punkt as a text segmentation method and a multilingual variant of SBERT as a means of text representation. The mentioned approach is able to identify most boilerplate instances in a dataset created for evaluation.

Keywords: Complaint Handling, Noise Reduction Models, Boilerplate Detection
ACM Classification: CCS → Computing methodologies → Artificial intelligence → Natural language processing

Acknowledgements

I would like to thank my supervisors Henrique Cardoso and André Restivo for all the support and guidance throughout this process.

I would also like to thank my friends and family and give a special thank you to my parents, Adília Alves and Reis Quarteu, and my brother Hugo for supporting me through these years, and for having the patience to listen to my despairs in my weakest moments and advice me through them.

Inês Quarteu

"You should be glad that bridge fell down.
I was planning to build thirteen more to that same design"

Isambard Kingdom Brunel

Contents

List of Figures

List of Tables

Abbreviations and Symbols

ERS	Portuguese Health Regulatory Body
NLP	Natural Language Processing
SBD	Sentence Boundary Detection
BERT	Bidirectional Encoder Representations from Transformers
AI	Artificial Intelligence
OCR	Optical Character Recognition

Chapter 1

Introduction

Portuguese citizens, and other entities, partake in daily interactions with the services provided by the economic operators working in the national territory or any public services. These exchanges often result in the submission of formal complaints, of various natures, with the respective regulatory body. The regulatory body is required to adequately handle these contacts, a process that can quickly grow in scale and become unmanageable due to the sheer number of contacts. The application of Natural Language Processing (NLP) techniques can help address information overload and achieve higher levels of user satisfaction in the provided services by automating the processing of textual data and thus enabling the understanding of the nature of each report and the consequent appropriate course of action.

The Portuguese Health Regulatory Body (ERS) is responsible for inspecting all healthcare providers operating in the national territory and regulating the activities these institutions offer to ensure the rights of all users and enforce regulatory legislation [9]. As an entity that operates on a national wide scale, ERS is required to manage a vast quantity of diversified data. One of the types of information that the organization must handle is the reports submitted by users as a consequence of their interactions with health-related services. These reports can be submitted through various procedures, from online forms to emails, letters, and physical complaint forms.

Some available complaint submission procedures implement predefined structures with fields where the complaint is included. The formats allow the author to input their complaint within specific fields provided, ensuring a standardized presentation of the complaint with relevant data in the predefined structure. The wide variety of templates within which information can be submitted, allied with the free nature of the user-generated content that characterizes complaints, constitutes a challenge to automating the report-handling process. Apart from a classification task, this process also entails the detection of duplicate complaints, as the submission of complaints regarding the same episode can co-occur through the different existing means of communication.

1.1 Motivation

The presence of noise in textual input data has been repeatedly shown to critically degrade the performance of various language models [2], such as BERT [18], as the accumulation of noise in the input text denotes decreased accuracy in the results. Particularly in classification tasks, when the input data is sufficiently noisy, using language models to categorize texts can be problematic. Therefore, when developing a model intended for a classification task, the input data must have minimal noise.

The data concerning the user-submitted complaints contains text from varied supports, which constitutes a challenge to producing a model to process each report automatically and adequately. As previously explained, user-generated data associated with every complaint is embedded within a structured template that changes with each submission origin and contains meta-information about the filing of the report itself. This information does not offer knowledge about the content of the written complaint and therefore constitutes noise in the textual input data. As a result, identifying the text relating to the claim in question and removing boilerplate data are necessary steps to clear noise from the textual data upon which the model will be assembled.

Applying adequate noise reduction methods that specifically target the removal of boilerplate text is expected to minimize the influence of the related issues on input textual data. Consequently, the subsequent modeling process will be optimized, namely, the classification and duplicate detection tasks mentioned above. Ultimately, the goal is to optimize the handling of complaints by the regulatory body, allowing for the automatic processing of all data content and promoting the most urgent complaints, thus facilitating a better operation of the institution and, consequently, a better health service for citizens.

1.2 Objectives

The primary pursuit of this dissertation is to examine the development of a method that diminishes noise in textual data, explicitly oriented for data resulting from applying text extraction algorithms to files and documents characterized by the presence of template structures. The development of this method must be based on the analysis of the textual data being worked on, the sources of noise within it, and, eventually, the definition and understanding of the properties of each noise source. The investigation documented in this thesis is guided by the definition of research questions presented below:

Research Question 1: Is it possible to develop a pipeline to automatically detect boilerplate content in a diverse document collection containing multiple templates? As the data provided by the ERS is not annotated and there is no prior knowledge of the template structures available at document creation, the solution developed must consider these constraints. Importantly, it is only possible for a human agent to understand which segments constitute template by analyzing the entire content and subsequently identifying recurring patterns. This strategy, apart

from being subjective to the annotator's opinion, is resource-dependent and, therefore, an obstacle in the generalized identification of boilerplate content. As a result, the solution developed should allow the identification of boilerplate from any data collection without being dependent on the annotation of the data.

Research Question 2: Given the development of a solution, does the defined process allow for the employment of scenario-specific techniques? If so, which techniques within the solution result in a more adequate identification of boilerplate content for the present case study? In order to be able to employ the solution developed in this work in more comprehensive case studies, beyond healthcare-related complaints, this thesis aims to define a generalized process pipeline that allows the selection of case-specific techniques for various implementation scenarios. Additionally, upon the definition of such a solution, various techniques must be explored, specifically for processing ERS's content, to understand which approach yields better outcomes.

Research Question 3: Does diminishing noise in data enable the effective employment of models in the data, specifically for classification? Upon the development of the solution, it is necessary to understand the impact the detection of boilerplate structures in the textual data and its subsequent elimination from each document has on the automatic processing of complaints. Furthermore, it is also important to comprehend if the solution outlined in this thesis enables the utilization of more content and data entries in the classification of each complaint procedure.

1.3 Document Structure

In addition to the introduction, this dissertation is structured in five other chapters. In Chapter 2, the fields of research on which the solution developed relies are explored. In Chapter 3, research related to the areas of work this dissertation touches upon is depicted. In Chapter 4, the problem is explored in depth and the solution proposed in this paper is discussed. Chapter 5 depicts the evaluation and application of the resulting data. Finally, Chapter 6 concludes the project's description and outlines some future work.

Chapter 2

Background

The problem addressed in this thesis is based on the fundamental distinction between template or boilerplate text and user-generated content, making it essential to thoroughly understand the attributes of each and the differences that set them apart. Various Natural Language Processing (NLP) fields offer valuable insights into the nature of textual data, as is the case for Sentence Boundary Detection (SBD), Text Representation, and String Similarity. These research fields within the larger scope of NLP allow for a more in-depth analysis of text structure by identifying significant segments, capturing their meaning, and leveraging these for comparisons, respectively.

This chapter introduces the study domains that underpin the research developed in this dissertation. Each domain is explored in detail to emphasize its significance, and state-of-art techniques within each field are examined. Subsequent work development builds on the background information outlined in this chapter.

2.1 Sentence Boundary Detection

Sentence Boundary Detection (SBD) is an essential data processing stage in NLP, aiming to correctly split text into individual sentences. Accurately identifying sentence boundaries helps break down a text into meaningful segments, enabling further analysis and processing at the sentence or segment level. Notably, despite the name of this field of research, the segments identified by these techniques may not be sentences and rather segments of text that present cohesive units for analysis, which can span beyond conventional grammatical sentence boundaries. This broader approach allows for a more comprehensive language understanding.

As it is classically defined, Sentence Boundary Detection is a task to identify the boundaries of sentences within a given text or document. The objective is to accurately determine where the beginning of a sentence is, as well as its ending. When the end of the sentence does not correspond to the end of the document, it must be followed by the beginning of the following sentence. In written language, sentences are typically separated by punctuation marks such as periods, question marks, or exclamation marks. However, these punctuation marks can sometimes be ambiguous, leading to difficulty in identifying sentence boundaries.

Indeed, there are widely used systems for sentence segmentation that have gained recognition and popularity in Natural Language Processing, that employ a rule-based approach to the task. NLTK's Punkt tokenizer [16] aims to address incorrectly predicted sentence boundaries, particularly in cases where periods appear after abbreviations. To overcome this issue, the tokenizer employs various strategies to detect abbreviations by analyzing the characteristics of each token, such as its length, collocational relationships, the presence of internal periods, and instances of abbreviations without ending periods. CoreNLP [21] also offers a rule-based approach for predicting sentence boundaries. It relies on events like periods, question marks, or exclamation marks to determine the end of a sentence.

Stanza [29], a multilingual system, takes a different approach using a BiLSTM model to handle tokenization and sentence splitting jointly. It treats these tasks as a character sequence tagging problem and predicts whether a character signifies a token's or a sentence's end. Additionally, SpaCy [13], another multilingual system, employs pre-trained models that utilize technologies such as CNN and transformers to tokenize the text and identify sentence boundaries accurately.

In [30], the notion of Sentence Boundary Detection being a 'solved' task in NLP was questioned. The paper found that state-of-art SBD systems had impressive performances on well-structured and formal texts adhering to strict syntactic rules. However, as the data sources had expanded and the prevalence of informal language increased, the limitations of existing SBD systems became apparent. Informal language often exhibits unconventional sentence structures, non-standard punctuation usage, and lacks clear sentence boundaries, posing challenges for traditional methods. Furthermore, domain-specific texts, such as technical or scientific documents, may contain specialized jargon and terminologies that require domain knowledge for accurate sentence segmentation. Therefore, the evolving landscape of language data necessitated the development of more robust and adaptive SBD systems that can handle the nuances and variations present in such data. Consequently, the research focused on this task persisted in exploring the questions raised by [30], and there was a shift to applying these techniques in textual data that does not follow classic structures and rules.

Some of this research chose to focus on the development of SBD systems expressly targeted to domain-specif data. There was particular interest in data within the legal context. Several works have been conducted to improve SBD performance in this domain, some of which will be briefly discussed. [34] explored tokenizers and SBD systems' performance in English legal domain using a curated dataset of adjudicatory decisions. As part of their work, the authors also developed a comprehensive set of annotation guidelines for sentence boundaries in legal texts, which were made publicly available. These guidelines were the foundation for future annotation processes and further research in this domain. [33] experimented with Punkt Model and Conditional Random Field (CRF) and Neural Network (NN) models on a legal dataset, achieving impressive F1 scores with the developed models. Similarly, [11] curated a German legal dataset and established a baseline performance of existing SBD systems, comparing it to CRF and NN models trained on the said dataset. The findings of this implementation reinforced the observation that off-the-shelf SBD systems tend to perform sub-optimally on legal data.

Other works [26, 35, 7, 25] also focused on the development and availability of comprehensive annotated datasets as a means to promote transfer learning and to facilitate the progress of the development of SBD across other domains and, importantly, across many languages.

Finally, [5] curated a dataset of legal data in six different languages and demonstrated that existing SBD methods perform poorly on multilingual legal data. Subsequently, mono and multilingual models were trained and evaluated, achieving remarkably high evaluation metrics and demonstrating state-of-art performance. Importantly, the authors tested the multilingual models in a zero-shot experiment on unseen Portuguese data and verified that the transformer model implemented in the paper outperformed the baseline trained on Portuguese texts.

2.2 Text Representation

One of the critical challenges in NLP is the complex, ambiguous, context-dependent, and inherently textual nature of human language, as most techniques employed to process text generally are constricted to numerical data as input. The need to translate between the two formats arises as a consequence, allowing the textual data to be processed by the designated models while conveying their meaning and context. Text representation refers to the expression of textual data through n-dimensional numerical vectors and, as such, bridges the gap between human language and machine learning techniques possibly applied to it. It is, therefore, an essential phase in developing any NLP approach and a critical field research area.

Different tasks may call for different units of text to be represented, from single words to more complex textual structures such as sentences, paragraphs, or even whole documents. The techniques employed with each unit also differ. Word embeddings — that is, the vector representation of words — describe the least complex of the mentioned textual structures and have, throughout the years, evolved to achieve representations that encode the meaning of words with a high level of accuracy. Sentences, paragraphs, and document embeddings, however, still have not achieved this effectiveness, but strides have been made more recently toward this goal, as shown below.

2.2.1 Word Embeddings

Word embeddings are fundamental tools for applying deep learning techniques to build solutions for NLP tasks due to their ability to capture semantic and contextual information about words. Embeddings derived from neural networks will center this section, as they represent words highly effectively.

The approaches based on neural networks produce word embeddings that map words as dense vectors in a continuous feature space. These techniques consider the words that neighbor a given word and the effects they may have on its meaning. As such, word embeddings within a defined vocabulary relate the meaning of the words, the relation between them, and the context in which they appear. The resulting representation of words in an n-dimensional space is unique for each word, and the proximity of these vectors is directly connected to the similarity of meaning of the context in which the correspondent words appear; that is, words that appear in similar contexts

tend to have similar vector representations. Additionally, resulting word embeddings retain several desirable properties: semantic similarities, analogical reasoning, and contextual understanding. Figure 2.1 illustrates some of these properties.

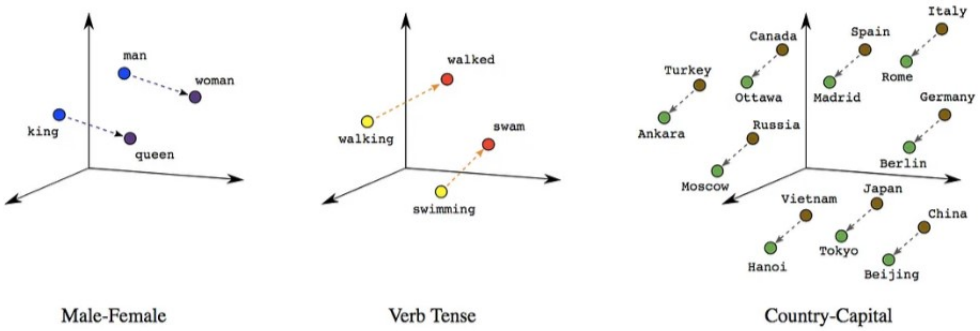


Figure 2.1: The vector difference between "king" and "queen" is similar to the vector difference between "man" and "woman." This characteristic is also illustrated between the pairs of words in the Verb Tense and Country-Capital feature spaces [1].

Neural network-based embeddings have emerged as state-of-art methods for generating text representations of single words. Generally speaking, these embeddings can be pre-trained and either directly employed to the task at hand or fine-tuned. Some common pre-trained embeddings are Word2Vec [23] and GloVe [27], fastText [4]. ELMo [28], and BERT [8] are also common methods employed to obtain word embeddings.

Within neural network-based embeddings, methods can commonly be classified into two main categories: static word embeddings and contextual word embeddings. These classifications are based on the influence the context has on the vector of each word. The choice between static and contextual embeddings depends on the specific NLP task and the level of context sensitivity required. Static or non-contextual word embeddings (such as Word2Vec, GloVE, and fastText) assign a fixed vector representation to each word in the vocabulary. These embeddings capture words' general semantic and syntactic properties without considering the specific context in which they appear. In other words, a word corresponds to a single vector, regardless of how the word is employed. For example, the word "address" is represented by a singular vector, regardless if it means "to speak to" or "location."

Contextual embeddings, also called dynamic or contextualized word embeddings, capture the meaning of words concerning the specific context in which they appear. These embeddings provide different vector representations for a word based on its varying context. So they can handle polysemy (words with multiple meanings) and capture fine-grained semantic relationships. They are particularly useful for tasks like named entity recognition, sentiment analysis, machine translation, and question answering, where the meaning of words can vary significantly depending on the surrounding text. Contrary to the static embeddings, the representation of the word "address" would be affected by its context, and there would be distinctions between the meanings of the word. ELMo and BERT are language models that allow the construction of word embeddigns

associated with these characteristics. It is worth noting that BERT will be explored in depth later on, as it is heavily employed in longer sequence representations, in addition to singular words.

2.2.2 Sentence Embeddings

In this section, the focus of discussion shifts towards more intricate structures, specifically sentences, paragraphs, and documents, as these represent the key units of textual information analyzed in the present work. While word embeddings excel at capturing the semantic and contextual information of individual words, sentence embeddings aim to encode a sentence's overall meaning and context as a single vector. Hence, the exploration expands to encompass higher-level structures to understand texts' overall meaning and composition better. By extending the scope beyond single words, the solutions presented in this work aim to capture the relationships that emerge at the sentence, paragraph, and document levels. This broader perspective allows for more comprehensive analysis and interpretation of text data, facilitating advancements in various natural language processing tasks, including sentiment analysis, text classification, summarization, and machine translation.

[14] surveys text representations that embed more complex text structures. The paper divides the main embedding techniques into the following categories: Count based; Compositional; Unsupervised; Supervised; Transformer; and Multimodal. Following this organization, this section presents the discussion of the methods described by each category, except for multimodal methods, as the techniques developed in this category are not relevant to the present work.

2.2.2.1 Count Based Techniques

Count-based or statistical methods have been regarded as classic techniques over the years. These methods hold significance primarily because they were among the earliest approaches to representing text sequences. Furthermore, they continue to serve as valuable benchmarks for evaluating the performance of more recent approaches.

Bag of Words The Bag of Words (BoW) approach is based on the principle that the order of words in a sentence can be disregarded, and only the frequency of each word is accounted for. The BoW model produces sentence representations as a matrix, where each column represents a unique word in the vocabulary, and each row represents a distinct sentence in the corpus. Consequently, the matrix's columns will equal the number of words in the vocabulary, and the number of rows will equal the number of documents in the corpus. In this display, every element counts the number of occurrences of that specific word in the sentence. For example, given a corpus composed of the sentences *"I have one orange cat and one black cat."* and *"I have one black dog"*, the corpus' BoW representation is as shown in the Table 2.1.

This type of representation is fairly easy to implement, and it has the particularity of storing the multiplicity of each word. However, it is worth noting that each sentence's vector representation does not have a fixed length as it grows with the vocabulary size. Additionally, BoW loses all

Table 2.1: Example of Bag of Words representation. In this example, *s1* represents "I have one orange cat and one black cat." and *s2* equals "I have one black dog."

	I	have	one	orange	cat	and	black	dog
s1	1	1	2	1	1	1	1	0
s2	1	1	1	0	0	0	1	1

information about the order of words in the sentences and, consequently, does not relate syntactic information.

Bag-of-n-grams Bag-of-n-grams appears as a follow-up to BoW that seeks to store some information about the order of elements inside the sentence. Although this method was employed in character-based n-grams, it is also possible to deploy this strategy with word and syllable-based n-grams. Employed in the previous example, with an $n = 2$ word-based n-gram, that is, resorting to bigrams, the resulting representation is as shown in Table 2.2.

Table 2.2: Example of Bag-of-2-grams representation. In this example, *s1* represents "I have one orange cat and one black cat." and *s2* equals "I have one black dog."

	I have	have one	one orange	orange cat	cat and	and one	one black	black cat	black dog
s1	1	1	1	1	1	1	1	1	0
s2	1	1	0	0	0	0	1	0	1

In contrast to BoW, the vocabulary can be expanded by including individual words and pairs or groups of words, as per the chosen n. However, a potential challenge is the resulting increase in the vocabulary size concerning the number of unique words. This expansion is likely to be substantial, given that incorporating word combinations leads to significant growth in vocabulary.

TF-IDF Embedding The Inverse Document Frequency (IDF) score of a term reflects the proportion of documents — that may incorporate smaller units of text, such as sentences — in the corpus that contain that particular word. The IDF equation can be expressed as:

$$IDF_{w_i} = \log \frac{|D|}{|\{d : w_i \in d\}|}$$

In this equation, $|D|$ represents the total number of documents in the corpus, and $|\{d : w_i \in d\}|$ represents the number of documents d that contain the word w at index i in the vocabulary. The equation values terms unique to a few documents in contrast to terms common across all documents.

Term Frequency (TF), on the other hand, is a measure that reflects the ratio of the number of times a term appears in a document to the total number of terms in that document. Consequently, if a word frequently appears throughout the documents of a specific corpus, its TF value will be high, whereas its IDF will be low, suggesting that the word may not be a meaningful representation

of the documents it is found in. Conversely, if a word is infrequent, occurring only in a restricted number of documents, its TF will be low, but its IDF will be higher.

