UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

JULY BIAS MACÊDO

DEVELOPMENT OF NATURAL LANGUAGE PROCESSING-BASED SOLUTIONS FOR RISK ANALYSIS: application to a hydropower company and an O&G industry

Recife
2022

JULY BIAS MACÊDO

DEVELOPMENT OF NATURAL LANGUAGE PROCESSING-BASED SOLUTIONS FOR RISK ANALYSIS: application to a hydropower company and an O&G industry

Doctoral thesis presented to the Programa de Pós-Graduação em Engenharia de Produção to Universidade Federal de Pernambuco for the doctorate degree attainment as part of the requirements of the Engenharia de Produção.

Concentration area: Operations Research.

Advisor: Prof. Dr. Márcio José das Chagas Moura.

Co-Advisor: Prof. Dr. Enrico Zio.

Recife

2022

Catalogação na fonte
Bibliotecário Gabriel Luz CRB-4 / 2222

M141d Macêdo, July Bias.

Development of natural language processing-based solutions for risk analysis: application to a hydropower company and an O&G industry / July Bias Macêdo. 2022.

136 f: figs., tabs.

Orientador: Prof. Dr. Márcio José das Chagas Moura.

Coorientador: Prof. Dr. Enrico Zio.

Tese (Doutorado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia de Produção, Recife, 2022.

Inclui referências.

1. Engenharia de produção. 2. Análise de riscos. 3. Relatório de acidentes. 4. Processamento de linguagem natural. 5. Mineração de texto. 6. Refinaria de petróleo. 7. Companhia hidroelétrica. I. Moura, Márcio José das Chagas (Orientador). II. Zio, Enrico (Coorientador). III. Título.

UFPE

658.5 CDD (22. ed.) BCTG / 2022 - 433

JULY BIAS MACÊDO

DEVELOPMENT OF NATURAL LANGUAGE PROCESSING-BASED SOLUTIONS FOR RISK ANALYSIS: application to a hydropower company and an O&G industry

Doctoral thesis presented to the Programa de Pós-Graduação em Engenharia de Produção to Universidade Federal de Pernambuco for the doctorate degree attainment as part of the requirements of the Engenharia de Produção. Concentration area: Operations Research.

Approved in: 20/12/2022.

EXAMINATION BOARD

Prof. Dr. Márcio José das Chagas Moura (Advisor)
Universidade Federal de Pernambuco

Prof. Dr. Marcelo Hazin (Internal Examiner)
Universidade Federal de Pernambuco

Profª. Drª. Isis Didier Lins (Internal Examiner)
Universidade Federal de Pernambuco

Drª. Marília Ramos
University of California (External Examiner)

Prof. Dr. Piero Baraldi
Politecnico di Milano (External Examiner)

ACKNOWLEDGEMENTS

I have the good fortune to be surrounded by wonderful people that have assisted me in many ways to accomplish this thesis.

I would like to express my deep gratitude to my advisor Prof. Dr. Márcio Moura for his guidance and constructive advice. His support since I was an undergraduate student and intellect is the major reason this thesis was completed.

Within CEERMA I also had helpful advice and comments from Prof. Dr. Isis Didier during the development of this thesis. Special thanks go to Prof. Dr. Caio Maior who at opportune times offered me his wise counsel; Plínio and Lavínia who were always available to help me; and all my colleagues who made me feel at home from my first day at CEERMA.

I would like to sincerely thank Dr. Prof. Enrico Zio and Dr. Prof Piero Baraldi for the incredible opportunity I had to work closely with them and their team and for their very warm hospitality.

I would like to express my gratitude to my family to whom I owe a lot. To my mother Lindaci who has made innumerable sacrifices for me. Thanks to my brother Shalom for his support and to encourage me to follow my dreams.

Finally, I wish my sincere thank you to Caio who has been a constant source of support, patience, and love.

To my grandma and Tonny, in loving memory.

Thanks to Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE) and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support through research scholarships and grants.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

“The mass crushes beneath it everything which is different, everything that is excellent, individual, qualified and select. Anybody who is not like everybody, who does not think like everybody, runs the risk of being eliminated.” (JOSÉ ORTEGA Y GASSET, 1930).

ABSTRACT

Risk Analysis (RA) is crucial to prevent and mitigate potential risk events; however, there are several challenges related to RA. For instance, accident investigation reports are useful sources of information to support safety professionals to propose measures to prevent or mitigate identified occupational accident root causes. Nevertheless, reports' low quality and lack of detail may limit their usefulness. Moreover, the quality of Quantitative Risk Analysis (QRA) strongly relies on the identification of all potential hazards with major consequences related to the operation of an industrial system, which is usually performed by multiple experts and consumes a considerable amount of time and effort. Since valuable knowledge about an industrial system is stored in the form of textual data, Natural Language Processing (NLP) techniques can be helpful since it can be applied to extract, organize, and classify information from text. Although several studies contributed to the advance of RA, most studies applying NLP focus primarily on automatically identifying patterns from reactive data, such as accident reports, and do not consider the quality of information contained in these documents. In addition, different forms of text data store relevant knowledge about industrial systems and their respective risks, especially proactive data such as documents resulting from preliminary risk studies, and adoption of these data could support preventive risk studies. The main purpose of this study is to develop NLP-based solutions to different issues faced in RA. Thus, this thesis presents two methodologies to (i) identify issues in a hydropower company's accident investigation reports that may compromise their usefulness as a decision support tool (ii) automatically identify risk features from documents to support the initial stage of QRA in Oil and Gas (O&G) industries. Occupational safety technicians can benefit from the methodology that helps to identify issues and propose improvements to the accident reports. In addition, the second methodology can help experts to identify and assess hypothetical accidental scenarios related to the operation of an industrial facility. Thus, this thesis may contribute to the prevention and mitigation of occupational and/or major accidents and consequently avoid/reduce property damage, economic and social disruption, environmental degradation, and human losses.

Keywords: risk analysis; accident investigation reports; natural language processing; text mining; oil refineries; hydroelectric power company.

RESUMO

A Análise de Riscos (RA) é essencial para a prevenção e mitigação de potenciais eventos de risco, porém há vários desafios relacionados à execução da análise. Por exemplo, relatórios de acidentes, são fontes úteis de informação para apoiar os especialistas de segurança a propor medidas preventivas/mitigativas das causas acidentais ocupacionais identificadas. Porém, a falta de detalhes e a baixa qualidade dos relatórios podem limitar a sua utilidade. Além disso, a qualidade da Análise Quantitativa de Risco (QRA) depende fortemente da identificação de todos os potenciais perigos com consequências graves, relacionados à operação do sistema industrial, o que consome uma quantidade considerável de tempo e esforço. Nesse contexto, o Processamento de Linguagem Natural (NLP) pode ser útil pois pode ser aplicado para extrair, organizar e classificar a informação do texto. Embora vários estudos tenham contribuído para o avanço da RA, a maior parte dos estudos que aplicam NLP à RA foca principalmente na identificação automática de padrões a partir de dados reativos, tais como relatórios de acidentes, e não consideram a qualidade da informação contida nestes documentos. Além disso, diferentes formas de dados de texto armazenam conhecimento relevante sobre os sistemas industriais e seus respectivos riscos, especialmente dados proativos, como documentos resultantes de estudos preliminares de riscos, e a adoção desses dados poderia apoiar estudos de risco preventivos. Por isso, esta tese apresenta duas metodologias baseadas em NLP para (i) identificar problemas em relatórios de acidentes que possam comprometer a utilidade desses documentos como ferramenta de suporte a decisão e (ii) para identificar características de risco a partir de documentos para apoiar a fase inicial da QRA. A primeira metodologia dá suporte aos técnicos de segurança para identificar problemas e propor melhorias/correções nos relatórios de acidente, contribuindo para uma melhor gestão de acidentes ocupacionais. Além disso a segunda metodologia pode auxiliar especialistas a identificar e avaliar cenários acidentais relacionados a operação de um sistema industrial. Dessa forma essa tese contribui para a prevenção e mitigação de acidentes e consequentemente evita/reduz danos a propriedade, econômicos e sociais, degradação ambiental e perdas humanas.

Palavras-chave: análise de riscos; relatório de acidentes; processamento de linguagem natural; mineração de texto; refinaria de petróleo; companhia hidroelétrica.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

BERT	Bidirectional Encoder Representations from Transformers
BoW	Bag-of-Words
DA	Data Augmentation
DL	Deep Learning
FMEA	Failure Mode and Effect Analysis
FT	Fault Tree
FNN	Feedforward Neural Network
HAZOP	Hazards and Operability Analysis
HALO	Hazard Analysis based on Language processing for Oil refineries
LDA	Latent Dirichlet Attribution
ML	Machine Learning
M	Moderate
NLP	Natural Language Processing
NT	Non-Tolerable
O&G	Oil and Gas
PrHA	Preliminary Hazard Analysis
QRA	Quantitative Risk Analysis
RNN	Recurrent Neural Network
SVM	Support Vector Machines
TM	Text Mining
TF-IDF	Term Frequency-Inverse Document Frequency
T	Tolerable
WSS	Within-Cluster-Sum of Squared Errors