TF-IDF, in contrast to Bag of Words, focuses on highlighting the most meaningful terms that accurately capture the semantic content of a document. The technique assigns higher importance to words that occur less frequently on the corpus, in contrast to those that are common in the same set of documents. Language-related words that tend to be a frequent occurrence in text, such as "the," "as," and "of" in English, are said to be non-relevant. The basic concept aims to provide more weight to the important terms that are included in the document, even if they are not often found across the corpus.

2.2.2.2 Compositional Methods

Compositional methods refer to techniques that build sentence or document representations based on the representations of their constituent elements. [24] defines compositionality as the concept through which the meaning of complex expressions is determined by the meanings of their constituent expressions and the rules leveraged to combine them.

The composition function is the method through which the elementary representations are combined. Composition functions can employ a number of different operations, as illustrated in Figure 2.2. The figure displays an example in which a sentence vector is obtained through the addition of the vectorial representation of its constituents, namely words. On the other hand, the figure also shows the creation of a sentence representation through the concatenation of its words. Composition functions range from simple operations, such as addition, concatenation, or averaging, to more complex operations considering syntactic and semantic relationships between words.

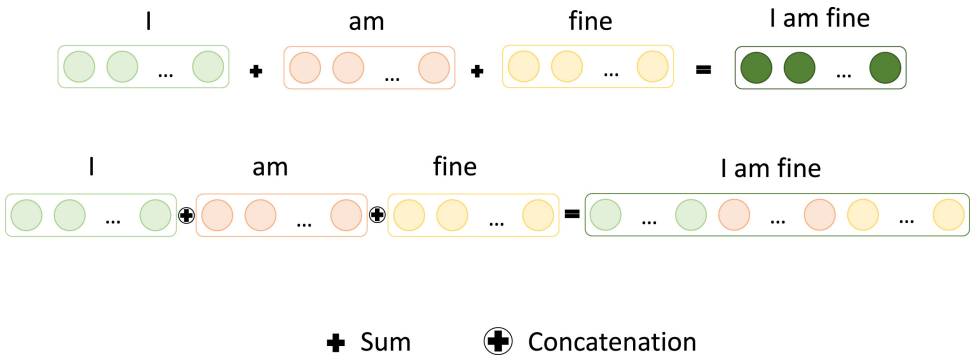


Figure 2.2: Two different approaches to apply compositionality for obtaining sentence representations: sum (top) and concatenation (bottom) [14].

2.2.2.3 Unsupervised Methods

Unsupervised methods for sentence embedding leverage unsupervised learning algorithms and patterns inherent in the text to capture the semantic meaning and relationships between sentences.

Within the unsupervised methods category, the techniques can be partitioned into two distinct classes: Paragraph vector and Encoder-Decoder based.

Paragraph vector based The Paragraph Vector [19], also known as Doc2Vec, was proposed as an unsupervised method for generating embeddings for sentences or paragraphs. It was introduced as an extension of the popular word embedding model, Word2Vec [23]. Paragraph Vector captures the semantic meaning of entire sentences or paragraphs, as opposed to Word2Vec, which only learns continuous representations for individual words.

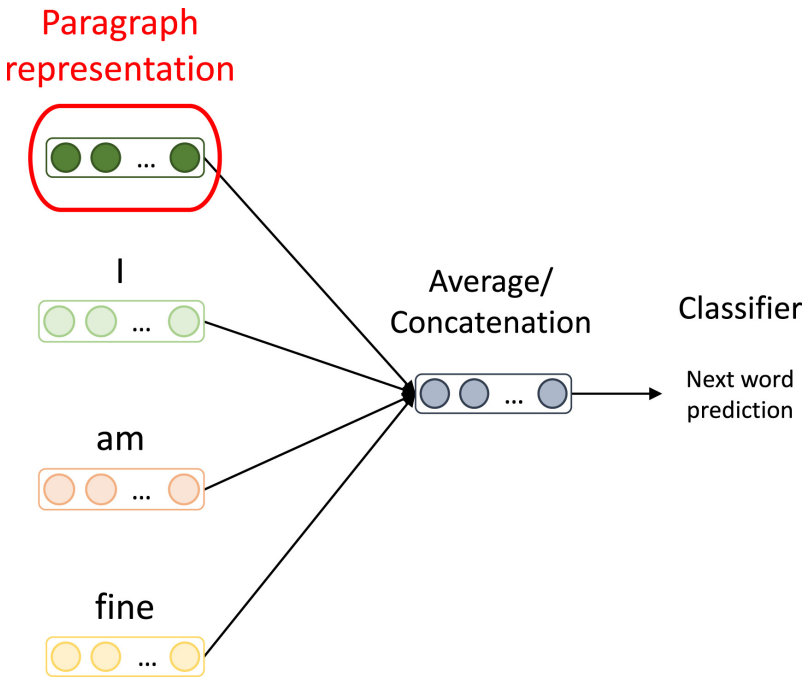


Figure 2.3: Paragraph Vector [19] framework [14]

[19] introduces the Paragraph Vector approach as a method that combines a paragraph representation with the embeddings of its constituent words. The architecture, depicted in Figure 2.3, makes use of the paragraph representation and averages or concatenates it with the representations of known words, used as contextual information, in order to predict the following word in the text. The paragraph vector is initialized randomly and then trained by predicting the following word in the sentence and corresponding word embeddings. Researchers and practitioners have since explored variations and improvements to the original approach, further enhancing the effectiveness and applicability of paragraph vector-based methods in NLP.

Unsupervised encoder–decoder methods Unsupervised encoder-decoder methods are a class of approaches that utilize an architecture consisting of an encoder and a decoder. This encoder-decoder architecture, often depicted as an hourglass structure, as shown in Figure 2.4, is commonly employed in tasks like Machine Translation, where the input and output are text sequences.

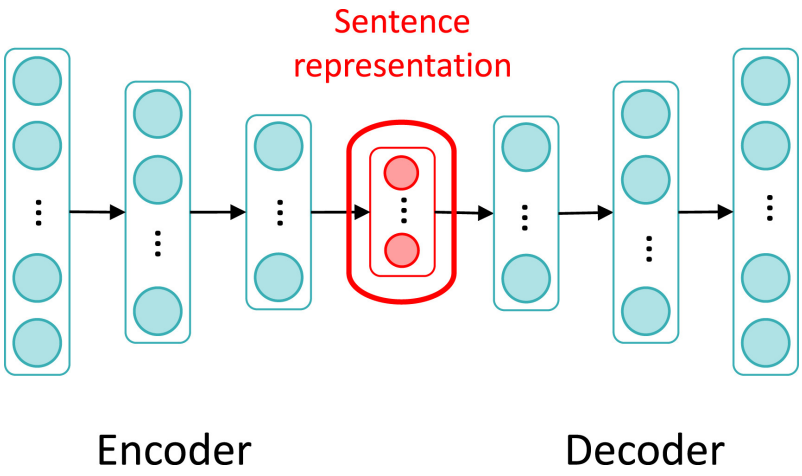


Figure 2.4: Encoder-Decoder architecture [14]

By leveraging this architecture, unsupervised encoder-decoder methods can effectively learn meaningful sentence embeddings. The encoder processes the input text, capturing its semantic information and compressing it into a condensed latent space representation. This latent vector serves as a rich representation of the input sentence. The decoder, in turn, reconstructs the sentence or generates a related sentence using the latent vector as a guide.

One popular method within this category is Skip-Thought [15]. Skip-Thought extends the encoder-decoder framework to capture the meaning of a sentence by training the encoder to generate a sentence's previous and subsequent sentences as output. The encoder learns to encode the current sentence in a way that facilitates the reconstruction of the adjacent sentences by the decoder. This approach allows the model to learn a comprehensive representation of the original sentence, considering its surrounding context.

2.2.2.4 Supervised Methods

Unlike unsupervised methods, supervised methods are trained to tackle specific tasks using labeled datasets created by human annotators. A distinguishing characteristic of these methods is that textual representations typically reside in the final layers before the prediction layer. Different Neural Networks are employed to obtain sentence embeddings through supervised learning.

Supervised learning methods offer the advantage of directly leveraging labeled data to optimize sentence embeddings for specific tasks. By training on labeled datasets, these methods can learn to generate highly informative embeddings relevant to the task's objective. However, they require annotated data, which may be costly or time-consuming. Additionally, the quality and diversity of the labeled dataset play a crucial role in the performance of supervised methods.

2.2.2.5 Transformer Based Methods

Transformer-based methods refer to a class of models and techniques based on the Transformer architecture, which was introduced in the seminal paper "Attention Is All You Need" [37]. The

Transformer architecture, illustrated in Figure 2.5, is a neural network architecture that incorporates the following key components: an encoder-decoder architecture, a self-attention mechanism, attention heads and multi-head attention, and positioning encoding.

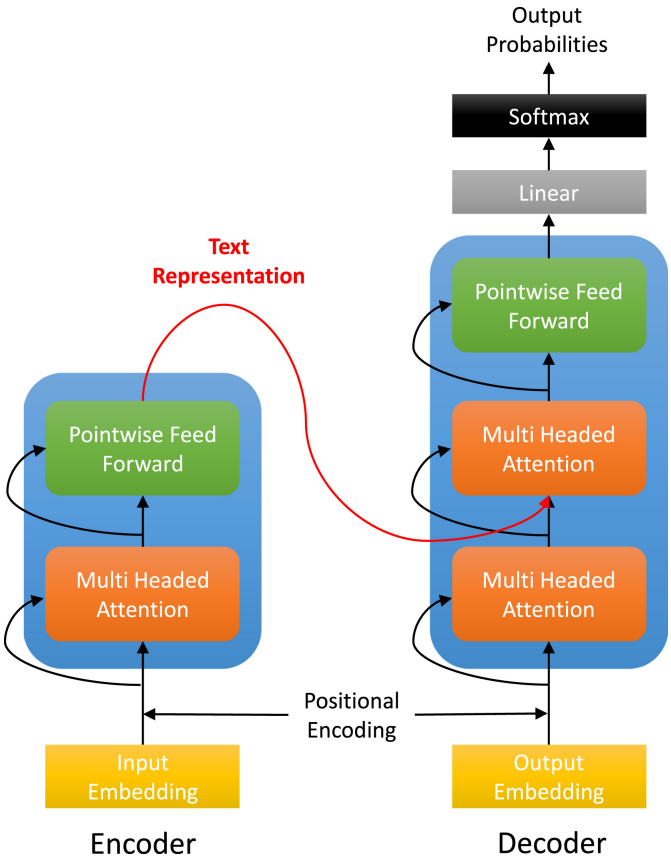


Figure 2.5: Transformer architecture [14]

Comprehending each component is crucial to capture the architecture fully. Beginning with the encoder-decoder architecture, it is vital to differentiate the roles of the encoder and the decoder. The encoder is responsible for obtaining textual representations or embeddings of the input sequence, while the decoder is used for generating word predictions when needed. Both the encoder and decoder work in parallel, which enables efficient computation. The self-attention mechanism computes the importance or attention weights for each position in the input sequence based on its relationship with all other positions, which allows the model to capture long-range dependencies and contextual information effectively. Self-attention is employed in both the encoder and decoder layers of the Transformer.

Transformers often employ multiple attention heads in each self-attention layer. Each attention head attends to different parts of the input sequence, allowing the model to capture different dependencies. Multi-head attention enables the model to capture diverse patterns and facilitates better representation learning. Positional encoding is a technique used to inject information about the relative or absolute positions of the elements in the input sequence. These positional encodings are added to the input embeddings, giving the model a sense of order and position. Transformers

require a way to incorporate positional information since they do not rely on recurrent connections.

Transformer-based methods have revolutionized the field of NLP and have become the *de facto* choice for many sequence-related tasks due to their ability to capture long-range dependencies, effectively model context, and achieve strong performance with large-scale pre-training and fine-tuning. Transformer-based methods have been successfully applied to various NLP tasks, including machine translation, language modeling, sentiment analysis, named entity recognition, question answering, and text summarization. Some important examples of Transformer-based models include BERT, Generative Pre-trained Transformer (GPT), GPT-2, and GPT-3.

Bidirectional Encoder Representations from Transformers (BERT) [8] employs a Transformer architecture consisting of an encoder with multiple stacked layers. Each layer has self-attention mechanisms and feed-forward neural networks. The language model uses a bidirectional approach, which allows it to consider both the left and right context of each word during pre-training.

BERT is pre-trained on large-scale corpora using unsupervised learning. It learns to generate contextualized word representations by predicting missing words (masked language modeling) and determining whether two sentences appear consecutively in the original text (next sentence prediction). After pre-training, supervised learning can fine-tune BERT on specific downstream tasks. The model is further trained on task-specific labeled data, adapting its parameters to the target task. Fine-tuning allows BERT to leverage its pre-trained knowledge and contextualized word embeddings to improve performance on various NLP tasks, such as text classification, named entity recognition, question answering, and sentiment analysis. In the literature, the BERT architecture has been applied in various contexts.

A version of BERT that holds particular significance for sentence representation is Sentence-BERT (SBERT) [31]. SBERT is an extension of BERT that involves supervised training on pairs of sentences using Siamese networks (two BERT architectures with tied weights) and triplet network structures. This approach enhances the representation learning process, particularly for sentence-level semantics and similarity tasks. It also allows for fine-tuning the pre-trained models to adapt them to specific downstream tasks or datasets. Fine-tuning involves further training the model on a task-specific dataset, which enables it to learn task-specific patterns and nuances, thus improving performance on the target task.

2.3 String Similarity

String similarity is a fundamental concept in the field of Natural Language Processing that quantifies the resemblance between two strings of characters. It involves various functions that measure the similarity based on factors such as the number of edits required to transform one string into another, the overlap of characters or n-grams, the cosine of the angle between string representations, the common prefix, or the number of different characters at corresponding positions. These metrics play a crucial role in applications like spell-checking, plagiarism detection, information retrieval, and data mining, where understanding the degree of similarity between strings is essential for the task's success.

As explained in [42] the similarity between two strings is measured through functions. These functions can be broadly classified into edit distance-based similarity, and token-based similarity. These categories will be explored in depth in the following sections.

2.3.1 Token-based similarity

The token-based similarity functions model each string as a set of tokens and measure the similarity between strings through the common tokens shared by their corresponding token representation. A higher number of common tokens reflects a higher similarity between strings, as the order of said tokens in the strings does not matter. The Jaccard index, the Sorensen-Dice, and Cosine similarity are some functions under this category. These functions are discussed next.

Jaccard Index Falling under the set similarity domain, this metric considers the presence or absence of individual tokens between sets. It relates the ratio between the number of common tokens and the total number of unique tokens. It is expressed in mathematical terms for the sets A and B by an equation where the numerator is the intersection between the sets, the shared tokens, and the denominator is their union or all the unique tokens.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Sorensen-Dice This metric, defined as a set similarity function, also considers the presence or absence of individual tokens between sets. It relates the ratio between the number of common tokens and the total number of tokens. It is expressed in mathematical terms for the sets A and B by an equation where the numerator is twice the intersection of the sets, the shared tokens, and the denominator are all tokens in both sets. The doubling of the intersection of sets takes into consideration that if a token is present in both sets, its total count is twice the intersection.

$$DICE(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

Cosine similarity As the nomenclature suggests, cosine similarity measures the cosine of the angle between two vectors in a vector space. These vectors are numerical representations of the strings, known as sentence embeddings discussed previously in Section 2.2.2. It is expressed in mathematical terms for the sentences $s1$ and $s2$ by the following equation:

$$COS(s1,s2) = \frac{|s1 \cap s2|}{\sqrt{|s1| \cdot |s2|}}$$

2.3.2 Edit distance-based similarity

The edit distance-based similarity approaches measure the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into another. More

needed operations correspond to a less similar metric between the two strings. Some of the algorithms under this category are the Hamming distance, the Levenshtein distance, and Jaro-Winkler. These functions are discussed next.

Hamming distance The computation of this distance involves overlapping one string with another and identifying the locations where the strings differ. It is important to note that the classical implementation of this distance metric was originally designed to handle strings of equal length. However, certain implementations overcome this limitation by adding padding either at the beginning or end of the strings. Regardless, the fundamental principle remains the same: the objective is to determine the number of positions where one string differs from the other.

Levenshtein distance The computation of this distance involves determining the number of edits required to transform one string into another. The allowed transformations include insertion (adding a new character), deletion (removing a character), and substitution (replacing one character with another). Through these operations, the algorithm attempts to modify the first string in order to match the second string. The resulting count of edits provides the edit distance between the two strings.

Jaro-Winkler This algorithm assigns higher scores to two strings under two conditions: firstly, if they contain the same characters within a certain distance from each other, and secondly, if the order of the matching characters is the same. Specifically, the algorithm determines the distance for finding similar characters by subtracting 1 from half the length of the longest string. For example, if the longest string has a length of 5, a character at the start of the first string must be found before or at the position of the second character in second string to be considered a valid match. This characteristic makes the algorithm directional and assigns a higher score when the matching starts from the beginning of the strings.

2.4 Summary

In this chapter, an exploration of key domains to the understanding of the solution proposed in this work has been conducted. These domains are Sentence Boundary Detection, Text Representation, and String Similarity. Sentence Boundary Detection aims to identify sentence boundaries within a given text, further segmenting text into smaller units that convey interesting meanings. Text representation is important for the work being developed as it transforms textual data into numerical vectors, facilitating the nuanced representation of complex text structures. Lastly, String Similarity measures quantify the resemblance between two strings and therefore allow for the comparison between textual segments.

The discussion of these domains serves as the background for the subsequent chapter, which delves into related work, providing a foundation for understanding the current state of the art in the research areas directly connected to the problem explored in this work.

Chapter 3

Related Work

The central concern of this thesis is to ensure the quality of the data collected from the documents associated with the complaints submitted to the ERS. The accurate and effective classification of each complaint process is of paramount importance as it has significant implications for the well-being of patients and the overall quality of health services provided in Portugal. The content of the documents that constitute ERS's data is characterized by the prevalence of boilerplate structures and user-generated content, and, given the aim is to model a process that classifies the complaint content, the boilerplate depicted in these documents constitutes noise in the data.

In this chapter, the importance of data and its quality for the development of Artificial Intelligence (AI) applications is emphasized through the discussion of data-centric AI, an emerging new way of thinking about how to build AI systems that emphasizes the crucial role of data in this research area. Furthermore, the research field of boilerplate removal is also discussed.

3.1 Data-centric AI

There has been a historical emphasis on developing better models while leaving the data largely unchanged in AI. This mode of operating is known as the model-centric approach. Researchers have focused on refining algorithms, optimizing model architectures, and fine-tuning hyperparameters to achieve superior performance. However, this approach often disregarded the potential problems inherent in the data, such as inaccurate labels, duplicates, and intrinsic biases. It also often led to an "overfitting" phenomenon where models became highly specialized and performed exceptionally well on the training data but failed to generalize to new or unseen data. This narrow focus on model improvement without adequate consideration of the quality and characteristics of the underlying data limited the effectiveness of AI systems in real-world scenarios.

In contrast, the data-centric AI paradigm shifts the attention towards improving the quality and quantity of data used to build AI systems. This approach recognizes that the data itself plays a crucial role in determining the maximum capabilities of a model. By giving paramount importance to data, data-centric AI aims to address the limitations of the model-centric approach and unlock the full potential of AI models. These dynamics are represented in Figure 3.1.