CONTENTS

1 INTRODUCTION

1.1 INITIAL REMARKS

Advances in technology and the growth in demand for different products have contributed to the increasing complexity of industrial systems, which brings huge challenges to the safe operation of these systems (HAO, NIE, 2022). Thus, the practice of Risk Analysis (RA), which involves hazard identification, risk assessment and management, is crucial to effectively guide investments for the prevention and mitigation of potential risk events. Shortfalls in RA, such as accuracy, decision making and communication, can have negative effects on economy, society, environment and business image (THEKDI, AVEN, 2022).

Professionals from occupational safety and process safety, which are two separate disciplines with different approaches, can work together to create a safe environment and occupational health. The main difference between occupational safety and process safety is that process safety deals with undesired events, usually with a focus on ‘loss of containment’, that happen at a lower frequency, and are more likely to result in major consequences (SWUSTE, THEUNISSEN, *et al.*, 2016). On the other hand, occupational safety typically addresses events involving personal safety at an individual level with small consequences (ALI, ARIFIN, *et al.*, 2022).

The risk models developed have been changing rapidly, one reason is the advance in computing performance and the ability to record, store and process massive amount of data. Another reason is the breakthroughs in the fields of Machine Learning (ML) and artificial intelligence that have enabled the efficient extraction of information from complex, high-dimensional and unstructured datasets (NATEGHI, AVEN, 2021).

Overall, RA should help inform risk management to prioritize the most critical scenarios. To achieve this purpose different risk assessment tools and data sources can be applied. However, there are several challenges related to RA such as the quality and scope of information, absence of knowledge exchange in practice and implementation of some risk assessment tools (GREENBERG, COX, *et al.*, 2020, KHODADADYAN, RESEARCHER, *et al.*, 2021).

For instance, most national and international regulators require companies to store a collection of investigation reports of occupational accidents. These reports are useful sources of information to extract valuable characteristics related to the event that can be explored to support safety professionals to take appropriate actions to remove or attenuate identified root causes (HÄMÄLÄINEN, TAKALA, *et al.*, 2017, SILVA, JACINTO, 2012). However, reports' low quality and lack of detail may limit their usefulness because reasonable resources are required for manual analysis, which is a complex and error-prone task (YOUNG, I. J. B.; LUZ; LONE, 2019).

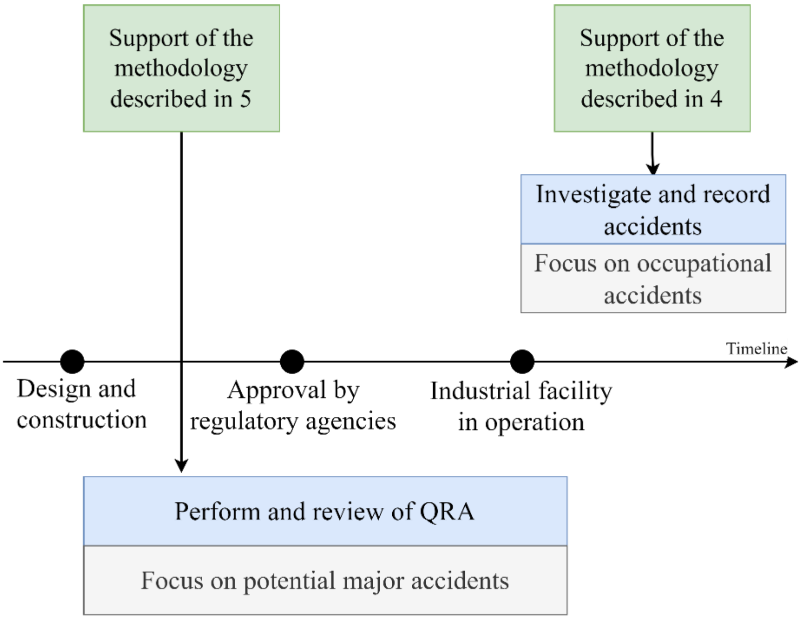
In the context of process safety, Quantitative Risk Analysis (QRA) has been widely applied to large technological systems. At its core, QRA seeks to answer three questions: 'what can go wrong?', 'how likely is it' and 'what are the consequences?'. Thus, the quality of QRA relies on the identification of all potential hazards performed in the initial stage of the analysis. This process is usually accomplished by different experts, which consumes a considerable amount of time and effort, and the results of this stage are recorded and periodically reviewed (APOSTOLAKIS, 2004).

Commonly, information about an industrial system is stored in the form of textual data. Although these documents store valuable knowledge, the number of available documents is overwhelming and difficult to be manually processed. Thus, Natural Language Processing (NLP) techniques can be helpful since it can be applied to extract, organize, and classify information from text, allowing the automatic identification of patterns (DRURY, ROCHE, 2019). Indeed, approximately 76% of activities in industries require natural language understanding (BAKER, HALLOWELL, *et al.*, 2020).

NLP has been successfully applied in different fields such as healthcare, marketing, education, and industry (GAGNE, HALL, *et al.*, 2019, HEIDINGER, GATZERT, 2018, YIM, WARSCHAUER, 2017, ZARE, 2019). NLP is an interdisciplinary field that uses different analysis tools and involves techniques from ML and Text Mining (TM). In a nutshell, NLP can be considered as a subdiscipline of artificial intelligence and computational linguistics that includes any manipulation of natural language to allow computers generating statements and/or words written in human languages (KHURANA, KOLI, *et al.*, 2017). Therefore, this study proposes the application of NLP to develop solutions for problems faced when carrying out RA.

Here we will present two methodologies based on NLP techniques that were developed to solve different real-world problems in the context of RA. The first proposed methodology uses different NLP approaches to identify issues in a hydropower company's accident investigation reports that may compromise their usefulness as a decision support tool. While the second methodology proposed aims to identify risk features in oil refineries in order to support the initial stage of QRA in Oil and Gas (O&G) industries. Thus, we believe that this thesis represents a positive contribution to RA, at different stages of the life cycle of an industrial system, by enabling an efficient use of human resources and risk management. For instance, Figure 1 illustrates when the methodologies described in this thesis could be applied in a generic process industry.

Figure 1 – Example of application of the methodologies in the timeline of an industrial facility


Support of the methodology described in 5
Support of the methodology described in 4
Investigate and record accidents
Focus on occupational accidents
Timeline
Design and construction
Approval by regulatory agencies
Industrial facility in operation
Perform and review of QRA
Focus on potential major accidents

Source: The author (2022).

Occupational safety technicians can benefit from the methodology that helps to identify problems and propose improvements and/or corrections to accident reports. In addition, the developed research can help risk analysts to identify and assess unexpected events related to the operation of an industrial facility. Thus, this thesis may contribute to the prevention and mitigation of occupational and/or major accidents (i.e., events with severe damage to people, to the environment and surroundings) and consequently avoid/reduce property damage, economic and social disruption, environmental degradation, and human losses.

1.2 MOTIVATION

This section presents the motivation behind both proposed NLP-based methodologies mentioned above.

1.2.1 Assessment of Accident Investigation Reports of a Hydropower Company

Work accidents can lead not only to huge financial losses for the organization but also to serious threats to people's integrity and environment (MAIOR, SANTANA, *et al.*, 2018). According to the Workplace Safety and Health Institute in partnership with International Labour Organization (HÄMÄLÄINEN, TAKALA, *et al.*, 2017), there are more than 2.78 million deaths and around 374 million non-fatal work accidents each year. On the other hand, these accidents are useful sources of information to extract valuable characteristics related to the event (SILVA, JACINTO, 2012).

In this way, systematic accident investigation reports retain knowledge that can be explored to support decision-making. In fact, most national and international regulators require companies to store a collection of accident investigation reports allowing safety professionals to take appropriate actions to remove or attenuate identified root causes after their analysis (ABDAT, LECLERCQ, *et al.*, 2014, BAVARESCO, ARRUDA, *et al.*, 2021). Typically, these reports are written in natural language, since free textual responses allow one to describe the event as one perceives it. Reasonable resources are required for manual analysis of accident investigation reports, which is a complex and error-prone task. Thus, a complete human review of the entire database is almost impossible, considering numerous reports produced by a company (BERTKE, MEYERS, *et al.*, 2012, KOC, GURGUN, 2022).

In addition, real reports oftentimes present drawbacks related to the quality of information filled out that may limit their usefulness. For example, incomplete or missing data, lack of standardization, and conflicting information are commonly found in these reports. In practice, these issues are a significant hurdle for analysts to extract useful insights, and then propose effective preventive measures to improve safety (YOUNG, LUZ, *et al.*, 2019).