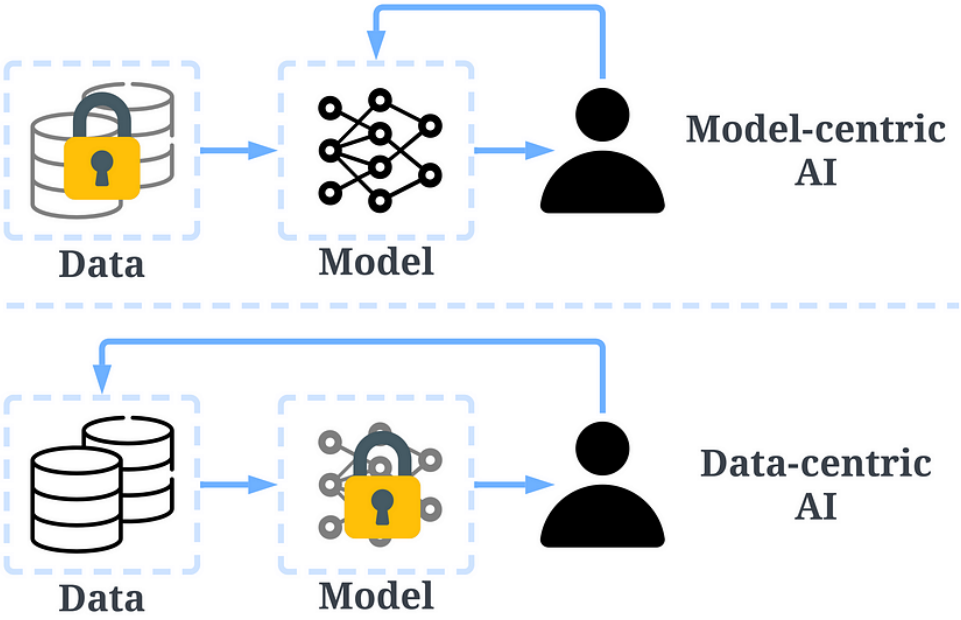


Figure 3.1: Comparison between data-centric AI and model-centric AI [43]

Figure 3.1 provides a visual comparison between model-centric AI and data-centric AI. As explained previously, a model-centric development largely leaves the dataset untouched, as the attention is on the iteration of the model. While this approach encourages advancements in models, it places significant trust in the data. Data is not merely the source upon which AI is built but rather a decisive factor in determining the quality of the model. In recent years, data-centric AI has emerged as a trend that shifts the focus from models to data, recognizing its critical role in achieving superior AI outcomes.

[44] organizes the goals of data-driven AI into three distinct goals: training data development, inference data development, and data maintenance, where each goal is associated with several sub-goals, and each task belongs to a sub-goal. Figure 3.2 illustrates this division into goals and sub-goals defined in the paper. The division is defined as follows:

- **Training data development** aims at gathering and developing training data that, by means of its quality, comprises the most comprehensive source of data in which to train machine learning models. It consists of five sub-goals, these being 1) data collection for compiling unprocessed training data, 2) data labeling for adding useful information to the data, 3) data preparation through the cleaning and transformation of data, 4) data reduction, which aims at improving model results by means of reducing the data size, and 5) data augmentation for enhancing data diversity while foregoing additional data collection.

- **Inference data development** is directed towards creating new evaluation sets in order to achieve an insight enjoying a degree of detail into the model or trigger a specific capability of the model through engineered data inputs. There are three sub-goals in this effort: 1) in-distribution evaluation, 2) out-of-distribution evaluation, and 3) prompt engineering.

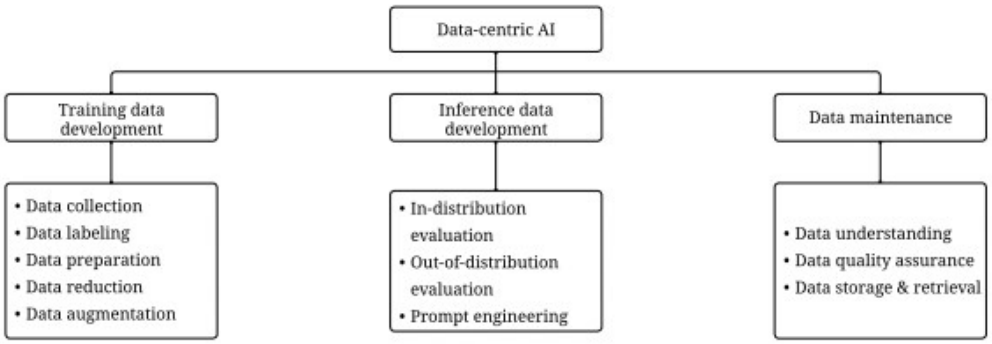


Figure 3.2: Data-centric AI goals and sub-goals [44]

- **Data maintenance** is essential throughout time in a real-world application. Indeed, a dynamic environment calls for the reliance on the perpetual maintenance of data that previously only took place in its creation. It involves three essential sub-goals: 1) data understanding, 2) data quality assurance, and 3) data storage & retrieval.

Within the framework of data-centric AI, noise reduction is taken as a task that is situated within the sub-goal of data preparation. Data preparation involves cleaning and transforming raw data into a suitable format for training machine learning models. Traditionally, this process has required substantial manual effort and trial-and-error approaches, where data researchers had to identify and address issues in the data painstakingly. However, state-of-the-art methods have emerged to streamline and optimize this process, leveraging advanced techniques such as search algorithms. These methods aim to automate and expedite the data preparation process by discovering effective strategies that enhance the quality and usefulness of the data.

Raw data tends to present potential issues, such as noise, inconsistencies, and irrelevant information, thus making it often unsuitable for model training, as these issues can significantly affect the accuracy and reliability of the resulting models. Indeed, if the model is trained on data that contains noise, outliers, or irrelevant features, it may overfit these artifacts, leading to reduced generalizability and performance limitations [41].

This phenomenon can have real-life implications. As noted earlier, humans have implicit biases, which may be expressed subconsciously. Data, as a human creation, often reflect these underlying preconceptions, and therefore the inability to properly treat this data and remove sensitive information such as race, gender, or sexual orientation can result in the engraining of such beliefs in a model's predictions [39]. The applications of said models can, consequently, create systems that further discriminatory beliefs and negatively impact people's lifes.

Furthermore, the performance of the model can be negatively affected if the raw feature values are in different scales or follow skewed distributions, as this can introduce distortions and inconsistencies in the learning process [3]. Hence, the critical process of cleaning and transforming data is indispensable to address these issues and ensure the production of accurate and unbiased models.

3.2 Boilerplate Removal

This work focuses specifically on the existence of boilerplate structures in the data, and as such, boilerplate removal techniques were explored. Identifying and removing boilerplate is one of the possible stages of data cleaning, and, as such, it is discussed in this section.

Boilerplate removal pertains to removing redundant and non-informative content in a text document. It is mainly applied to extracting a text's most relevant and essential information and eliminating all components that fall outside that definition. Boilerplate content can be defined as any repetitive information that does not add further value to the document, such as page headers, footers, legal notices, and ads. This area of research is mainly directed at web pages and HTML structures, which hinders the application of many techniques in the problem at hand [12, 40, 17, 38, 6].

Existing boilerplate removal approaches can commonly be categorized into two groupings. The first set of procedures follows predefined rules and implements tools that target extracting content from web pages previously regarded to possess specific structural and textual properties. On the other hand, the remaining grouping can be described as implementing machine learning approaches using predefined features to separate content from boilerplate.

These methods mainly rely on hand-crafted features for categorization, which leads to models customized to a particular distribution of web pages. Recent research tries to deviate from this formulation, forgoing the definition of hand-crafted rules.

[20] proposes a sequence labeling procedure for boilerplate removal from web pages, BoilerNet. To do so, the issue of content extraction, or boilerplate removal, is modeled as a sequence labeling problem. In this approach, each web page is conceptualized as a sequence of text blocks labeled as either content or boilerplate. Each text block is represented through a sparse vector that encodes the original text and the HTML tags.

Bidirectional LSTMs, the foundation of the BoilerNet architecture, can learn intricate non-local dependencies in sequence models. The vectors mentioned previously undergo an embedding layer transformation into lower-dimensional dense vectors. The subsequent many-to-many bidirectional LSTM layers get the dense representations in succession. A fully connected layer with a sigmoid activation function is employed to compress the concatenated outputs of the final LSTM layer into 1-dimensional vectors and get the final classification probabilities. Binarized cross-entropy is used to train the model. The architecture of BoilerNet is represented in 3.3.

In the end, BoilerNer is shown to either achieve comparable or superior performances to those registered by, at the time, state-of-art architectures[17, 38]. It was also established that, in web pages, the position of the text blocks encodes essential information.

Other works have tried to generalize the definition of boilerplate to domains outside of websites, expanding the methods used in their detection to be deployed in textual data that does not obey an HTML structure.

In [10] the problem of boilerplate detection is reformulated as a motif extraction task. In graph theory, a motif refers to a small, recurring subgraph pattern that appears more frequently than expected in a network. The problem reformulation is established to expand the boilerplate

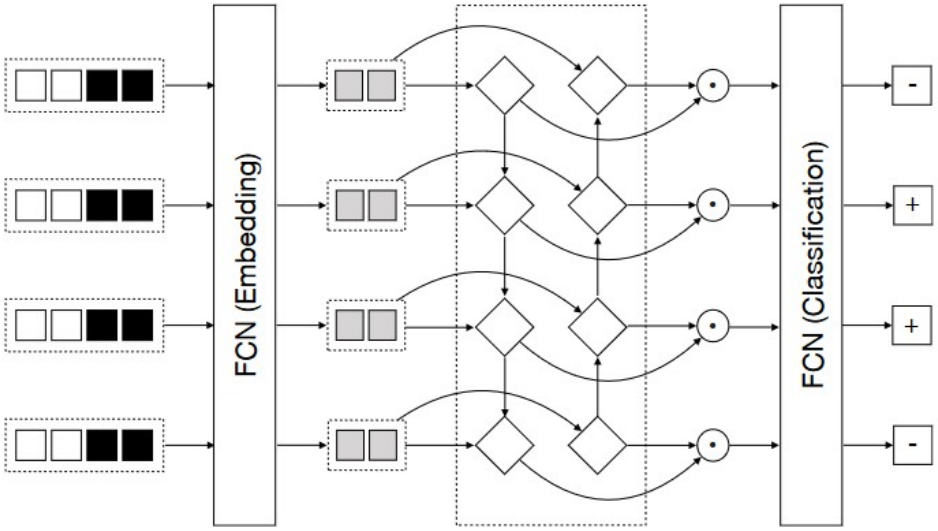


Figure 3.3: BoilerNet architecture with one LSTM layer [20]

definition to include generic text without any previous conception about the type of text or its structure. It is, therefore, a departure from past procedures.

The method developed is processed in two steps. First, the data is submitted to the extraction of all motifs following the user's prerequisites. Motifs are rigid blocks of text separated by gaps with varying lengths. Gaps are sets of joker elements, which can represent any symbol in the vocabulary associated with the data. Five metrics constrain these structures: minimum number of words for each block; total length of all blocks; number of documents they occur in; number of possible gaps, and maximal span of each gap. This work focuses on maximal motifs, defined as motifs such that no other motif covers the same strict positions by being more specific, that is, being longer or converting a joker into a fixed symbol.

Later, the extracted motifs are clustered and added as new features in the extended vector space model representing the documents. The motifs' original words are also removed from the documents where they occur.

3.3 Summary

This chapter focuses on data-centric AI, which recognizes the crucial role of data in building effective AI systems. The chapter further discusses the goals of data-driven AI and emphasis its relevance to the problem explored in this thesis. The significance of data and data quality is highlighted, as is the importance of boilerplate removal in the ERS data. Various techniques have been explored for boilerplate removal, enhancing the quality of data for subsequent tasks.

This work discusses state-of-art techniques in these research areas but departs from these in the next chapter, where a specific approach to address noise reduction through boilerplate removal

in data does not allow for implementing these techniques, which rely on the existence of large annotated datasets.

Chapter 4

Solution Approach

The present chapter focuses on understanding the case study at hand. The sections that follow explore the challenges existing noise poses in ERS' data. A comprehensive and in-depth noise analysis identifies the various factors contributing to data noise. This analysis sheds light on the sources, patterns, and characteristics of noise, enabling a deeper understanding of its impact on the research problem. Additionally, the chapter outlines the research steps undertaken to address the problem. By thoroughly investigating and exploring potential solutions, the research aims to mitigate the effects of noise. These sections provide valuable insights into the research process and demonstrate the systematic approach taken to tackle the noise issue, ultimately paving the way for a robust solution.

4.1 Understanding the Case Study

ERS's nationwide reach often entails the submission of tens of complaints daily. These reports can be received through diverse channels, from online forms, emails, and letters, to physical complaint forms, allowing users to choose the most convenient method for submitting their complaints.

When a user submits an initial report, it sets in motion a process of documentation within ERS. This process involves the creation of auxiliary statements and the generation of a comprehensive record of the reported incident and its subsequent handling. One essential aspect of this documentation is the creation of a summary that captures the content of the complaint. This summary provides a concise overview of the issues raised, enabling a quick understanding of the nature of the complaint. These auxiliary statements and summaries, along with other relevant documents, form part of the data that ERS processes to address each specific unresolved issue, represented in Table 4.1.

In addition to the documentation and record-keeping, ERS stores information about the entities participating in processing each complaint. This includes data on the individuals filing the complaints and the institutions implicated in the reported incidents. The data stored on these entities include their names, address, and contact information.

Table 4.1: Types of complaint-related documents

ALEGACOES	Healthcare provider's allegations in response to the complaint
ANEXO_EXP	Document attachments
ANEXO_RSP	Document attachments
ANEXO_ROL	Document attachments
OFICIO	ERS's response to the complaint and further communications
RECLAMACAO	Complaint description as exposed by the complainee
SINTESE_ERS	Complain summary written by ERS
SINTESE_PRESTADOR	Complain summary written by the involved healthcare provider

It is important to note that given the data's sensitive nature and personal information, the data utilized in this project cannot be publicly shared. This precaution is taken to ensure the privacy of the individuals involved in the reports, and, therefore, the data presented in this document will be strategically censored, as shown in Figure 4.1. Only non-identifiable information will be revealed, allowing for a comprehensive understanding of the issue while upholding the privacy and rights of the individuals who have shared their experiences and concerns.

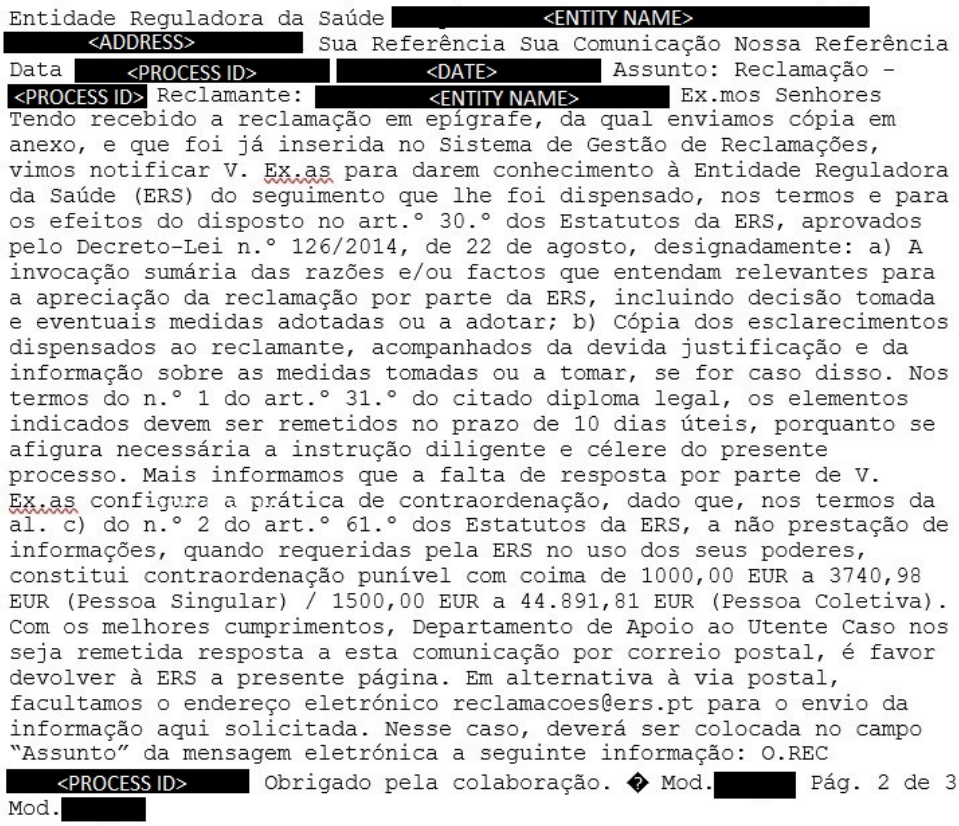

Entidade Reguladora da Saúde <ENTITY NAME>
<ADDRESS> Sua Referência Sua Comunicação Nossa Referência
Data <PROCESS ID> <DATE> Assunto: Reclamação -
<PROCESS ID> Reclamante: <ENTITY NAME> Ex.mos Senhores
Tendo recebido a reclamação em epígrafe, da qual enviamos cópia em anexo, e que foi já inserida no Sistema de Gestão de Reclamações, vimos notificar V. Ex.as para darem conhecimento à Entidade Reguladora da Saúde (ERS) do seguimento que lhe foi dispensado, nos termos e para os efeitos do disposto no art.º 30.º dos Estatutos da ERS, aprovados pelo Decreto-Lei n.º 126/2014, de 22 de agosto, designadamente: a) A invocação sumária das razões e/ou factos que entendam relevantes para a apreciação da reclamação por parte da ERS, incluindo decisão tomada e eventuais medidas adotadas ou a adotar; b) Cópia dos esclarecimentos dispensados ao reclamante, acompanhados da devida justificação e da informação sobre as medidas tomadas ou a tomar, se for caso disso. Nos termos do n.º 1 do art.º 31.º do citado diploma legal, os elementos indicados devem ser remetidos no prazo de 10 dias úteis, porquanto se afigura necessária a instrução diligente e célere do presente processo. Mais informamos que a falta de resposta por parte de V. Ex.as configura a prática de contraordenação, dado que, nos termos da al. c) do n.º 2 do art.º 61.º dos Estatutos da ERS, a não prestação de informações, quando requeridas pela ERS no uso dos seus poderes, constitui contraordenação punível com coima de 1000,00 EUR a 3740,98 EUR (Pessoa Singular) / 1500,00 EUR a 44.891,81 EUR (Pessoa Coletiva). Com os melhores cumprimentos, Departamento de Apoio ao Utente Caso nos seja remetida resposta a esta comunicação por correio postal, é favor devolver à ERS a presente página. Em alternativa à via postal, facultamos o endereço eletrónico reclamacoes@ers.pt para o envio da informação aqui solicitada. Nesse caso, deverá ser colocada no campo "Assunto" da mensagem eletrónica a seguinte informação: O.REC
<PROCESS ID> Obrigado pela colaboração. Mod. Pág. 2 de 3
Mod.

Figure 4.1: Example of censorship of sensitive information on the data. This file illustrates communication between ERS and a hospital about the proceedings within a complaint.

4.1.1 Noise Analysis

The content derived from the communications between ERS and the intervening entities plays a vital role in analyzing and classifying user-submitted complaints. As such, it is essential to acknowledge the presence of noise within this data, striving to comprehend its origins and impact on the data. Understanding and addressing these noise sources is crucial for effectively analyzing the data and gaining meaningful insights that can contribute to resolving user complaints and improving regulatory processes.

In the context of data analysis, noise refers to any irrelevant information that can inhibit the accurate interpretation of the data by automatic means. For this project, the previously mentioned data has two primary sources of noisy interference: Optical Character Recognition (OCR) errors and boilerplate content.

4.1.1.1 Optical Character Recognition

The regulatory body accepts various entry formats that often involve non-textual representations of documents, including scanned paper and images. For this data to be processed as intended, it must be transformed into a machine-readable format, and, to achieve this, the data undergoes an OCR process.

OCR is a procedure used to extract textual content from image files. By analyzing the shapes and patterns of characters, OCR enables the conversion of these visual elements into text that computer processes can interpret. The output of OCR can be characterized by mistakes, especially when the quality of the scanned image is poor. Additionally, OCR encounters particular challenges when applied to the Portuguese language, for example, the inclusion of characters that do not exist in other languages, like *ç*, and accentuation, which further hinders the quality of the resulting text.

Consequently, the data resulting from the application of OCR techniques is riddled with errors, as shown in Figure 4.2. The example illustrates how egregious OCR noise can be in the data, as it results in a text that is unrecognizable as Portuguese, even for human readers.