Given that, our goal is to develop an approach that enables safety analysts and decision-makers to critically evaluate and identify potential issues and/or undesirable patterns on accident investigation reports. To that end, we use NLP techniques, such as Bag-of-Words (BoW) (ZHANG, Yin, JIN, *et al.*, 2010), Term Frequency-Inverse Document Frequency (TF-IDF) (XIANG, 2022), Doc2Vec (CHANG, XU, *et al.*, 2018), and Latent Dirichlet Attribution (LDA) (XING, LEE, *et al.*, 2020). The methodology considers three main steps: (i) unsupervised exploratory analysis of categories in the accident investigation reports; (ii) supervised ML to classify the accidents considering the categories found in the original reports; and (iii) restructuring the original groups and, then, reclassifying the accidents using the categorization proposed after (i).

In summary, step (i) supports the analysis of the reports by providing an overview of the report content and main topics, which allows us to identify possible problems and propose improvements/corrections, and steps (ii) and (iii) allow us to compare the effects of possible changes in the accident reports by comparing the performance of the ML classifiers. Thus, we propose a systematic analysis that would both (a) support safety technicians in identifying possible problems in a set of accident investigation reports and (b) allow a quantitative assessment of the quality of the information extracted from the reports. Thus, the result obtained through the proposed methodology would be better accident investigation reports that could be used by methodologies already proposed by other authors.

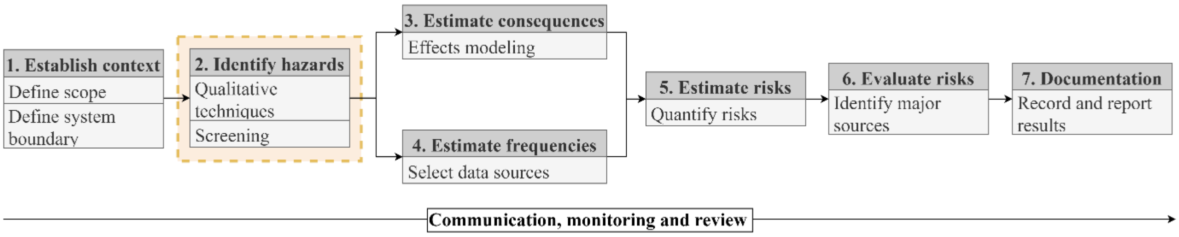
The proposed methodology to assess accident investigation reports was published in International Journal of Occupational Safety and Ergonomics (MACÊDO, RAMOS, *et al.*, 2022). In addition, we analyze the modified accident investigation reports resulting from the methodology to illustrate its practical usefulness to support decision making. The goal is to determine whether there will be an injury leave, in which this information is fundamental to effectively defining strategies for the worker's absence. The results and methodology described was accepted in Journal of Risk and Reliability (RAMOS, Plinio, MACÊDO, *et al.*, 2022).

1.2.2 Support QRA in O&G Industries

Oil refineries are expensive, complex systems that provide essential resources by converting crude oil into useful products such as fuels, lubricants, and asphalt. Oil refining encompasses a great variety of physical and chemical processes, and it is divided into three basic steps: separation, conversion, and treatment (DEMIRBAS, BAMUFLEH, 2017). These stages contain highly flammable, explosive, and/or toxic substances, which are handled and stored in extreme conditions. In these high-risk environments, small errors can proliferate into process inefficiency, poor working conditions, and, ultimately, major consequences such as life and property losses and environmental impact (LONGO, PADOVANO, *et al.*, 2021, PRAMOTH, SUDHA, *et al.*, 2020). The consequences of the loss of containment of these materials depend on different variables such as the nature of the released material and its physical state, and the environmental conditions (CASAL, 2017).

Although great efforts have been made towards the prevention of major accidents, they are far from being eradicate. Thus, it is indispensable to develop new methods to support and improve risk studies. QRA is a systematic approach for identifying and analyzing accidental scenarios, and it is characterized through the methodical use of data and knowledge to describe causes, probabilities, and consequences of potential accidents (AVEN, KESSENICH, 2020, HE, LI, *et al.*, 2018). National and international regulators demand QRA for both new and existing installations in order to provide a thorough picture of the hazards and, then, manage and minimize the potential risks related to their operation (VINNEM, RØED, 2020, WANG, Qianlin, ZHANG, *et al.*, 2018). In a nutshell, QRA involves seven main steps (CCPS, 2008, TNO, 2005), illustrated in Figure 2.

Figure 2 – Main steps of QRA process.



Source: adapted from (CCPS, 2008; VILLA et al., 2016; ZENG; ZIO, 2017).

The second step comprises hazards identification; in this step experts recognize relevant scenarios that may arise, assessing and reporting their likelihood and potential consequences (STEIJN, VAN KAMPEN, *et al.*, 2020). To that end, experts usually adopt qualitative or semi-quantitative technique such as PrHA (YAN, XU, 2019), HAZard and OPerability analysis (HAZOP) (GUIOCHET, 2016), Failure Mode and Effect Analysis (FMEA) (BHATTACHARJEE, DEY, *et al.*, 2020), and Fault Tree (FT) (RAMOS, Marília, THIEME, *et al.*, 2020). The selection of the hazard analysis tool depends on the system lifecycle as well as the information available (VILLA, PALTRINIERI, *et al.*, 2016). Hazard identification is essential for QRA success, since its main objective is to identify most of the possible undesired events that may occur during the system operation (MARHAVILAS, FILIPPIDIS, *et al.*, 2019). PrHA is widely adopted to identify different hazardous situations, categorize them, and prioritize the most critical ones to be further examined quantitatively (Figure 2, step 5) (BENEKOS, DIAMANTIDIS, 2017, KURIAN, SATTARI, *et al.*, 2020).

The process depicted in Figure 2 can be very time-consuming in practice, mainly depending on the complexity of the system analyzed and on the diverse backgrounds of the experts in the team that execute the risk assessment. In this context, the methodology here developed aims at reducing the efforts required to perform the initial stage of QRA.

Given that, this study applies TM to extract information from text data and fine-tune pre-trained Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN, CHANG, *et al.*, 2018) to identify risk features in an oil refinery. To that end, the pre-trained BERT model was fine-tuned with specific-domain datasets to perform three tasks: (i) to predict the potential consequences of accidents related to the operation of an oil refinery, and (ii) to classify each scenario in terms of severity of the consequence, and (iii) likelihood of occurrence. Each dataset used to train and test the three models was built based on PrHA documents available for an oil refinery. It is noteworthy that there are different documents that store information as textual data. We here focus on PrHA spreadsheets because they summarize and store information from experts and other refinery documents; thus, PrHA documents contain valuable textual information. These documents were developed by a group of experts specific to each production unit of the oil refinery. To perform the risk analysis, they followed a standard (ANP, 2014) that guides them in filling out the forms, and defining the potential consequences, and the frequency and severity categories for each of the scenarios.

We expect that the developed models could be highly useful for new plants that depend on the approval of the environmental regulators to start the development of the facility design and construction. In these situations, almost no specific information is available for that plant, and then risk experts usually rely on partially relevant risk studies performed for similar facilities. Thus, with a model trained based on all the available information garnered from past risk studies, experts could use that entire source of knowledge to reduce uncertainty. Instead of starting the analysis from scratch, risk analysts could reuse knowledge, imbued in the trained models, from previous studies or use QRA performed for similar plants as a starting point to identify potential consequences, qualitatively characterize frequency and severity of accidental scenarios, and prioritize the most critical events. It is worth emphasizing that although the models' predictions are limited by training data, this would not hinder the QRA, since the postulated scenarios need to undergo a validation stage performed by experts.

We claim that such a method is particularly important for oil refineries because regulatory agencies require highly detailed PrHA in order to provide valuable information to decision makers and surrounding communities and to be suitable for QRA (BAYBUTT, 2018). Thus, risk analysts are demanded to postulate and analyze hundreds or even thousands of scenarios. For instance, for a medium-size oil refinery that processes around 200k barrels of oil per day, a PrHA resulted in the identification of more than 3,000 accidental hypotheses. In addition, experts can critically evaluate scenarios that have been misclassified by the models and determine whether the model's prediction makes sense, i.e., indicating an error made by experts during the PrHA. Thus, the predictions provided by the models would also allow the experts to correct the PrHAs.

Some results of the proposed methodology have been presented (MACÊDO, AICHELE, *et al.*, 2020b, a, 2021, MACÊDO, MOURA, AICHELE, *et al.*, 2021, MACÊDO, MOURA, LINS, *et al.*, 2022, RAMOS, Plinio, MACÊDO, *et al.*, 2022) and published in Process Safety and Environmental Protection (MACÊDO, MOURA, AICHELE, *et al.*, 2022). In addition, we developed a web app, known as HALO1, process number BR512022000211-6, registered in the national institute of industrial property (MACÊDO, MOURA, LINS, *et al.*, 2021).

[1] http://nlprisk.ceerma.com/

1.3 OBJECTIVES

1.3.1 General Objective

The main objective of this thesis is to develop NLP-based solutions to two problems that may be faced in the context of Risk Analysis. Thus, this thesis proposes:

- A methodology to support safety technicians in identifying possible problems in a set of accident investigation reports and to propose improvements using NLP and ML methods.
- A methodology to automatically extract text data from documents to build NLP-based models to identify risk features and thus reduce effort required to perform initial stage of QRA.