While this thesis does not focus on the development of post-OCR techniques or methods that correct the text produced by this procedure, the errors detailed here still constitute noise in data. This noise, apart from hindering the usage of such data in automatic complaint processing tasks, also is a limitation to the solutions explored in this thesis, and as such, an approach to preprocess the data and minimize its impact is described.

4.1.1.2 Boilerplate

Boilerplate or template text plays a significant role in numerous documents across various domains. It encompasses standardized content that is frequently reused, allowing for consistency and efficiency in document creation. Although templates often have predefined information fields in their structure where context-specific information can be inserted, these standard structures remain recognizable, making them easily identifiable as the same textual segment.

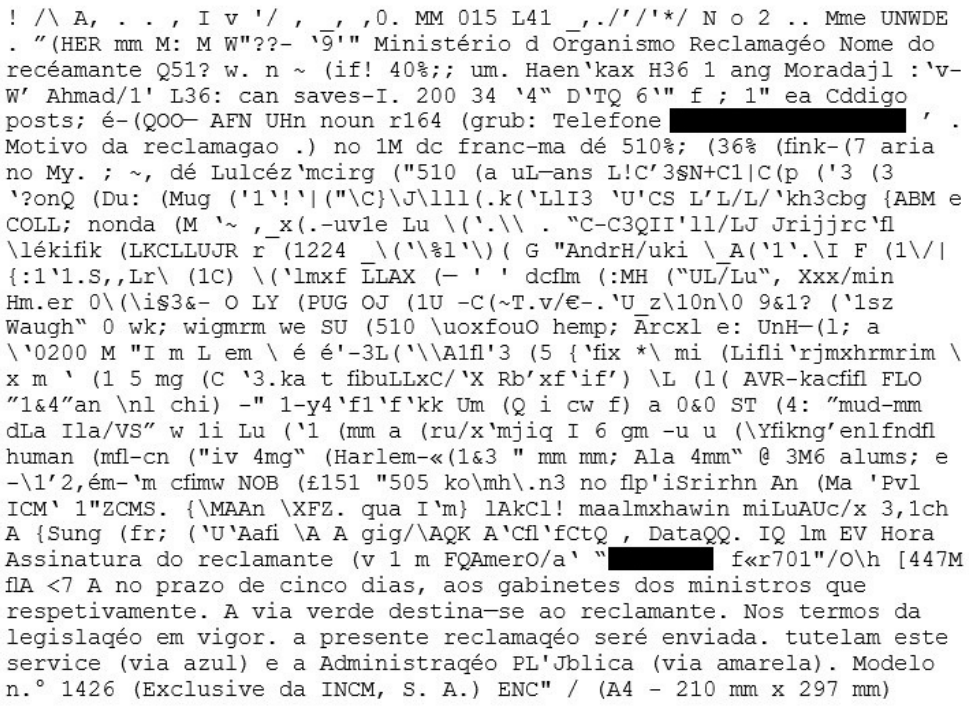


Figure 4.2: Example of data resulting from OCR

From a data analysis perspective, these segments of boilerplate can be considered noise sources. They introduce blocks of content that remain constant in format and substance across different instances, thereby lacking relevant information for deeper understanding and analysis of the data. Consequently, these repetitive layouts can hinder the process and all subsequent tasks when attempting to extract significant insights from documents.

The communications stored by ERS exhibit a notable abundance of varied boilerplate content. Given the diverse nature of communication channels and the various types of content conveyed, the regulatory body relies on many standardized structures to facilitate effective and consistent messaging, both to and from other entities. The diversity of communication channels and types of content manifests into a large variety of templates deployed, some of which are illustrated in Figure 4.3.

Due to its ubiquitous nature and diverse manifestations, detecting boilerplate in the ERS data is not straightforward. This thesis aims to delve into the complexities associated with boilerplate identification and removal, considering the vast array of forms it can assume. The research explores techniques to develop effective methodologies and algorithms that can accurately distinguish and eliminate boilerplate content, enabling more efficient document processing.

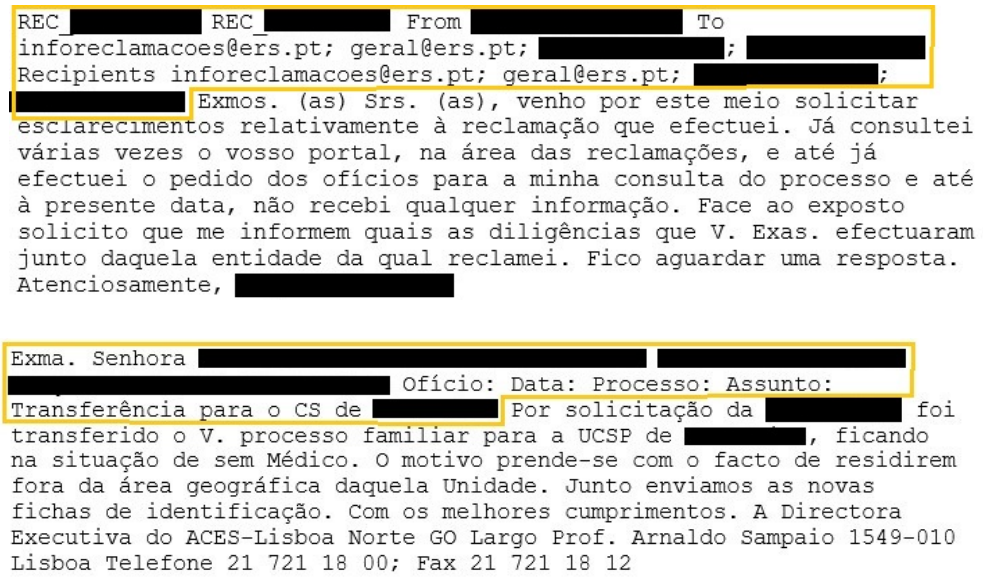


Figure 4.3: Examples of different boilerplate in documents

4.2 Data Preprocessing

Data preprocessing is essential to ensure data analysis quality and facilitate the deployment of future methods onto the data. Its purpose is to convert raw data into a format suitable for further operations, enhancing it and eliminating inconsistencies or errors.

One of the primary considerations concerning the quality of the explored data is the presence of OCR errors. As discussed previously, OCR techniques are prone to introducing errors when converting scanned images or documents into editable text. Its impact on the data has already been analyzed, having resulted, in some examples, in incomprehensible text, even for a human reader. As such, the preprocessing procedure mainly focuses on post-OCR strategies. These techniques look to minimize the mistakes introduced by the OCR procedures on the content by correcting any instance identified as a grammatical error.

The algorithm developed for the preprocessing stage, illustrated in Algorithm 1, starts by eliminating null entries. It proceeds to operate tokenization on the content of each row of the dataset by splitting each document in all empty spaces between characters. It then iterates over each token and checks whether it matches any of the following elements: a sequence of special punctuation, a number, an acronym — any sequence of capitalized letters —, an email, a file name, or a URL link. A token is eliminated if it is a special punctuation sequence, as these are immediately assumed to be errors introduced by OCR techniques; otherwise, the token remains unchanged.

If it does not correspond to any of the aforementioned elements, the token is spell-checked. In order to spell-check each token, the process leverages a European Portuguese variation of the Hunspell dictionary, an open-source spell-checker widely used and available for an immense range

Algorithm 1 Solution preprocessing stage

```
1:  procedure TEXT_PREPROCESSING(document)
2:      tokens = tokenize document
3:      for all token in tokens do
4:          if token == a string of special punctuation then delete token
5:          else if token == ( number or acronym or email or file name or URL link ) then
6:              continue
7:          else
8:              new_token = spell-check token
9:              if token is correct then continue
10:             else
11:                 distance = DICE(token, new_token)
12:                 if distance > 0.7 then continue
13:                 else
14:                     token = new_token
15:                 end if
16:             end if
17:         end if
18:     end for
19:     return tokens
20: end procedure
```

of languages. Projecto Natura[1] developed the European Portuguese dictionary employed in this process, from which the latest version was chosen[2]. The implementation of Hunspell used was sourced from *spylls*[3], a Python library that ports prominent spellchecker and allows for their easy deployment.

In case the word exists, it is, once more, left unchanged; else, the first suggestion presented by the dictionary is selected, and the distance between the suggested word and the original token is calculated. The metric used for this operation is the Sørensen–Dice coefficient, implemented in the *strsimpy* library[4]. If the distance between the words is below a predefined threshold — which, in this case, was 0.7 —, the original token is kept, and if not, it is switched by the dictionary's suggestion. The choice to keep tokens too dissimilar from the suggested word is based on the assumption that these tokens may relay important information that would not be found in the dictionary, such as addresses or entity names.

Ultimately, the resulting data is a cleaner version of the raw data, as most of the errors introduced by OCR are either eliminated or corrected.

[1] `https://natura.di.uminho.pt/wiki/doku.php?id=dicionarios:main`
[2] The dictionary file is named *hunspell-pt_PT-20220304.tar.gz*
[3] `https://git.sr.ht/~martijnbraam/sysls`
[4] `https://github.com/luozhouyang/python-string-similarity/tree/master/strsimpy`

4.3 Pipeline Description

As discussed previously, boilerplate or template text refers to standardized content that is reutilized across multiple documents. Despite allowing context-specific changes, these structures remain recognizable as adaptations of the same components or, in other words, as similar, if not the same, textual segments. The identifiable structure of boilerplate implies that two iterations of the same template will exhibit, in principle, more significant similarity to each other than to other randomly selected textual segments. It is also worth noting that user-generated content, unlike boilerplate, exhibits high variability, and, as such, its segments are unlikely to be identified as similar across data entries.

The approach adopted takes advantage of the distinct features of boilerplate text by recognizing and leveraging the reoccurring structures present throughout the dataset documents. The developed solution is structured into a pipeline, illustrated in Figure 4.4, with the following stages: text segmentation, corpus representation building, template embedding definition, and textual segment labeling. Each stage of the pipeline is explored in depth below.

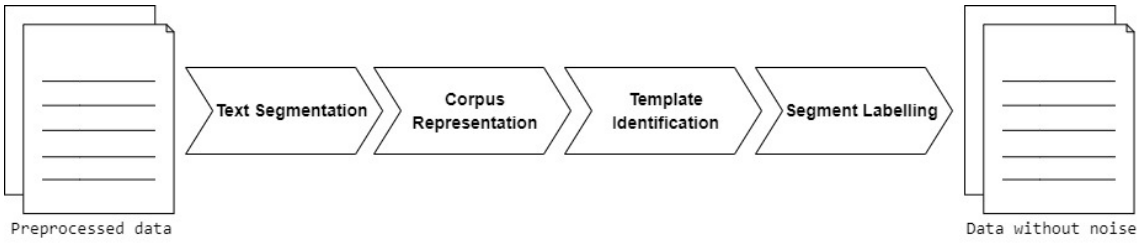


Figure 4.4: An overview of the solution pipeline

Text Segmentation After the preprocessing stage, the data is subjected to segmentation techniques, discussed in Section 4.3.1. The primary objective of this step is to define text segments that accurately capture meaningful structures within the document. Various approaches were explored, and within these, different structures were highlighted, from sentences to constant-length text windows. These structures are essential to understand the document's organization and content comprehensively. Successfully segmenting the text into meaningful units allows for the application of the ensuing stages. This stage of the algorithm is further presented in Algorithm 2.

Algorithm 2 Text Segmentation

```
1: procedure TEXT_SEGMENTATION(document, segmentation_technique)
2:     segments = []
3:     segments = apply segmentation_technique to document
4:     return segments
5: end procedure
```

Corpus Representation This stage of the pipeline is described in Algorithm 3. The text segments obtained from the segmentation process are subsequently utilized to construct the corpus

representation. The corpus representation refers to the definition of a vector space in which the textual segments found in the corpus are embedded. This crucial step involves transforming each segment into a numerical representation that effectively captures the underlying meaning and context of the segments, enabling consequential comparisons.

As the pipeline iterates through the segments, the cosine similarity between each segment's representation and the embeddings already included in the vector space is calculated, and the embedding with the highest similarity is found. Suppose the cosine similarity between both representations is higher than a predefined threshold, defined in the algorithm as the hyperparameter *min_similarity*. In that case, the segment being analyzed is interpreted as an instantiation of the template that originated the textual component represented in the vector space. Consequently, the embedding is updated to depict the new segment.

Two methods were explored for updating the embeddings in the vector space. One involves calculating the mean between the new segment and the existing embedding, taking into account the number of segments already represented by the embedding, that is, segments that, like the current one, have updated the embedding after being registered as similar to it. It is worth noting that the existing embeddings are seen as a generalization of a number of segments and, therefore, a representation of said segments. The second method calculates the weighted mean of these same segments. The weight employed in the second approach was selected as the similarity score between the segment and the existing embedding since it values segments closer to the existing representation and devalues those further apart.

If the segment is not similar to any in vector space, it is added to the corpus representation as a new embedding. The corpus embeddings are progressively built by incorporating the embeddings of new segments into the existing corpus representation. This iterative process ensures that the corpus embeddings collectively represent the entire range of text segments, effectively capturing the semantic essence of the corpus as a whole.

To optimize performance, for every predefined number of iterations — identified in the algorithm as the hyperparameter *reset_corpus* —, the corpus undergoes a cleaning process in which embeddings that do not appear in the corpus frequently enough are eliminated from the vector space. The frequency of each insertion is compared to the $min\_frequency$ percentage of the total number of documents processed. Given the substantial data size, this periodic cleaning helps alleviate the computational burden of the search and calculation of similarity between every new segment and the embeddings in the vector space, thus reducing processing time.

The definition of the hyperparameters *reset_corpus* and $min\_frequency$ directly impact the time the algorithm takes to build the corpus representation. To understand their impact, a sample of 10000 data entries was preprocessed, and the resulting text was segmented through the employment of NLTK Punkt sentence tokenizer. Afterward, a corpus representation was built upon this collection of data and its segments. This process was repeated and timed multiple times, for $min\_similarity = 0.85$, with varying values to the discussed hyperparameters. The resulting observations are presented by averaging the measured time for every 100 iterations, illustrated in Appendix A.

Algorithm 3 Corpus Representation

Require: $min\_similarity$, $reset\_corpus$, $min\_frequency$
procedure CORPUS_REPRESENTATION($corpus$, $model$)
  $corpus\_rep = []$
  for all $document$ in $corpus$ **do**
    $segments = text\_segmentation(document$, segmentation_technique$)$
    $seg\_embeddings = []$
    $seg\_embeddings =$ apply $model$ to $segments$
    for all $embedding$ in $seg\_embeddings$ **do**
      $\langle avg\_embed, n\_segments \rangle =$ closest vector to $embedding$ in $corpus\_rep$
      $similarity = COS(avg\_embed, embedding)$
      if $similarity < min\_similarity$ **then**
        $corpus\_rep$ appends $\langle embedding, 1 \rangle$
      else
        new_embed = update avg_embed with $embedding$
        $corpus\_rep(\langle avg\_embed, n\_segments \rangle) = \langle new\_embed, n\_segments + 1 \rangle$
      end if
    end for
    if number of documents $\%reset\_corpus == 0$ **then**
      for $\langle avg\_embed, n\_segments \rangle$ in $corpus\_rep$ **do**
        if $n\_segments < min\_frequency \times$ number of documents **then**
          $corpus\_rep$ remove $\langle avg\_embed, n\_segments \rangle$
        end if
      end for
    end if
  end for
  return $corpus\_rep$
end procedure

The figures displayed in the section show that the lack of cleaning of the vector space is quickly reflected in the steep growth of the average time of each iteration in a cycle of 100 iterations. In both demonstrations, the scenario where the corpus representation is never filtered — $reset\_corpus = None$ and $min\_frequency = 0$ — the average processing time for each document increases throughout the experiment, while in all other scenarios, the time per iteration tends to remain stable to the end of the experiment.

Template Identification Once the corpus representation is known, the pipeline advances to identify the embeddings that frequently occur across the corpus. These embeddings, named template embeddings, capture the essence of boilerplate texts. That is, they identify segments that persist throughout the corpus and maintain a distinct, recognizable structure. The pipeline, once again, identifies the embeddings consistently appearing in many documents by analyzing the occurrence frequency. In practice, segments in more than a pre-defined $template\_frequency$ of the total documents analyzed to create the corpus representation are signaled as template embeddings.

These embeddings serve as a representation of the recurrent textual structures, enabling the recognition of boilerplate text. This stage is illustrated by Algorithm 4.

Algorithm 4 Template Identification

```
1: procedure TEMPLATE_ID(corpus, template_frequency)
2:     template_embeddings = []
3:     corpus_rep = corpus_representation(corpus, model)
4:     for ⟨avg_embed, n_segments⟩ in corpus_rep do
5:         if n_segments ≥ template_frequency × len(corpus) then
6:             template_embeddings append avg_embed
7:         end if
8:     end for
9:     return template_embeddings
10: end procedure
```

Segment Labelling This stage of the pipeline is described in Algorithm 5. After identifying template representations, the segments undergo another iteration. Each segment is once again projected onto the vector space containing the identified template embeddings and compared to each to determine their similarity. If a segment exhibits sufficient similarity to any of the templates, it is labeled as a template segment. The label indicates that the segment aligns with the recurring patterns and structures captured by the template embeddings, suggesting it belongs to the boilerplate content. On the other hand, if the segment does not meet the similarity threshold with any of the templates, it is marked as a non-template segment. This labeling process effectively categorizes the segments into template or user-generated content. This last stage can also be expanded to be implemented in new documents.

It is worth noting that this solution allows for the identification of boilerplate segments no matter the position they are in the document. That is, the positioning of a segment is not taken into consideration in the algorithm described in this section, as segments are only classified based on their similarity to known templates.

4.3.1 Segmentation Techniques

Text segmentation plays a crucial role in the pipeline described as it identifies and separates distinct units of text, enabling more granular processing.

As explored in previous research, classic baseline sentence segmentation techniques may not be sufficient when dealing with domain-specific and user-generated content. These types of data often possess unique characteristics, such as informal language, abbreviations, slang, and domain-specific terminology. Additionally, the data usually does not follow strict syntactic rules, which is, in fact, a hindrance to the model's performance.

Various segmentation techniques have been employed and compared to tackle this challenge. These techniques encompass baseline methods, a procedure developed explicitly for multilingual

Algorithm 5 Segment Labelling

Require: *min_similarity*
  template_embeddings = *template_id*(*corpus*, template_frequency)
  procedure SEGMENT LABELLING(*document*)
    template = []
    content = []
    segments = *text_segmentation*(*document*, segmentation_technique)
    seg_embeddings = []
    seg_embeddings = apply *model* to *segments*
    for all *embedding* in *seg_embeddings* **do**
      avg_embed = closest vector to *embedding* in *template_embeddings*
      similarity = *COS*(*avg_embed*, *embedding*)
      if *similarity* ≥ *min_similarity* **then**
        template appends *embedding*
      else
        content appends *embedding*
      end if
    end for
    return *template*, *content*
  end procedure

legal-themed data, and a novel approach that deviates from Sentence Boundary Detection. The employment of these techniques is discussed in what follows.

4.3.1.1 NLTK Punkt

NLTK Punkt is a widely used method for sentence segmentation in Natural Language Processing. The sentence tokenizer available for this model is *PunktSentenceTokenizer*. This sentence tokenizer implements an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences; and then uses that model to find sentence boundaries. This approach has been shown to work well for many European languages, as is the case for Portuguese.

While the NLTK sentence tokenizer is able to correctly identify the boundaries of sentences in text that follows strict syntactic rules, its performance is not as reliable on data marked by the absence of punctuation and traditional structure. NLTK's performance is illustrated in Table 4.2, where it is shown to be able to break the Portuguese textual segment given in the first two examples into the appropriate sentences but struggles to identify the appropriate boundaries in the last excerpt of text.

The last entry of the table is particularly relevant to this work since it is a passage of one commonly found header in the data being processed. The technique is unable to segment this structure from the rest of the content properly. It is important to note that the example denotes different mistakes in the segmentation that may turn out as challenges in other stages of the pipeline solution: first, it isn't able to identify the boundaries of each information field, notably splitting <ADDRESS> into different segments, and secondly, it isn't able to recognize the departure from

Table 4.2: Examples of text segmentation using NLTK

Original text	**NLTK Punkt segmentation**
Olá! Como estás hoje? Espero que estejas bem.	Olá! Como estás hoje? Espero que estejas bem.
A viagem foi incrível! Paisagens deslumbrantes... Sentir o vento no rosto é maravilhoso. Regresso em breve.	A viagem foi incrível! Paisagens deslumbrantes... Sentir o vento no rosto é maravilhoso. Regresso em breve.
Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) ⟨ENTITY NAME⟩, ⟨ADDRESS⟩ Sua Referência Sua Comunicação Nossa Referência Data ⟨PROCESS ID⟩ ⟨DATE⟩ Assunto: Reclamação - ⟨PROCESS ID⟩ Prestador: ⟨PROVIDER⟩, ⟨ADDRESS⟩ Ex.mo(a) Senhor(a) Tomámos conhecimento da reclamação mencionada em assunto, que nos mereceu a melhor atenção.	Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) ⟨ENTITY NAME⟩, ⟨ADD RESS⟩ Sua Referência Sua Comunicação Nossa Referência Data ⟨PROCESS ID⟩ ⟨DATE⟩ Assunto: Reclamação - ⟨PROCESS ID⟩ Prestador: ⟨PROVIDER⟩ , ⟨ADDRESS⟩ Ex.mo(a) Senhor(a) Tomámos conhecimento da reclamação mencionada em assunto, que nos mereceu a melhor atenção.

the header to the rest of the text, representing part of it in the same sentence as content that does not belong in the structure.

4.3.1.2 multiLegalSBD

In [5], the authors develop mono and multilingual transformer models to specifically tackle the Sentence Boundary Detection (SBD) task in multilingual legal data. The paper found that the models achieved state-of-the-art performance in the task. Importantly for this solution, the paper also found that the proposed transformer model outperformed the techniques defined as the baseline, including NLTK, in sentence boundary detection tasks in Portuguese legal texts. The model hasn't been trained in Portuguese data, and this application is seen as a zero-shot experiment.

Legal data presents challenges for SBD systems due to its composition of smaller parts like paragraphs and clauses and its utilization of long sentences with complex structures such as citations, parentheses, and lists, which deviate from standard sentence structure, making it notably distinct from standard text. The data processed in this work resembles legal data in that, much like the latter, it does not adhere to stringent syntactic rules and therefore does not exhibit only conventional sentence structures.

As seen previously, traditional approaches to SBD demonstrate difficulties in identifying non-standard textual structures, multiLegalSBD[5], on the other hand, shows promising results in this regard. In Table 4.3, multiLegalSBD performance is illustrated through the examples that previously demonstrated NLTK's segmentation. In the first two examples, multiLegalSBD is shown to be able to identify the boundaries of the sentences correctly and, therefore, split the Portuguese

[5] `https://huggingface.co/models?search=rcds/distilbert-sbd`

text into the appropriate sentences. Unlike the first method, multiLegalSBD can detect the boundaries in text sourced from the ERS data. The text mentioned above contains a commonly used header and the initial part of a document's content. While this technique divides the header into its various fields, it can distinguish between the two components that compose the text.

Table 4.3: Examples of text segmentation using multiLegalSBD

Original text	**multiLegalSBD segmentation**
Olá! Como estás hoje? Espero que estejas bem.	Olá! Como estás hoje? Espero que estejas bem.
A viagem foi incrível! Paisagens deslumbrantes... Sentir o vento no rosto é maravilhoso. Regresso em breve.	A viagem foi incrível! Paisagens deslumbrantes... Sentir o vento no rosto é maravilhoso. Regresso em breve.
Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) ⟨ENTITY NAME⟩, ⟨ADDRESS⟩ Sua Referência Sua Comunicação Nossa Referência Data ⟨PROCESS ID⟩ ⟨DATE⟩ Assunto: Reclamação - ⟨PROCESS ID⟩ Prestador: ⟨PROVIDER⟩, ⟨ADDRESS⟩ Ex.mo(a) Senhor(a) Tomámos conhecimento da reclamação mencionada em assunto, que nos mereceu a melhor atenção.	Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) ⟨ENTITY NAME⟩, ⟨ADDRESS⟩ Sua Referência Sua Comunicação Nossa Referência Data ⟨PROCESS ID⟩ ⟨DATE⟩ Assunto: Reclamação - ⟨PROCESS ID⟩ Prestador: ⟨PROVIDER⟩, ⟨ADDRESS⟩ Ex.mo(a) Senhor(a) Tomámos conhecimento da reclamação mencionada em assunto, que nos mereceu a melhor atenção.

4.3.1.3 Sliding Window

A sliding window technique was developed to exploit other text segmentation units and circumvent the challenges created by the segmentation of text into phrases. In fact, although multiLegalSBD, in the illustrated example, can clearly distinguish between the header and the rest of the content, it splits the header structure into several segments, which may have undesired consequences in later stages of the pipeline and therefore, the need to explore other techniques to identify significant text segments arose.

A sliding window is a common approach used in algorithms for efficiently processing elements in a sequence. It involves creating a fixed-size window that moves across the sequence, processing a subset of elements within the window at each step.

In practice, the application of this technique to the segmentation step of the pipeline came to light in the creation of segments that represented n-grams of the tokens that took part in the text. The size of these n-grams was defined as ten tokens each, as the goal became to have units that were useful for identifying template segments that were relatively short while also long enough that commonplace wording was not identified as boilerplate.

The computational burden also had to be considered in the window size selection because this technique implies, for a window of size n and a document of m tokens, the existence of $m-n+1$ segments. In the implementation of this segmentation technique, the window of size n is also taken as an algorithm hyperparameter, even though the implementation discussed assigns

the value 10 to it. In the end, a document segmented through this process is represented by many more embeddings in the vector space than in the other two approaches. Therefore, the corpus vector space must be more often filtered through, and therefore the hyperparameter *reset_corpus* should be adjusted in accordance.

4.3.2 Sentence Embeddings

Sentence embeddings enable the representation of textual segments through numerical vectors, capturing the underlying meaning of each segment and the semantic relationships between different texts. By leveraging sentence embeddings to encode the semantic information of the corpus, the project's pipeline gains the ability to extract meaningful insights, as, for example, it becomes possible to measure the semantic similarity between segments and, therefore, identify identical or almost identical pairings of strings.

The described solution operates two different sentence embedding models, facilitating a comparative analysis that provides valuable insights. The chosen models have been trained on different data sources and languages.

ERS operates in the Portuguese national territory, so its data is mainly written in Portuguese. In light of the principles discussed above, it is important to implement models trained differently in Portuguese data to assess their ability to handle sentences in the language. Specifically, two models were reviewed: a multilingual model and a Portuguese-specific model.

4.3.2.1 Multilingual model

The multilingual model chosen for application is "paraphrase-multilingual-mpnet-base-v2"[6]. The model is pre-trained SBERT [32] variant specifically designed for the task of paraphrase identification. Importantly, it is trained in a large corpus containing a diverse range of languages, including Portuguese.

The model's architecture and training approach enables it to generate high-quality sentence representations, which can be used in the construction of a corpus representation. Its ability to encode and understand the meaning of sentences in different languages makes it a versatile option for multilingual NLP applications and specifically makes it suitable for the problem at hand.

4.3.2.2 Portuguese

The chosen model for the application is "stjiris/bert-large-portuguese-cased-legal-tsdae-gpl-nli-sts-MetaKD-v1"[7] [22]. This model is derived from a legal variant of BERTimBAU [36], a pre-trained BERT model for Brazilian Portuguese that achieves state-of-the-art performances ins NLP tasks specifically for this language variant. The model is fine-tuned on a large corpus of legal texts in Portuguese, making it well-suited for legal domain-specific applications. It is designed to

[6] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[7] https://huggingface.co/stjiris/bert-large-portuguese-cased-legal-tsdae-gpl-nli-sts-MetaKD-v1

capture the nuances and specificities of legal language, including legal terminology, syntax, and semantic structures commonly found in legal documents.

Although the current data is not characterized by the presence of text specifically from the legal domain, the decision to choose this language model was based upon the intuition that it might have valuable insights into the composition of template segments. Indeed, the already observed template follows well-defined structures similar to those found in the legal domain. Moreover, the model is also of interest as it is specifically trained on data from the European variant of the Portuguese language.

4.4 Summary

The chapter extensively explores ERS's complaint data and the challenges it poses to the development of automatic processing methods, with a particular focus on the noise embedded within it. A comprehensive analysis of the noise is performed to comprehend its sources and assess its impact on data analysis. Two particular sources of noise are identified: Optical Character Recognition (OCR) errors and boilerplate content.

To address the issues arising from the noise, a data preprocessing algorithm is implemented to rectify OCR errors and minimize their influence on subsequent tasks. Moreover, a pipeline is devised to detect and extract boilerplate content, facilitating the identification of recurrent structures in the data through text segmentation, corpus representation building, template embedding definition, and, finally, segment labeling. Several techniques are introduced for the representation of text segmentation and the consequent representation of textual segments.

The following chapter is set to validate the effectiveness of the proposed pipeline by applying it to ERS data, through the employment of the various techniques discussed in this chapter. The implementations will be evaluated on their performance in accurately identifying and classifying template segments, ultimately contributing to the resolution of user complaints and the enhancement of regulatory processes.

Chapter 5

Experimental Evaluation

In this chapter, the focus shifts toward evaluating and validating the proposed solution for addressing noise in the ERS data. This chapter aims to scrutinize the pipeline's ability to detect and extract boilerplate content. By subjecting the approach to rigorous testing on real-world ERS data, this chapter seeks to provide empirical evidence of its performance in accurately identifying and classifying template segments. The employment of clean text resulting from the implementation of the proposed solution will then be explored.

5.1 Solution Evaluation

The solution proposed in this work allows for different approaches in some stages of the pipeline. As such, there is a need to understand which variant of the developed solution is the most appropriate for deployment. In this section, the various implementations of the original solution will be attested through their application to a dataset built specifically for this task. The results of each implementation will be evaluated, and they will be compared.

5.1.1 Evaluation Dataset

As already explored, the data provided by the ERS is characterized by the multiplicity of boilerplate contents. Each data entry may employ different template content at different points in the document, and, as such, it is difficult to gauge global knowledge about all the original boilerplate applied in the data. There is notable difficulty in understanding the application of boilerplate accordingly through the partition of data by its format and origin. Labeling the content as either template or user-generated content manually is, as a consequence, an incredibly strenuous task due to the careful analysis and notation required for each entry. Each piece of text needs to be examined by a human to determine whether it falls within a perceived template or if it contains unique, user-generated information. This process can be time-consuming, especially when dealing with large volumes of data or complex document structures, as is the case.

In order to circumvent the inability to annotate the data manually, this work defines a new approach to the evaluation process. The approach entails the creation of a dedicated dataset designed

explicitly for evaluating the proposed solutions. This strategy aims to overcome the limitations of manual annotation by leveraging automated techniques and carefully curated data. Creating a specific dataset for evaluation purposes involves designing a set of documents that encompasses a wide range of scenarios and variations commonly encountered in ERS's data. The dataset must include different types of boilerplate content and various examples of user-generated content. It is essential to ensure that the dataset adequately represents the complexities and nuances present in real-world regulatory data. By employing a curated dataset, it is possible to evaluate the performance and effectiveness of the proposed solutions in an environment that closely mimics the challenges faced in actual data analysis.

To follow these guidelines, the dataset was built upon data exclusively comprised of user-generated content. This approach facilitated the controlled introduction of boilerplate text into otherwise clean data, allowing for the selection of appropriate boilerplate segments and the automatic labeling of the content. Additionally, each entry's ground truth was constructed based on this approach. It is important to note that the selected boilerplate segments aimed to represent those present in ERS's content faithfully, and, as a consequence, these structures were engraved with arbitrary information. That is, the templates' predefined information fields were filled with the appropriate entity details. Each of the chosen entities for this boilerplate padding was randomly selected from all the entries in ERS's data.

Importantly, in order to evaluate the approaches without the influence of external occurrences, the absence of OCR errors was also a prerequisite for this dataset. While the presence of such errors could be minimized through the preprocessing procedure previously presented, the noise they introduce in the data would still have a noticeable impact on the evaluation metrics of all the implemented solutions. This impact does not need to be accounted for, as the objective of this section is merely to compare the different boilerplate removal techniques.

The development of the evaluation dataset started with the selection of the complaints submitted through an online form, specifically the content categorized as 'RECLAMACAO,' as defined in Table 4.1, which refers to user-submitted complaint text. The complaint should not contain anything else but the text written by the complaining individual and therefore is void of template segments. The content submitted through the online form is represented and stored as text. Consequently, the gathered data does not have inherent noise associated with mistakes introduced by OCR and only contains user-generated content.

Upon the collection of data, three methods are defined to inject boilerplate content into the text. These methods are differentiated by the position in which the template structures are added and by the types of templates added to the content. These template structures were chosen as they represent some of the boilerplate text easily recognizable in the content.

The implemented techniques are as follows:

- **Before User-Generated Content**: This method defines four different template structures. One of these templates is randomly chosen and appended as content at the beginning of the document. The structures defined in this technique are illustrated in Figure 5.1.

Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) <ENTITY NAME>, <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data O.REC_[DDDDD/DDDD] RS_[DDDDD/DDDD] <DATE> Assunto: Reclamação - Exp_[DDDDD_DDDD] Prestador: <ENTITY NAME>

Entidade Reguladora da Saúde <ENTITY NAME>, <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data O.REC_[DDDDD/DDDD] RS_[DDDDD/DDDD] <DATE> Assunto: Reclamação - Exp_[DDDDD_DDDD] Reclamante: <ENTITY NAME>

From <ENTITY NAME> To <EMAIL> Cc <EMAIL> Recipients <EMAIL>

De: <EMAIL> [mail to: <EMAIL>] Enviada: <WEEK DAY> , [DD de MONTH de YEAR HH:MM] Para: <EMAIL> Assunto:

Figure 5.1: Template structures defined to be added before the user-generated content

- **After User-Generated Content**: Defines the addition of a template segment to the end of the document. It defines three template structures, illustrated in Figure 5.2.

Com os melhores cumprimentos, Departamento de Apoio ao Utente [Mod.DDD_DD] Pág. [D de D] [Mod.DDD_DD]

Com os melhores cumprimentos. <ENTITY NAME> <ADDRESS> Tcl. <PHONE NR> - Fax. <FAX> - E-mail: <EMAIL> mailto: <EMAIL>

www.ordemdospsicologos.pt CONHEÇA A CAMPANHA www.encontreumasaida.pt | PROCURA UMA SAÍDA EM encontreumasaida.pt Esta mensagem é confidencial e é propriedade da Ordem dos Psicólogos Portugueses. Destina-se unicamente ao conhecimento do destinatário nela identificado. Caso não seja o destinatário, não está autorizado a ler, reter, imprimir, copiar, divulgar, distribuir ou utiliza-la, parcial ou totalmente. Caso tenha recebido esta mensagem indevidamente, queira informar de imediato o remetente e proceder à destruição de todas as cópias.

Figure 5.2: Template structures defined to be added after the user-generated content

- **Between User-Generated Content**: This method defines four different template structures and appends the chosen template segment as content to a point in the middle of the content. The template can only be applied to documents where the original content contains line breaks, as these are straightforward patterns of signaling sentence ends, in opposition to a more complex method of segmenting the text into sentences. The structures defined in this technique are illustrated in Figure 5.3

As it is possible to visualize, some of the template structures defined allow for the addition of varied information in a number of fields. The areas that accept customization are either marked by the characters ⟨ and ⟩ or `[` and `]`, which describe the type of accepted information. Specifically, the first pair of characters is used to describe the type of information needed for that field, while `[` and `]` mark segments within which the text must follow a defined format. For example, if a tag marks `[Mod.DDD_DD]`, the value introduced here follows the tag by randomly selecting digits to stand in for the `D` character. On the other hand, ⟨ENTITY NAME⟩ calls for the introduction of the name of an entity specifically. The information required to build the template is taken from the available

Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) <ENTITY NAME>, <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data [O.REC_DDDDD/DDDD] [RS_DDDDD/DDDD] <DATE> Assunto: Reclamação - [Exp_DDDDD_DDDD] Prestador: <ENTITY NAME>

Entidade Reguladora da Saúde <ENTITY NAME>, <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data [O.REC_DDDDD/DDDD] [RS_DDDDD/DDDD] <DATE> Assunto: Reclamação - [Exp_DDDDD_DDDD] Reclamante: <ENTITY NAME>

This message has been scanned for viruses and dangerous content by MailScanner and is believed to be clean. This message has been scanned for viruses and dangerous content by MailScanner and is believed to be clean.

Esta mensagem é confidencial e é propriedade da Ordem dos Psicólogos Portugueses. Destina-se unicamente ao conhecimento do destinatário nela identificado. Caso não seja o destinatário, não está autorizado a ler, reter, imprimir, copiar, divulgar, distribuir ou utiliza-la, parcial ou totalmente. Caso tenha recebido esta mensagem indevidamente, queira informar de imediato o remetente e proceder à destruição de todas as cópias.

Figure 5.3: Template structures defined to be added between the user-generated content

data on the ERS's entities, which supplements names, addresses, and contact information, such as fax, phone numbers, and emails.

The following strategy guided the employment of the different methods of template creation and addition: one-third of all documents were added template through the before user-generated content method; another third of all documents followed the after user-generated content method; the remaining third of all documents were added template through both of these methods, that is, the user-generated content was padded with template in the beginning and end. Of the documents that presented the suitable characteristics, that is, documents that had at least two paragraphs, only half was added template through the between user-template content method. This strategy was employed in order to introduce varied types of template segments in the original data while specifically positioning them in different areas of the document.

The ground truth for each entry is built through the start and end indexes of each segment of the template once it is introduced in the original content. The ground truth labels the tokens in the template segments as the positive class and the tokens that belong to the user-generated content as the negative class. The characteristics of the resulting dataset are presented in Table 5.1.

Table 5.1: Evaluation dataset description

Full dataset	26 136 entries
1 template segment	10 486 entries
2 template segments	12 145 entries
3 template segments	3 505 entries
Before user-generated content	17 443 entries
Between user-generated content	10 169 entries
After user-generated content	17 679 entries
Positive tokens in ground truth	1 174 748 tokens
Negative tokens in ground truth	466 8247 tokens

5.1.2 Evaluation Metrics

The evaluation of the different implementations of the developed solution is based on the metrics of Accuracy, Precision, Recall, and F1-score. These metrics make use of the following four fundamental components to assess the solution's performance:

- **True Positives (TP)**: True Positives are the cases where the solution correctly predicts the positive class. In the context of this work, it refers to the number of boilerplate tokens predicted as such.
- **True Negatives (TN)**: True Negatives are the cases where the solution correctly predicts the negative class. In the context of this work, it refers to the number of user-generated content tokens predicted as such.
- **False Positives (FP)**: False Positives are the cases where the solution incorrectly predicts the positive class. In the context of this work, it refers to the number of boilerplate tokens that are wrongly predicted as user-generated content.
- **False Negatives (FN)**: False Negatives are the cases where the solution incorrectly predicts the negative class. n the context of this work, it refers to the number of user-generated content tokens that are wrongly predicted as boilerplate.

The metrics mentioned previously are described shortly.

Accuracy This is a simple and commonly used metric for evaluating the performance of a model. It measures the proportion of correct predictions from the total number of predictions made for a given dataset. Its formula is:

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

Precision Measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive. It answers the question: Of all the instances the model predicted as positive, how many were actually positive? The formula to calculate precision is:

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

Recall Also known as sensitivity or true positive rate, it is an evaluation metric used to measure the proportion of correctly predicted positive instances (true positives) out of all actual positive instances. Recall answers the question: Of all the actual positive instances, how many did the model correctly identify?