1.3.2 Specific Objectives

Given the general objectives of this research, some specific objectives are similar for both problems and can be listed as follows:

- Data extraction: A script for each RA problem was developed to extract all text data used as to build the models. Previously prepared PrHAs documents for an oil refinery were provided, as doc files, and accident investigation reports for a hydroelectric power company were provided, as excel files, to this study;
- Data preprocessing: Preprocessing operations were performed in order to convert the text into a cleaner format and to obtain a numerical representation suitable to train machine learning algorithms;
- Data organization: The data extracted was assessed and organized in order to build a dataset for each task.

Regarding the methodology to assess accident investigation reports. We can list the following secondary objectives:

- Topic modeling: Identification and critical assessment of the main topics on the accident investigation reports using LDA algorithm to identify patterns, searching for useful information to perform a further assessment;

- Problem definition: Identification of potential issues and solutions/corrections based on the exploratory analysis and definition of a classification task;
- Baseline classifiers: Train different classifiers combining different ML algorithms and feature vector representations to perform supervised task using the original data;
- Manual curation: Implementation of the proposed changes/corrections on the accident investigation report;
- Final classifiers: Build ML classifiers to perform supervised task using the curated data to compare with the baseline results and check the impact on the performance of the algorithms;
- Application of resulting accidents reports: Use the reports resulting from the methodology to perform a classification task in order to illustrate the practical usefulness of the improved reports to support decision making.

Regarding the methodology to support QRA. We have the following secondary objectives:

- Modeling process: Adjustments on BERT architecture to perform text (i) multiclassification task and (ii) multilabel classification task;
- Fine-tune BERT: Part of each dataset was used to train the adjust pre-trained BERT. Thus, a different classifier was obtained for each dataset. The remaining observations were used to test the performance of the classifiers.
- Rare scenarios investigation: Experiments combining Data Augmentation (DA) and under sampling were performed to identify the most suitable approach to handle rare accident events;
- HALO web app: development of the Hazard Analysis based on Language processing for Oil refineries (HALO), a web app to support risk analyst in identifying and assessing different accident scenarios related to chemical spills in oil refineries.

1.4 OUTLINE OF THE THESIS

Besides this introduction chapter, this thesis has five additional chapters briefly described as follows:

- Chapter 2: Provides the concise theoretical background about RA and NLP techniques;
- Chapter 3: Presents works that applied TM and NLP to the RA context;
- Chapter 4: Presents the proposed methodology based on NLP to assess accident investigation reports. Applies the proposed methodology to a hydropower company accident database. To complete the analysis, shows an application of resulting accident reports to predict injury leave given the information contained in these documents;
- Chapter 5: Explains the proposed methodology to support the initial stage of QRA for identifying risk features using NLP. Applies the proposed methodology to a specific oil refinery. Discuss how to extract valuable features regarding rare risk events. And finally, describes the web-app developed to support the initial stages of QRA;
- Chapter 6: Concludes remarks.

2 THEORETICAL BACKGROUND

This chapter provides key concepts and definitions on RA and describes classical and advanced NLP techniques. The applications provided in Chapters 4 and 5 use these concepts.

2.1 RISK ANALYSIS

There is an enormous effort and interest in different industries and society to manage risks. However, there are many difficult issues and challenges in risk management, related in particular to the foundation and performance of RA. RA include identification of hazards, cause and consequence analysis, and risk assessment; thus, RA allows managing hazards properly to prevent potential accidents from happening (ISO, 2018).

Overall, any unexpected event that is caused by an unsafe act or condition may disrupt the workflow in an industrial process, regardless of whether the event causes injury or property damage. Thus, such events should be seen as a warning that an accident may occur. Accident investigation is a safety technique designed to find out and report the causes that led to the given accident. The internal reporting and investigation of accidents aim to prevent accidents and the occurrence of similar events in the future. Given that, the importance of a good investigation lies in being able to extract some preventive benefit from past unexpected events (JONES, KIRCHSTEIGER, *et al.*, 1999, SALGUERO-CAPARROS, SUAREZ-CEBADOR, *et al.*, 2015).