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

F1-score It is a commonly used evaluation metric in classification problems that combines precision and recall into a single value. It provides a balanced measure of the model's performance by considering both the ability to identify positive instances correctly and minimize false positives and negatives. Its formula is:

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

5.1.3 Evaluation Results

In this project, the solution pipeline allowed for the selection of different techniques in some of its stages, such as the segmentation of data and the construction of the corpus representation. In order to thoroughly test the chosen methodologies for these pipeline steps, approaches were defined for all possible combinations of techniques for each stage. These are, specifically, the segmentation techniques NLTK Punkt, multilegalSBD and sliding window, the sentence embeddings "paraphrase-multilingual-mpnet-base-v2" and "stjiris/bert-large-portuguese-cased-legal-tsdae-gpl-nli- sts-MetaKD-v1" and the corpus update techniques of mean and weighted mean.

In the end, twelve approaches were defined and subsequently evaluated. These approaches follow a nomenclature structure so that they are easily identifiable. First, they are recognized by the segmentation technique implemented — the possible names are *nltk*, *sbd* and *10* — followed by the sentence embedding model deployed — either *multi* or *pt*, for the multilingual and Portuguese language models respectively — and, finally, the corpus update strategy — either *mean* or *mean-w*. The definition of these approaches is further detailed in Table 5.2.

Table 5.2: Implemented approaches definition

	Text Segmentation	**Sentence Embedding**	**Embedding Update**
nltk_multi_mean	NLTK Punkt	multilingual	mean
nltk_multi_mean-w	NLTK Punkt	multilingual	weighted mean
nltk_pt_mean	NLTK Punkt	portuguese	mean
nltk_pt_mean-w	NLTK Punkt	portuguese	weighted mean
sbd_multi_mean	MultiLegalSBD	multilingual	mean
sbd_multi_mean-w	MultiLegalSBD	multilingual	weighted mean
sbd_pt_mean	MultiLegalSBD	portuguese	mean
sbd_pt_mean-w	MultiLegalSBD	portuguese	weighted mean
10_multi_mean	sliding window	multilingual	mean
10_multi_mean-w	sliding window	multilingual	weighted mean
10_pt_mean	sliding window	portuguese	mean
10_pt_mean-w	sliding window	portuguese	weighted mean

Additionally, it is essential to discuss the hyperparameters defined for each approach. These variables were defined through an iteration process that allowed for the understanding of the best environment for each approach. They were defined as such: for approaches based on NLTK or MultiLegalSBD, $min\_similarity = 0.85$, $reset\_corpus = 1000$, $min\_frequency = 0.001$ and $template\_frequency = 0.005$; approaches based on the sliding window technique were run with

$min\_similarity = 0.90$, $reset\_corpus = 100$, $min\_frequency = 0.001$ and $template\_frequency = 0.005$.

It is essential to understand that since the evaluation dataset contains content that was not processed by OCR techniques, it does not have errors resulting from these processes, and, as such, this dataset is not preprocessed as previously defined in Section 4.2.

The results are divided into Tables 5.3, 5.4 and 5.5. For ease of analysis, these tables organize the results through the segmentation technique the approaches employ: NLTK Punkt, multiLegalSBD and the sliding window technique, respectively. The analysis of these results will be structured to allow a deeper understanding of each of the stages of the pipeline and the techniques employed in each of them.

Table 5.3: Evaluation metrics for all the implementations based on NLTK Punkt

	nltk_multi_mean	***nltk_multi_mean-w***	***nltk_pt_mean***	***nltk_pt_mean-w***
Accuracy	**0.902**	0.894	0.890	0.886
Precision	**0.717**	0.717	0.627	0.610
Recall	**0.763**	0.712	0.446	0.430
F1-score	**0.704**	0.667	0.484	0.466

Firstly, in Table 5.3, the approach *nltk_multi_mean* performs better than all other approaches in all evaluation metrics. While this is true, all approaches have high accuracy values, as the lowest performance rounds the 88% value, compared to the 90% achieved by the *nltk_multi_mean* approach. Importantly, the approach *nltk_multi_mean-w* has comparable performance to *nltk_multi_mean*. The same can be said for *nltk_pt_mean* and *nltk_pt_mean-w*, two approaches that score similarly in all metrics.

In both pairs of approaches, the difference between them is in the technique employed in the stage that updates the corpus representation. While *nltk_multi_mean* and *nltk_pt_mean* simply update the existing embedding by calculating the mean between it and the new segment vector, *nltk_multi_mean-w* and *nltk_pt_mean-w*, employ a weighted mean in this stage. The performance of these models seems to indicate that the deployment of mean in this stage creates a better generalization of the repeated structures in the data, and that, perhaps the weights applied in the calculation of weigthed mean were not the most appropriate.

Importantly, there is a noticeable performance drop-off between the approaches that employ the multilingual language model and the language model developed for Portuguese legal text. This performance decline is clearly noticed when comparing *nltk_multi_mean* and *nltk_multi_mean-w* precision, recall and f1-score values to those of *nltk_pt_mean* and *nltk_pt_mean-w*. The precision metric achieved by the approaches that employ the Portuguese variant reveal that from all the text predicted as template only 63% and 61%, respectively, was actually correctly labelled while the recall metric reveals that only 45% and 43% of the existing template text was labelled as such.

These dynamics can also generally be verified in Table 5.4. Out of the implementations based on the segmentation of text through the multiLegalSBD model *sbd_multi_mean* performs the best in all evaluation metrics. Additionally, the approaches that implement the multilingual language

Table 5.4: Evaluation metrics for all the implementations based on multiLegalSBD

	sbd_multi_mean	***sbd_multi_mean-w***	***sbd_pt_mean***	***sbd_pt_mean-w***
Accuracy	**0.922**	0.918	0.869	0.868
Precision	**0.858**	0.845	0.686	0.682
Recall	**0.559**	0.542	0.291	0.287
F1-score	**0.643**	0.627	0.374	0.369

model — *sdb_multi_mean* and *sbd_multi_mean-w* — convey very similar performances, as do those that employ language model developed for Portuguese — *sbd_pt_mean* and *sbd_pt_mean-w*. The first pair achieves much higher evaluation metrics, specifically for precision, recall, and f1-score. Within these pairs, the approaches that rely on mean to update the corpus representation perform slightly better in all metrics.

Table 5.5: Evaluation metrics for all the implementations based on the sliding window of size 10

	10_multi_mean	***10_multi_mean-w***	***10_pt_mean***	***10_pt_mean-w***
Accuracy	0.914	**0.916**	0.856	0.854
Precision	0.784	**0.814**	0.358	0.358
Recall	0.620	**0.625**	0.259	0.262
F1-score	0.672	**0.683**	0.294	0.296

Finally, in Table 5.5, the division between the approaches that employed the multilingual and Portuguese language models is clearly pronounced. Once again, the approaches can be grouped by their performance into two pairs within which the methods achieve similar evaluation metrics. The sliding window-based approaches, however, present the biggest performance drop-off between the pairings, as the techniques that rely on the multilingual language model for text representation attain 67% and 68% in F1-score, while the approaches based on the Portuguese language model only reach values lower than 30% in this same metric. Additionally, the best performance out of all the approaches in this table is achieved by the *10_multi_mean-w*, which implements a mean-weighted update to embeddings in the corpus representation, and slightly outperforms *10_multi_mean*. It is also worth mentioning that the methods that implement the sliding window segmentation technique were not employed in the evaluation dataset as a whole. These methods are very computationally heavy, as the number of segments represented in each data entry is close to the number of tokens that define its text content. This characteristic is particular to this segmentation technique. As a consequence of time restraints, these methods were only evaluated in 10% of the dataset.

In the end, out of all the approaches evaluated, *nltk_multi_mean* achieved the best performance. This approach utilizes the segmentation technique mentioned earlier as the baseline, which was anticipated to yield poorer performance. One possible explanation for the limited success of multiLegalSBD could be its inability to recognize complete template structures, as shown in the examples documented in Section 4.3.1, demonstrating a tendency to divide them into smaller units. The metrics obtained from the multiLegalSBD-based approaches support these findings. In each case, the precision of these methods surpasses that of the comparable nltk-based approach, while

the recall value is considerably lower. This suggests that the initial approaches are better at correctly identifying user-generated content but are not as effective in recognizing boilerplate content as such. On the other hand, the sliding window-based approaches' performances may indicate that the definition of $n = 10$, may not have been the most appropriate for the task.

Notably, the choice of text representation technique had the most impact on the approaches' performances as the multilingual language model has consistently much better metrics than the language model trained on Portuguese legal data. This discrepancy is unexpected, but it may be a result of the specificity of the data, as, while previously defined as having structures similar to those present in legal data, the textual content that characterizes the ERS data does not occur in many legal-specific terminologies.

Once again, in all the implemented approaches, there is no note-worthy difference between the methods employed to update the representations in the vector space. In this aspect, mean tends to achieve better performances, but this is also not always the truth.

5.1.4 Qualitative Evaluation

In order to further understand the approach with the best results, some of the entries of the dataset created for the evaluation of the proposed solution are used as a basis for visual comparison between the results derived from the pipeline application and the established ground truth.

Appendix B provides examples of data entries from the fictional dataset. Each example is presented in two iterations, offering a comprehensive view of the content. In the first iteration, the content is displayed with the injected boilerplate highlighted, making it easier to identify. The second iteration, on the other hand, labels the segments identified as boilerplate by utilizing the *nltk_multi_mean* approach. These examples allow for a deeper exploration of the performance and limitations of this approach.

As reflected in the evaluation metrics, the *nltk_multi_mean* approach indicates that it is generally effective in correctly identifying boilerplate textual segments. The examples chosen for evaluation consistently demonstrate that the segments injected as templates are accurately labeled as such by the approach. This observation is particularly evident in Figures B.1 and B.2, where the segments of text highlighted in the images that depict the ground truth and the boilerplate identified by the approach are fully matched.

On the other hand, it is also very commonly observed that segments of text preceding or following the injected template components are also identified as boilerplate. This phenomenon can be attributed to the limitations of NLTK Punkt as a method for sentence segmentation, analyzed in depth in Section 4.3.1.1, as the approach raises some difficulties in segmenting text when it does not exhibit normal punctuation usage. This behavior is exemplified at its extreme in Figure B.5. In the example, the user-generated content is restricted to one sentence, and there is no punctuation to limit the injected template structure from it. As a result, the whole document is understood to be a single text segment, and therefore, it is all labeled as template. Additionally, other segmentation issues can be noted in Figure B.8, as the example illustrates a scenario in which the template is separated, and the structure is not identified as boilerplate as a whole. In this case, the character

. after 'Fax' was interpreted as the end of a sentence, and, as such, the information field that followed and the rest of the structure were labeled as user-generated content.

In the end, aside from the mentioned issues, the approach *nltk_multi_mean* proved to be highly successful in labeling the boilerplate structures as intended.

5.2 Cleaned complaint data application

In this section, the approach with the best evaluation performance for each text segmentation technique — except for the sliding window technique — is employed in a set of data that is then classified. The classification of the data is done using a model built within the larger scope of the ongoing project with ERS and made available for this validation. The model identifies themes and subjects in each data entry which will then be compiled into a single data field for each complaint process. In other words, while the model is run for each content in the data, the predictions that arise from this procedure for both themes and subjects, are reunited with the predictions made for all other data entries that belong to the same ERS process.

While this approach does not validate the solution employed in this work it illustrates how the resulting data may be used in future tasks. Additionally, it also demonstrates how the removal of boilerplate content from the data may allow this data to be employed in the automatic classification of complaints.

The approaches chosen for this analysis were *nltk_multi_mean* and *sbd_multi_mean*, as these were the approaches that achieved better performances on the previous evaluation. As discussed earlier, no sliding window techniques were employed at this stage. It is worth noting that these approaches were applied with the hyperparameters defined in Section 5.1.3.

Since the ground truth is associated with a complaint process and not with each individual data entry, the processes with known ground truths were sampled, and the data contained in each process were collected. The proposed pipeline processed these data entries through the already-mentioned approaches. The result was, as expected, the labeling of sequences of text as either template or user-generated content. Subsequently, the text identified as boilerplate was extracted from the content, which came to be composed only of user-generated content.

The classification model was employed in each of the now-cleaned data entries, and the themes and subjects of each were identified. The results of this multi-labeling task were then merged into a set of predictions in an information field. In turn, once all content was classified, the data entries were again combined into a single process, and the themes and subjects identified for each textual content were combined into the predictions for said process. Additionally, the model was employed in the raw version of the data entries selected, and the results of these operations were also compiled into a single field for each complaint process.

It is worth noting that, in addition to the model and the ground truth, predictions of processes' themes and subjects previously made were also made available. These predictions result from the employment of the classification model in some of the original content contained within the processes, as not all data was deemed usable in this task due to the noise that permeated it. The

comparison between these predictions and those achieved by the approaches developed in this paper allows for a deeper understanding of the potential employment of this solution in cleaning textual content and the consequences and possibilities that arise from it.

As such, some observations about the previous predictions and those obtained through raw data entries and the application of the noise reduction approaches will be presented. Tables 5.6 and 5.7 detail, respectively, the precision and recall metrics for each of the classification tasks described. In these tables, 'Original predictions' refers to the predictions made available from a previous experience, 'Raw data entries' refers to the predictions made upon the unprocessed content, 'Preprocessed + *nltk_multi_mean*' refers to the predictions made upon the data processed by this approach and 'Preprocessed + *sbd_multi_mean*' also refers to the predictions made through the employment of *sbd_multi_mean*.

Table 5.6: Precision metrics for all process classification tasks

Original predictions	0.733
Raw data entries	0.496
Preprocessed + *nltk_multi_mean*	0.471
Preprocessed + *sbd_multi_mean*	0.464

Table 5.7: Recall metrics for all process classification tasks

Original predictions	0.568
Raw data entries	0.764
Preprocessed + *nltk_multi_mean*	0.764
Preprocessed + *sbd_multi_mean*	0.762

The metrics illustrated in the tables show a noticeable gap between the original predictions that were made available and all the other predictions. This gap is registered in the precision metric, with the first model obtaining a performance of 73% while the others do not record more than 50%, and in the recall metric, with the predictions built on all data inputs achieving at least 76% performance while the original records only 57%. The difference between these metrics suggests that the input of more data entries in the classification task allows for a wider set of predictions not seen previously, but it also allows for the introduction of new incorrect predictions. This last outcome may be a direct consequence of the classification model having been developed for classifying specific types of data and, therefore, registering poor labeling for new unseen documents.

Additionally, the three classification tasks made upon all data entries from each selected complaint process detail very similar evaluation metrics, with 'Raw data entries' signaling a slightly better performance than predictions based on data processed by one of the developed approaches in both metrics.

In the end, this observation does not allow for larger extrapolations, but it signals a notable difference between the predictions based on all data entries and the previous approach. While obviously, the discrepancy between these percentages has no concrete implications, it does at least indicate the potential that employing all ERS data introduces into the exploration of the automatic

claims processing problem. It isn't possible to derive further implications, specifically between the predictions based on raw data and the different cleaning approaches, as the model employed was not developed to be able to classify all data entries within complaint processes.

5.3 Summary

This chapter assesses the solution pipeline's efficacy in detecting and extracting boilerplate content from ERS data. A dedicated evaluation dataset comprising user-generated content with carefully injected boilerplate text, facilitated the evaluation of different implementation variants, which were then compared. *nltk_multi_mean* was identified as the best-performing approach explored. Finally, the potential of using the cleaned data in developing a classification model in automatic complaint processing was explored.

Chapter 6

Conclusion

This dissertation aimed to provide a noise reduction approach for textual data generated by text extraction methods applied to files. The background for the developed solution was founded on key topics such as Text Representation, Sentence Boundary Detection, and String Similarity. Additionally, the examination of related works focused on data-centric AI, highlighting the importance of data in the development of effective AI systems. Data preparation received special attention, as did its relation to the research's objective. Noise removal has arisen as a critical issue within data preparation, and boilerplate removal, in particular, was a focal point of this stage of research since it thoroughly explores this theme. While varied, most of its applications are conducted on web data, in which the boilerplate assumes a well-known structured layout, and therefore the solution described in this paper differs from those typically explored in this field.

This chapter concludes the work developed and finalizes with an overview of each research question enumerated at the beginning and how they were fulfilled. Additionally, potential future work is listed and discussed.

6.1 Research Questions

Research Question 1: Is it possible to develop a pipeline to automatically detect boilerplate content in a diverse document collection containing multiple templates? This thesis presents the development of a solution strategy that addresses the issues raised by data noise, specifically OCR mistakes and boilerplate material. A thorough examination of noise sources and their influence on data analysis was carried out. To address these concerns, a data preprocessing system was developed to correct OCR mistakes and minimize their impact on the following activities. Finally, a pipeline for boilerplate content recognition and extraction was built. In the end, the solution presented in this work does not depend on previously labeled content, and, as such, it is possible to affirm that, indeed the development of a pipeline to automatically detect boilerplate content within a collection of data that presented numerous templates was possible.

Research Question 2: Given the development of a solution, does the defined process allow for the employment of scenario-specific techniques? If so, which techniques within the solution result in a more adequate identification of boilerplate content for the present case study? The developed solution implemented a pipeline structured into four distinct stages: text segmentation, corpus representation building, template embedding definition, and textual segment labeling. These stages allow for the employment of different techniques. For the case study at hand, different strategies were explored for text segmentation, text representation, and embedding updates. The experimental evaluation rigorously assessed the efficacy of the solution pipeline in detecting and extracting boilerplate content from the data. A dedicated evaluation dataset was utilized, featuring carefully injected boilerplate text to test different implementation variants. The *nltk_multi_mean* approach, which employed the NLTK sentence tokenizer, a multilingual language model and mean as an update strategy, emerged as the best-performing method, demonstrating its capability to identify and classify template segments effectively.

Research Question 3: Does diminishing noise in data enable the effective employment of models in the data, specifically for classification? This research question was not fully undertaken. As discussed previously, there seem to be indications that classification upon raw data entries results in better predictions for each complaint process. However, this observation was based upon a model that was not developed to be able to classify all data entries, which may have impacted the displayed results. The present question can only be fully answered through the development of a classification model built on the assumption that this content should be excluded from the handling of complaints.

6.2 Future Work

It is important to acknowledge the limitations of this work. While promising results were achieved for the identification and removal of the templates that characterize the ERS data, further research and exploration are required to tackle more complex noise sources and optimize the method's efficiency. Specifically, the techniques employed in the various stages of the pipeline should be further explored. It is extremely important that text segmentation, specifically, is examined, as the approach with the most promising results is based on a baseline technique, and therefore, it is believed that the employment of more appropriate segmentation methods would result in better performances. Additionally, there should also be a future effort to find and implement models that allow a text representation more suitable to the domain in which this data is included.

Finally, as has already been mentioned, the implications of adding this clean textual data to the data input in the classification task should be explored carefully, and there should be a renewed focus on developing an automatic complaint-handling system through this data.

References

[1] Meet ai's multitool: Vector embeddings. Available at https://cloud.google.com/blog/topics/developers-practitioners/meet-ais-multitool-vector-embeddings, last accessed in July 2023.

[2] Sumeet Agarwal, Shantanu Godbole, Diwakar Punjani, and Shourya Roy. How much noise is too much: A study in automatic text classification. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 3–12, 2007.

[3] Md Manjurul Ahsan, MA Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta, and Zahed Siddique. Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, 9(3):52, 2021.

[4] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[5] Tobias Brugger, Matthias Stürmer, and Joel Niklaus. Multilegalsbd: A multilingual legal sentence boundary detection dataset. *arXiv preprint arXiv:2305.01211*, 2023.

[6] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. Block-based web search. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 456–463, 2004.

[7] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. Multieurlex–a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *arXiv preprint arXiv:2109.00904*, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] ERS Entidade Reguladora da Saúde. ERS — A ERS, 2020. Available at https://www.ers.pt/pt/institucional/a-ers/, last accessed in January 2023.

[10] Matthias Gallé and Jean-Michel Renders. Boilerplate detection and recoding. In Maarten de Rijke, Tom Kenter, Arjen P. de Vries, ChengXiang Zhai, Franciska de Jong, Kira Radinsky, and Katja Hofmann, editors, *Advances in Information Retrieval*, pages 462–467, Cham, 2014. Springer International Publishing.

[11] Ingo Glaser, Sebastian Moser, and Florian Matthes. Sentence boundary detection in german legal documents. In *ICAART (2)*, pages 812–821, 2021.

[12] Suhit Gupta, Gail Kaiser, David Neistadt, and Peter Grimm. Dom-based content extraction of html documents. In *Proceedings of the 12th international conference on World Wide Web*, pages 207–214, 2003.

[13] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

[14] Francesca Incitti, Federico Urli, and Lauro Snidaro. Beyond word embeddings: A survey. *Information Fusion*, 89:418–436, 2023.

[15] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. *Advances in neural information processing systems*, 28, 2015.

[16] Tibor Kiss and Jan Strunk. Unsupervised multilingual sentence boundary detection. *Computational linguistics*, 32(4):485–525, 2006.

[17] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450, 2010.

[18] Ankit Kumar, Piyush Makhija, and Anuj Gupta. Noisy text data: Achilles' heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 16–21, Online, November 2020. Association for Computational Linguistics.

[19] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

[20] Jurek Leonhardt, Avishek Anand, and Megha Khosla. Boilerplate removal using a neural sequence labeling model. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 226–229, New York, NY, USA, 2020. Association for Computing Machinery.

[21] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014.

[22] Rui Melo, Professor Pedro Alexandre Santos, and Professor João Dias. A Semantic Search System for Supremo Tribunal de Justiça.

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[24] Jeff Mitchell and Mirella Lapata. Vector-based models of semantic composition. In *proceedings of ACL-08: HLT*, pages 236–244, 2008.

[25] Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. *arXiv preprint arXiv:2110.00806*, 2021.

[26] Joel Niklaus, Veton Matoshi, Pooja Rani, Andrea Galassi, Matthias Stürmer, and Ilias Chalkidis. Lextreme: A multi-lingual and multi-task benchmark for the legal domain. *arXiv preprint arXiv:2301.13126*, 2023.

[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[28] ME Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee, and L Zettlemoyer. Deep contextualized word representations. arxiv 2018. *arXiv preprint arXiv:1802.05365*, 12, 2018.

[29] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*, 2020.

[30] Jonathon Read, Rebecca Dridan, Stephan Oepen, and Lars Jørgen Solberg. Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, 2012.

[31] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[32] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084, 2019.

[33] George Sanchez. Sentence boundary detection in legal text. In *Proceedings of the natural legal language processing workshop 2019*, pages 31–38, 2019.

[34] Jaromir Savelka, Vern R Walker, Matthias Grabmair, and Kevin D Ashley. Sentence boundary detection in adjudicatory decisions in the united states. *Traitement automatique des langues*, 58:21, 2017.

[35] Jaromir Savelka, Hannes Westermann, Karim Benyekhlef, Charlotte S Alexander, Jayla C Grant, David Restrepo Amariles, Rajaa El Hamdani, Sébastien Meeùs, Aurore Troussel, Michał Araszkiewicz, et al. Lex rosetta: transfer of predictive models across languages, jurisdictions, and legal domains. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 129–138, 2021.

[36] Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*, 2020.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Thijs Vogels, Octavian-Eugen Ganea, and Carsten Eickhoff. Web2text: Deep structured boilerplate removal. In *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40*, pages 167–179. Springer, 2018.

[39] Mingyang Wan, Daochen Zha, Ninghao Liu, and Na Zou. In-processing modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, 17(3):1–27, 2023.

[40] Shanchan Wu, Jerry Liu, and Jian Fan. Automatic web content extraction by combination of learning and grouping. In *Proceedings of the 24th international conference on World Wide Web*, pages 1264–1274, 2015.

[41] Xue Ying. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022. IOP Publishing, 2019.

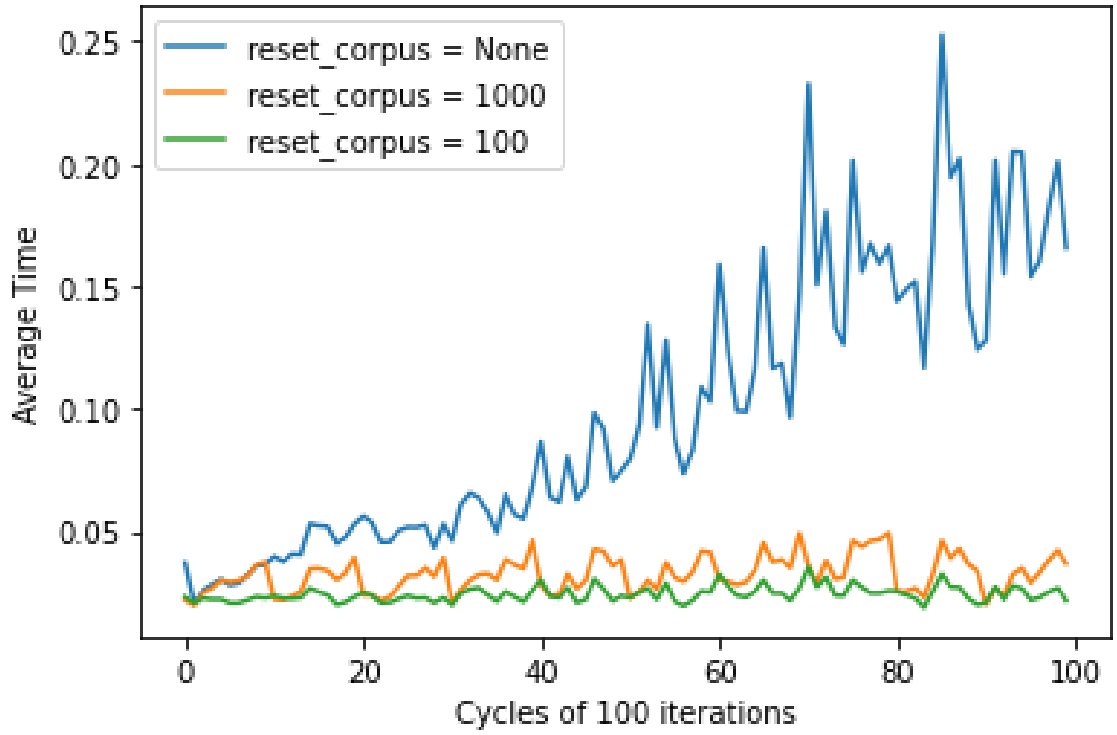
[42] Minghe Yu, Guoliang Li, Dong Deng, and Jianhua Feng. String similarity search and join: a survey. *Frontiers of Computer Science*, 10:399–417, 2016.

[43] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, and Xia Hu. Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 945–948. SIAM, 2023.

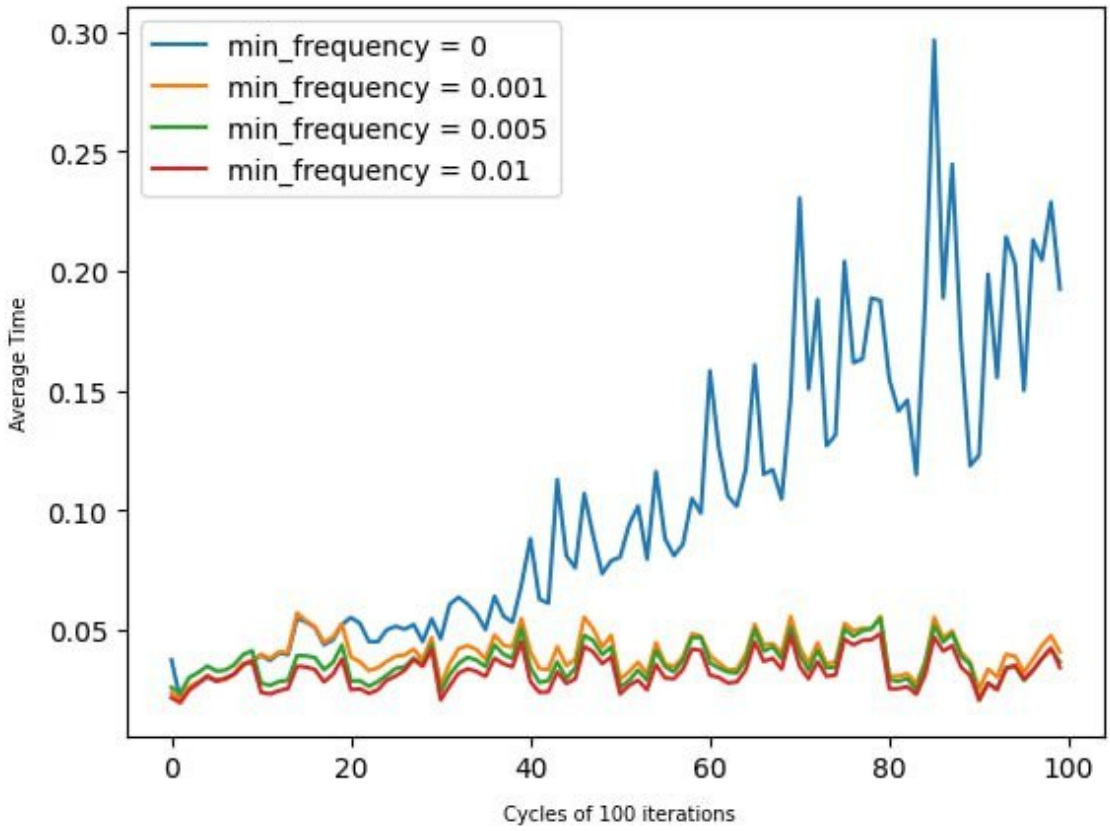
[44] Daochen Zha, Zaid Pervaiz Bhat, Kwei-Herng Lai, Fan Yang, Zhimeng Jiang, Shaochen Zhong, and Xia Hu. Data-centric artificial intelligence: A survey. *arXiv preprint arXiv:2303.10158*, 2023.

Appendix A

Solution Approach- Pipeline Description



(a) Average iteration time per 100 iteration cycle, for *reset_corpus* as *None*, 1000 and 100. *min_frequency* was set as 0.001 for every approach.



(b) Average iteration time per 100 iteration cycle, for *min_frequency* as 0, 0.001, 0.005 and 0.01. *reset_corpus* was set as 1000 for every approach.

Figure A.1: Impact of definition of hyperparameters *reset_corpus* and *min_frequency* in corpus representation

Appendix B

Experimental Evaluation - Qualitative Evaluation

-A [redacted] solicitei a marcaçao de uma consulta, via e-mail, de saude materna, dado estra gravida e ser uma necessidade e direito de qualquer gravida o acompanhamento pré-natal, sendo que já tinha em minha posse o resultados dos primeiros exames e análises que fiz relativamente ao 1º trimestre.

Até ao dia de hoje, [redacted] ainda não obtive qualquer resposta relativamente ao meu pedido de consulta, que á data já contabilizo 6(2 via email e 4 presenciais-onde escrevem sempre numa folha e não há qualquer feedback do estado desse pedidos, ficando qualquer um a questionar-se qual o destino dessas mesmas folhas.

No dia [redacted] dirigi-me, mais uma vez ao centro de saude em questao, de forma a obter resposta aos meus pedidos e mais uma vez ninguem tinha algo para me dizer senão "já estão a chamar, tem que aguardar", sendo que por esse motivo e dado que já estou com 22 semanas de garvidez e nenhum acompanhamento solicitei que me fornecessem, entao, as credenciais para exames e anaálise e ainda a declaração para a segurança social e ainda o cheque dentista, para não ter de voltar e fazer o pedido desses doc, e para poder fazer os exames e mostrar a um profissional do privado, evitando assim custos maiores, e ficando apenas com o encargo da consulta e nao dos exames, pois são um direito meu e quero ter acesso a eles.

Para meu espanto e perplexidade foi me fornecido apenas o cheque dentista pela admnistrativa, como se o mais preocupante neste momento fossem os meus dentes e não uma vida que se está a gerar.

Foi me dada a indicação que no dia de hoje [redacted], estaria no centro uma médica ao qual iam expor a minha situaçao, mas que não garantiam nada, pois não é uma médica de saúde materna... Estou a aguradar até ao momento e não obtenho resposta e, dado que estou a trabalhar, e não podendo estar constantemente neste "vai e vém", eu exijo que sejam tomada medida relativamente a este tipo de procedimento que está a ser tomado e respostas ao meu pedido, pois se a determinado momento,eu descobrir que ou a criança ou eu tenhamos algum problema que pudesse ser detectado ou impedido com o devido acompanhamento irei imputar responsabilidades a quem de direito.

Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod.[redacted]Pág. 3 de 5
Mod.[redacted]

(a) Boilerplate injected into content

-A [redacted] solicitei a marcaçao de uma consulta, via e-mail, de saude materna, dado estra gravida e ser uma necessidade e direito de qualquer gravida o acompanhamento pré-natal, sendo que já tinha em minha posse o resultados dos primeiros exames e análises que fiz relativamente ao 1º trimestre.

Até ao dia de hoje, [redacted] ainda não obtive qualquer resposta relativamente ao meu pedido de consulta, que á data já contabilizo 6(2 via email e 4 presenciais-onde escrevem sempre numa folha e não há qualquer feedback do estado desse pedidos, ficando qualquer um a questionar-se qual o destino dessas mesmas folhas.

No dia [redacted] dirigi-me, mais uma vez ao centro de saude em questao, de forma a obter resposta aos meus pedidos e mais uma vez ninguem tinha algo para me dizer senão "já estão a chamar, tem que aguardar", sendo que por esse motivo e dado que já estou com 22 semanas de garvidez e nenhum acompanhamento solicitei que me fornecessem, entao, as credenciais para exames e anaálise e ainda a declaração para a segurança social e ainda o cheque dentista, para não ter de voltar e fazer o pedido desses doc, e para poder fazer os exames e mostrar a um profissional do privado, evitando assim custos maiores, e ficando apenas com o encargo da consulta e nao dos exames, pois são um direito meu e quero ter acesso a eles.

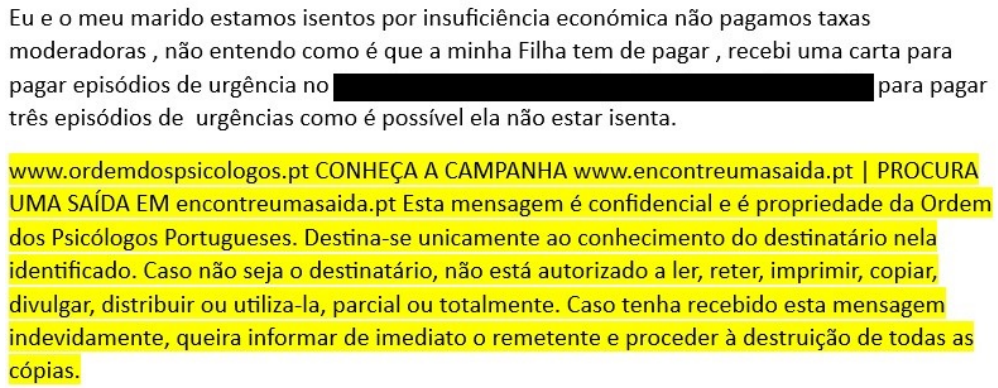
Para meu espanto e perplexidade foi me fornecido apenas o cheque dentista pela admnistrativa, como se o mais preocupante neste momento fossem os meus dentes e não uma vida que se está a gerar.

Foi me dada a indicação que no dia de hoje [redacted], estaria no centro uma médica ao qual iam expor a minha situaçao, mas que não garantiam nada, pois não é uma médica de saúde materna... Estou a aguradar até ao momento e não obtenho resposta e, dado que estou a trabalhar, e não podendo estar constantemente neste "vai e vém", eu exijo que sejam tomada medida relativamente a este tipo de procedimento que está a ser tomado e respostas ao meu pedido, pois se a determinado momento,eu descobrir que ou a criança ou eu tenhamos algum problema que pudesse ser detectado ou impedido com o devido acompanhamento irei imputar responsabilidades a quem de direito.

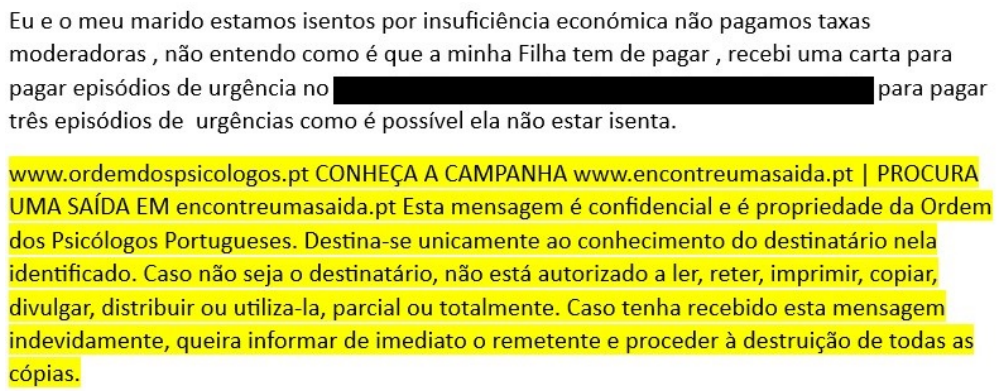
Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod.[redacted]Pág. 3 de 5
Mod.[redacted]

(b) Boilerplate identified by *nltk_multi_mean*

Figure B.1: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example



(a) Boilerplate injected into content



(b) Boilerplate identified by *nltk_multi_mean*

Figure B.2: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example

Entidade Reguladora da Saúde <ENTITY NAME>, [redacted] <ADDRESS> [redacted] Sua Referência
Sua Comunicação Nossa Referência Data O.REC_[redacted] RS_[redacted] <DATE>
Assunto: Reclamação - Exp_[redacted] Reclamante: <ENTITY NAME>

Na Ida ao [redacted], na tentativa de apanhar uma consulta do dia, (urgência)

que nos tinha sido solicitado já por email, que teriamos de lá estar às 8h, (56).

Chegamos lá por volta das 7h30, estando já lá algumas pessoas à porta. Quando distribuiram as senhas a todas as pessoas, informaram que as quatro pessoas que entraram já tinha esgotado as 5 urgências, logo não conseguimos .e essas mesmos estavam agendadas para o final do dia

Ao tentar arranjar uma solução, conseguiram

uma consulta para as 9h, onde a mesma foi dada as 11h30 e porque nós reclamamos que estava a demorar a atender e ninguém chamava para o consultório, e além disso soubemos pelo próprio médico que não tinha nenhum paciente agendado desde as 9h e estivémos aquelas horas todas à espera, cono se estivessem a gozara com a nossa cara.

Além desta situação desagradável e a

Esta mensagem é confidencial e é propriedade da Ordem dos Psicólogos Portugueses. Destina-se unicamente ao conhecimento do destinatário nela identificado. Caso não seja o destinatário, não está autorizado a ler, reter, imprimir, copiar, divulgar, distribuir ou utiliza-la, parcial ou totalmente. Caso tenha recebido esta mensagem indevidamente, queira informar de imediato o remetente e proceder à destruição de todas as cópias.

Aindau nós E Es outros pacientes estiveram às 7h da manhã à espera ao relento ao frio e à chuva.

sem condições nenhumas

E é uma vergonha este cs não ter condições para acomodar o docentes, falta de organização, e terem medicos a mais de 2 anos de baixa e não os substituiur.

(a) Boilerplate injected into content

Entidade Reguladora da Saúde <ENTITY NAME>, [redacted] <ADDRESS> [redacted] Sua Referência
Sua Comunicação Nossa Referência Data O.REC_[redacted] RS_[redacted] <DATE>
Assunto: Reclamação - Exp_[redacted] Reclamante: <ENTITY NAME>

Na Ida ao [redacted], na tentativa de apanhar uma consulta do dia, (urgência)

que nos tinha sido solicitado já por email, que teriamos de lá estar às 8h, (56).

Chegamos lá por volta das 7h30, estando já lá algumas pessoas à porta. Quando distribuiram as senhas a todas as pessoas, informaram que as quatro pessoas que entraram já tinha esgotado as 5 urgências, logo não conseguimos .e essas mesmos estavam agendadas para o final do dia

Ao tentar arranjar uma solução, conseguiram

uma consulta para as 9h, onde a mesma foi dada as 11h30 e porque nós reclamamos que estava a demorar a atender e ninguém chamava para o consultório, e além disso soubemos pelo próprio médico que não tinha nenhum paciente agendado desde as 9h e estivémos aquelas horas todas à espera, cono se estivessem a gozara com a nossa cara.

Além desta situação desagradável e a

Esta mensagem é confidencial e é propriedade da Ordem dos Psicólogos Portugueses. Destina-se unicamente ao conhecimento do destinatário nela identificado. Caso não seja o destinatário, não está autorizado a ler, reter, imprimir, copiar, divulgar, distribuir ou utiliza-la, parcial ou totalmente. Caso tenha recebido esta mensagem indevidamente, queira informar de imediato o remetente e proceder à destruição de todas as cópias.

Aindau nós E Es outros pacientes estiveram às 7h da manhã à espera ao relento ao frio e à chuva.

sem condições nenhumas

E é uma vergonha este cs não ter condições para acomodar o docentes, falta de organização, e terem medicos a mais de 2 anos de baixa e não os substituiur.

(b) Boilerplate identified by *nltk_multi_mean*

Figure B.3: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example

Entidade Reguladora da Saúde <ENTITY NAME>, <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data O.REC_ RS_ <DATE> Assunto: Reclamação - Exp_ Reclamante: <ENTITY NAME>

O meu pai, , SNS nº, em efetuou cirurgia a catarata, . De há um ano para cá tem-se queixado com falta de visão, pelo que se dirigiu à médica de familia que fez pedido de consulta em oftalmologia a - 7 meses.

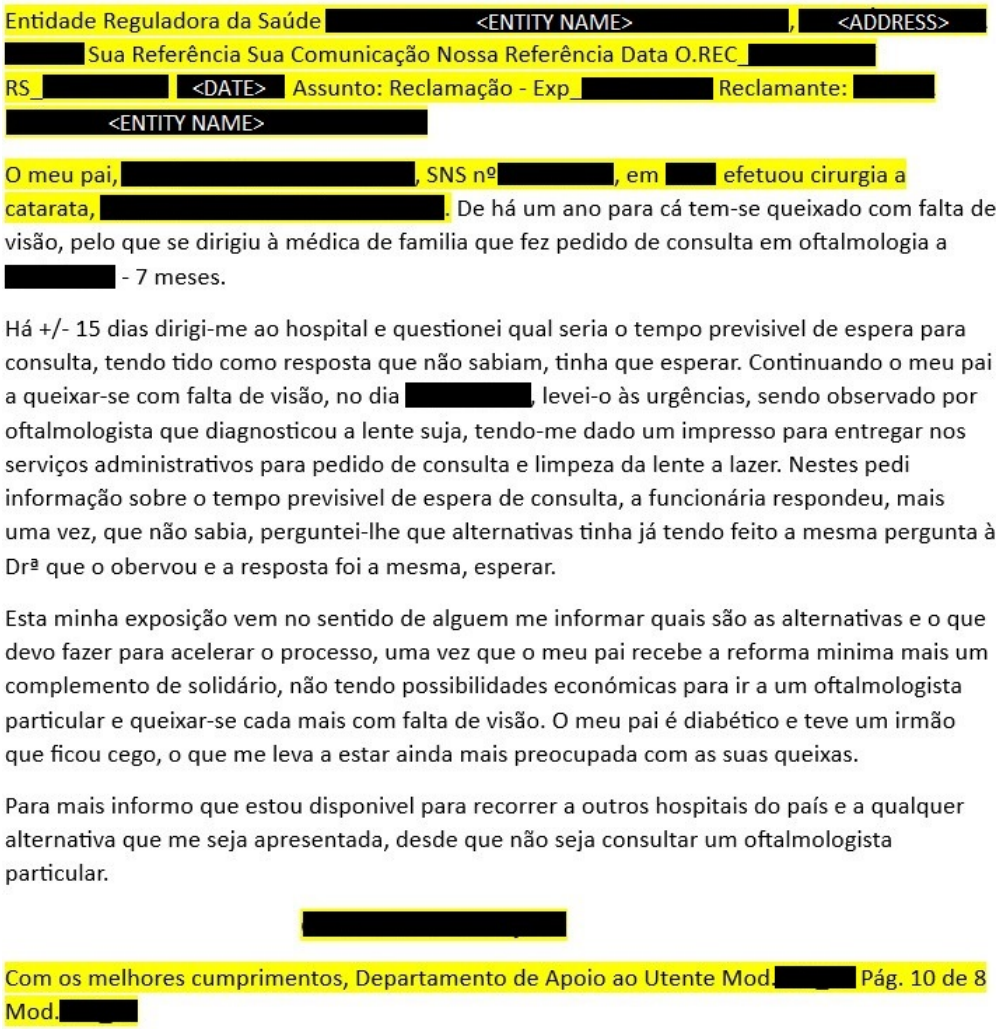
Há +/- 15 dias dirigi-me ao hospital e questionei qual seria o tempo previsivel de espera para consulta, tendo tido como resposta que não sabiam, tinha que esperar. Continuando o meu pai a queixar-se com falta de visão, no dia , levei-o às urgências, sendo observado por oftalmologista que diagnosticou a lente suja, tendo-me dado um impresso para entregar nos serviços administrativos para pedido de consulta e limpeza da lente a lazer. Nestes pedi informação sobre o tempo previsivel de espera de consulta, a funcionária respondeu, mais uma vez, que não sabia, perguntei-lhe que alternativas tinha já tendo feito a mesma pergunta à Drª que o obervou e a resposta foi a mesma, esperar.

Esta minha exposição vem no sentido de alguem me informar quais são as alternativas e o que devo fazer para acelerar o processo, uma vez que o meu pai recebe a reforma minima mais um complemento de solidário, não tendo possibilidades económicas para ir a um oftalmologista particular e queixar-se cada mais com falta de visão. O meu pai é diabético e teve um irmão que ficou cego, o que me leva a estar ainda mais preocupada com as suas queixas.

Para mais informo que estou disponivel para recorrer a outros hospitais do país e a qualquer alternativa que me seja apresentada, desde que não seja consultar um oftalmologista particular.

Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod. Pág. 10 de 8 Mod.

(a) Boilerplate injected into content



Entidade Reguladora da Saúde [redacted] <ENTITY NAME> [redacted], [redacted] <ADDRESS> [redacted] Sua Referência Sua Comunicação Nossa Referência Data O.REC_[redacted] RS_[redacted] [redacted] <DATE> Assunto: Reclamação - Exp_[redacted] Reclamante: [redacted] [redacted] <ENTITY NAME>

O meu pai, [redacted], SNS nº[redacted], em [redacted] efetuou cirurgia a catarata, [redacted]. De há um ano para cá tem-se queixado com falta de visão, pelo que se dirigiu à médica de familia que fez pedido de consulta em oftalmologia a [redacted] - 7 meses.

Há +/- 15 dias dirigi-me ao hospital e questionei qual seria o tempo previsivel de espera para consulta, tendo tido como resposta que não sabiam, tinha que esperar. Continuando o meu pai a queixar-se com falta de visão, no dia [redacted], levei-o às urgências, sendo observado por oftalmologista que diagnosticou a lente suja, tendo-me dado um impresso para entregar nos serviços administrativos para pedido de consulta e limpeza da lente a lazer. Nestes pedi informação sobre o tempo previsivel de espera de consulta, a funcionária respondeu, mais uma vez, que não sabia, perguntei-lhe que alternativas tinha já tendo feito a mesma pergunta à Drª que o obervou e a resposta foi a mesma, esperar.

Esta minha exposição vem no sentido de alguem me informar quais são as alternativas e o que devo fazer para acelerar o processo, uma vez que o meu pai recebe a reforma minima mais um complemento de solidário, não tendo possibilidades económicas para ir a um oftalmologista particular e queixar-se cada mais com falta de visão. O meu pai é diabético e teve um irmão que ficou cego, o que me leva a estar ainda mais preocupada com as suas queixas.

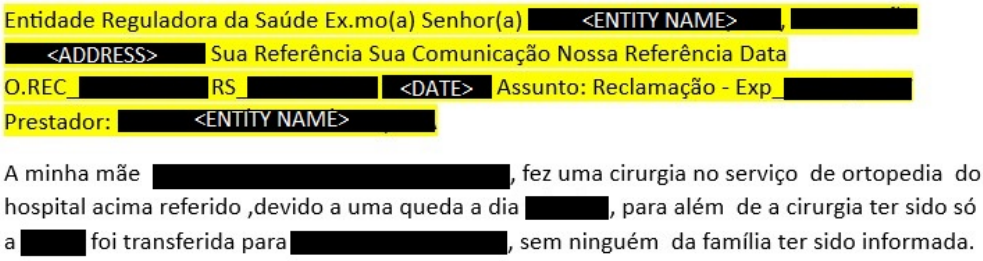
Para mais informo que estou disponivel para recorrer a outros hospitais do país e a qualquer alternativa que me seja apresentada, desde que não seja consultar um oftalmologista particular.

[redacted]

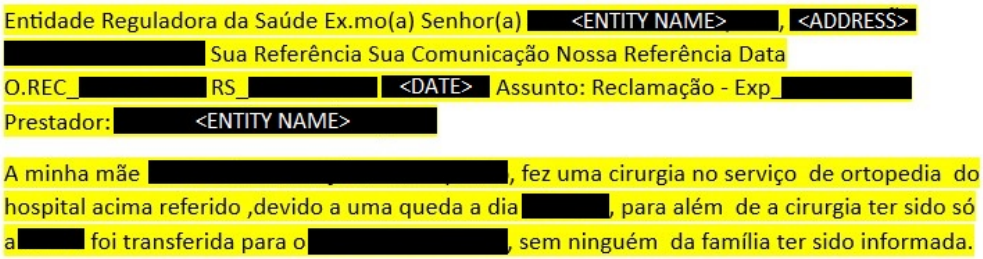
Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod.[redacted] Pág. 10 de 8 Mod.[redacted]

(b) Boilerplate identified by *nltk_multi_mean*

Figure B.4: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example



(a) Boilerplate injected into content



(b) Boilerplate identified by *nltk_multi_mean*

Figure B.5: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example

Ex.mos Sr.s eu [redacted] por este meio venho reclamar da exigência de pagamento a meu ver indevida de taxas moderadoras.

No dia [redacted] desloquei-me á urgência do hospital barlavento algarvio Portimão enviado pelo serviço (saúde 24) Episódio nº [redacted], depois da consulta fui ao guiché da admissão de doentes urgência perguntar se tinha taxa a pagar, onde me responderam que como fui enviado pelo serviço (saúde 24) não pagava, agora recebi carta em casa para efectuar esse pagamento.

2º Situação: No dia [redacted] desloque-me á urgência do mesmo hospital pelo mesmo motivo episodio [redacted] onde depois de ir a triagem fui logo encaminhado para a especialidade de oftalmologia fui levado por uma auxiliar, depois de terminar esta consulta de onde aguardo cirurgia, fui efectuar pagamento taxa moderadora pagamento esse concretizado. Agora recebi carta para fazer pagamento de taxa moderadora de consulta externa nº [redacted], consulta externa essa inventada por eles visto não ter existido essa consulta porque foi consulta de urgência enviado logo no mesmo momento para especialidade. Enviei ao hospital carta registada com aviso de receção a contestar esta situação, junto enviei cópia do recibo de pagamento da consulta nº [redacted] até agora fui ignorado para qualquer resposta. Sem outro assunto os meus cumprimentos. [redacted]

(a) Boilerplate injected into content

Ex.mos Sr.s eu [redacted] por este meio venho reclamar da exigência de pagamento a meu ver indevida de taxas moderadoras.

No dia [redacted] desloquei-me á urgência do hospital barlavento algarvio Portimão enviado pelo serviço (saúde 24) Episódio nº [redacted], depois da consulta fui ao guiché da admissão de doentes urgência perguntar se tinha taxa a pagar, onde me responderam que como fui enviado pelo serviço (saúde 24) não pagava, agora recebi carta em casa para efectuar esse pagamento.

2º Situação: No dia [redacted] desloque-me á urgência do mesmo hospital pelo mesmo motivo episodio [redacted] onde depois de ir a triagem fui logo encaminhado para a especialidade de oftalmologia fui levado por uma auxiliar, depois de terminar esta consulta de onde aguardo cirurgia, fui efectuar pagamento taxa moderadora pagamento esse concretizado. Agora recebi carta para fazer pagamento de taxa moderadora de consulta externa nº [redacted], consulta externa essa inventada por eles visto não ter existido essa consulta porque foi consulta de urgência enviado logo no mesmo momento para especialidade. Enviei ao hospital carta registada com aviso de receção a contestar esta situação, junto enviei cópia do recibo de pagamento da consulta nº[redacted] até agora fui ignorado para qualquer resposta. Sem outro assunto os meus cumprimentos. [redacted]

www.ordemdospsicologos.pt CONHEÇA A CAMPANHA www.encontreumasaida.pt | PROCURA UMA SAÍDA EM encontreumasaida.pt Esta mensagem é confidencial e é propriedade da Ordem dos Psicólogos Portugueses. Destina-se unicamente ao conhecimento do destinatário nela identificado. Caso não seja o destinatário, não está autorizado a ler, reter, imprimir, copiar, divulgar, distribuir ou utiliza-la, parcial ou totalmente. Caso tenha recebido esta mensagem indevidamente, queira informar de imediato o remetente e proceder à destruição de todas as cópias.

(b) Boilerplate identified by *nltk_multi_mean*

Figure B.6: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example

Chegada ao centro dentro do horário, às 20:30, foi-me negada consulta.

Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) <ENTITY NAME> , <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data O.REC_ RS_ <DATE> Assunto: Reclamação - Exp_ Prestador: <ENTITY NAME>

Justificaram dizendo que se o médico atendesse mais alguém, poderia sair depois das 21:00 e que as portas é que fecham às 21:00, as inscrições fecham antes. Bem, neste dia, pelo menos, as portas fecharam às 20:30 e as inscrições bem antes disso!

É indecente uma situação destas, negar o acesso à saúde, dentro de um horário perfeitamente razoável, numa vila onde as urgências hospitalares ficam tão fora de mão.

Está na hora de resolver o problema de e de tantos outros meios pequenos onde se aproveitam da inércia do povo e do facto de terem uma população mais envelhecida e não reclamarem de nada.

Voltei no às 11h, hora se abertura, e a única casa de banho que existe estava imunda, com um cheiro nauseabundo, o chão alagado, sem papel higiénico, nem papel para as mãos.

Centro de saúde terceiro mundista.

Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod. Pág. 4 de 3 Mod.

(a) Boilerplate injected into content

Chegada ao centro dentro do horário, às 20:30, foi-me negada consulta.

Entidade Reguladora da Saúde Ex.mo(a) Senhor(a) <ENTITY NAME> , <ADDRESS> Sua Referência Sua Comunicação Nossa Referência Data O.REC_ RS_ <DATE> Assunto: Reclamação - Exp_ Prestador: <ENTITY NAME>

Justificaram dizendo que se o médico atendesse mais alguém, poderia sair depois das 21:00 e que as portas é que fecham às 21:00, as inscrições fecham antes. Bem, neste dia, pelo menos, as portas fecharam às 20:30 e as inscrições bem antes disso!

É indecente uma situação destas, negar o acesso à saúde, dentro de um horário perfeitamente razoável, numa vila onde as urgências hospitalares ficam tão fora de mão.

Está na hora de resolver o problema de e de tantos outros meios pequenos onde se aproveitam da inércia do povo e do facto de terem uma população mais envelhecida e não reclamarem de nada.

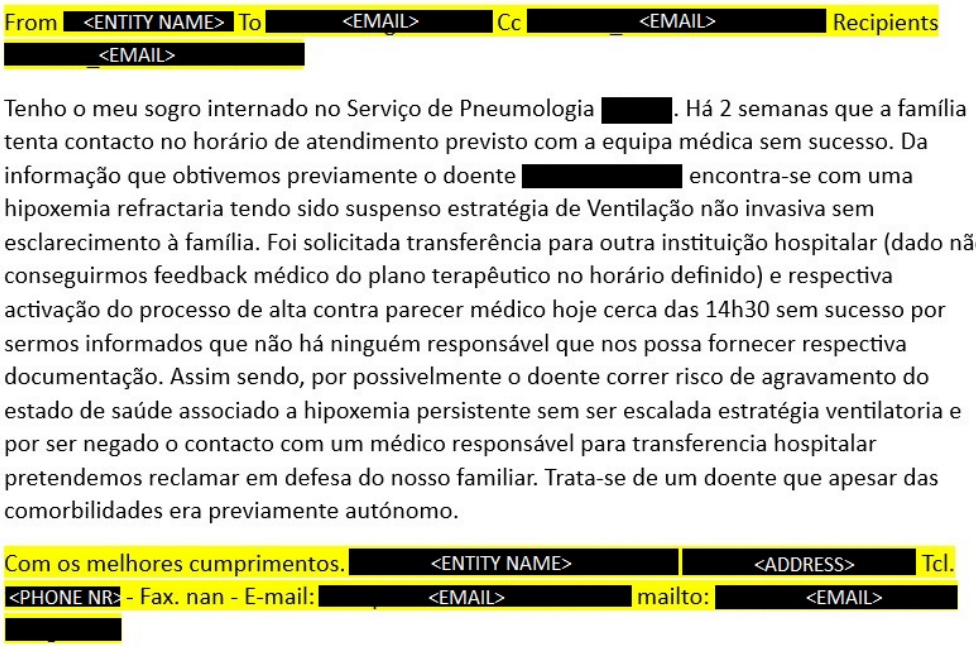
Voltei no às 11h, hora se abertura, e a única casa de banho que existe estava imunda, com um cheiro nauseabundo, o chão alagado, sem papel higiénico, nem papel para as mãos.

Centro de saúde terceiro mundista.

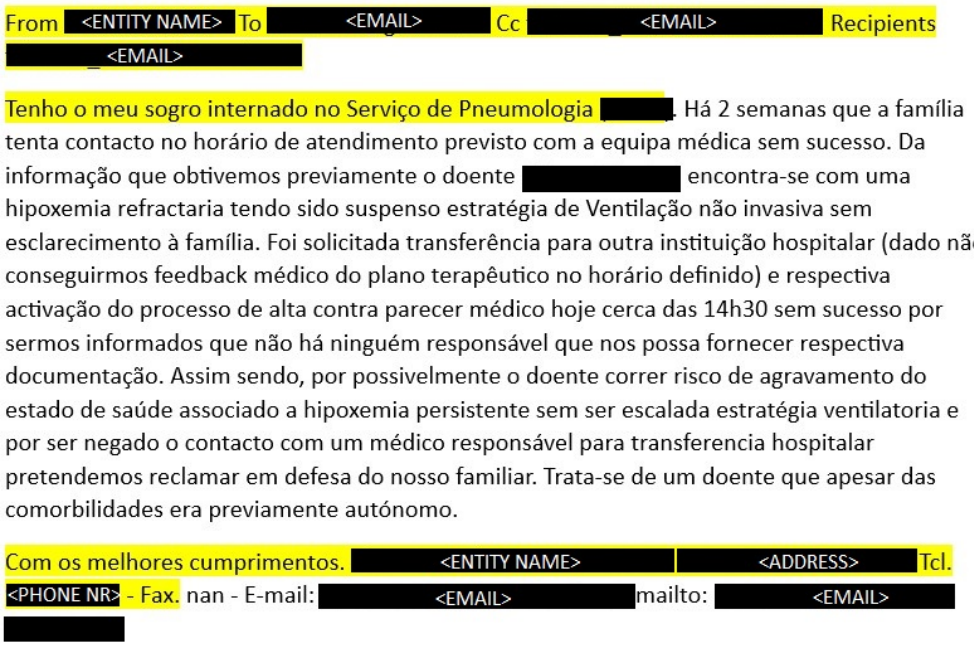
Com os melhores cumprimentos, Departamento de Apoio ao Utente Mod. Pág. 4 de 3 Mod

(b) Boilerplate identified by *nltk_multi_mean*

Figure B.7: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example



(a) Boilerplate injected into content



(b) Boilerplate identified by *nltk_multi_mean*

Figure B.8: Comparison between the ground truth and the results achieved through *nltk_multi_mean* in an example