Typically, these reports are written in natural language and oftentimes present drawbacks related to the quality of information filled out that may limit their usefulness, such as inconsistent or missing information and non-standardization. Overall, the proposed works so far by several authors (e.g., AHMADPOUR-GESHLAGI et al., 2020; ANDRZEJCZAK; KARWOWSKI; THOMPSON, 2014; BAKER; HALLOWELL; TIXIER, 2020; HUGHES et al., 2018; LOMBARDI; FARGNOLI; PARISE, 2019; MUGURO et al., 2020; SINGLE; SCHMIDT; DENECKE, 2020; TIXIER et al., 2016) aim to facilitate the risk management by extracting information from accident investigation reports. These authors assume that set of reports contain good information and focus on the performance metrics of supervised (TIXIER, HALLOWELL, *et al.*, 2016a) and/or unsupervised (LOMBARDI, FARGNOLI, *et al.*, 2019) algorithms. With that being said, we believe that NLP can be applied to support safety analysts and decision-makers to critically evaluate and identify potential issues and/or undesirable patterns on accident investigation reports.

Other widely used tool, particularlly in the O&G industry, to analyse and manage risks is QRA that oftentimes resorts to systematic approaches for characterizing a risk. The early stage of QRA consist of hazards identification and analysis, which represent some of the most difficult steps, due to the many possibilities (scenarios) of what may go wrong (PASMAN, ROGERS, 2018, RAMOS, Marília, LÓPEZ DROGUETT, *et al.*, 2020, ZENG, ZIO, 2017). To perform these steps, a team of experts usually attends several meetings aimed at brainstorming all hazards and potential leakages, their possible causes, expected frequency and consequences. To that end, the experts need to consider several engineering documents to gather relevant information about the system and its environment (PASMAN, ROGERS, 2018, ZIO, AVEN, 2013).

There are different techniques that are widely adopted in the early stages of QRA. The choice of the right technique depends on different factors, as available resources, the amount and quality of the data, and the complexity of the system analyzed. The aim of PrHA is to identify all possible leakages and the accidental events that may occur and to provide a qualitative estimate of the severity and likelihood of each accidental scenario (LI, Xinhong, CHEN, *et al.*, 2018).

Simpy put, PrHA is an approach to screen out the low-risk scenarios, while the most critical events are further analyzed by a quantitative approach to estimate their physical effects generally related to fire, explosion, and toxic dispersion. Finally, this piece of information is conflated for all critical events to calculate the individual and social risks associated with the entire facility, which are compared to the risk tolerability criteria established by regulatory agencies. For oil refineries, these same steps are also performed for existing facilities with the objective of presenting evidence that both risk estimates are still below the thresholds. For instance, this is a demand required by the environmental regulator in order to permit the plant's life extension.

According to ISO 31010 (ISO, 2018), Table 1 and Table 2 show the consequence and likelihood classes respectively that are commonly adopted in PrHA. Their combination represents the risk category (ARUNRAJ, MAITI, 2007). Note that the categories are defined in terms of the damage to human life. It is worth to mention that other assets could also be analyzed such as environment, property, or reputation. However, the scope of this work is limited to human life.

Table 1 – Description of the consequence levels in terms of the effects to human life.

Consequence		
Category		Effects
I	Negligible	without injuries
II	Minor	minor injuries or first aid treatment
III	Moderate	serious injuries inside or mild injuries outside the facility
IV	Significant	fatality inside or serious injuries outside the facility

Source: adapted from (ISO, 2018).

Table 2 – Description of the likelihood categories.

Likelihood		
Category		Description
A	Remote	conceptually possible, but there are no records in the literature
B	Unlikely	unlikely to occur in normal conditions
C	Possible	might occur sometime
D	Likely	will probably occur

Source: adapted from (ISO, 2018).

The results of the PrHA are usually reported as spreadsheets, as illustrated in Table 3, which is an example that represents the description of a potential accident due to the release of contaminated and oily water from a basin of the industrial wastewater treatment unit in an oil refinery. Table 3 also contains the operating conditions, a list of equipment existing in the analyzed subsystem, and the pipeline material. Note that, given the initiating event (i.e., small leakage or large leakage), a variety of consequences may occur (e.g., a small leakage might cause a toxic vapour clou and/or irritation), and may have different impact to human life, which is defined by the ‘severity’ column, whereas the rate of occurrence of each initiating event is indicated in the column ‘likelihood’.

For an oil refinery with a capacity of processing about 230,000 barrels per day, the PrHA resulted in a dataset of 1,635 reports similar to that of Table 3. These spreadsheets summarize the assessment performed by the experts and represent their tacit knowledge; thus, these documents contain valuable information about the risks related to the operation of different subsystems present in the oil refinery. Given that, NLP techniques can be applied to automatically extract the text and assess the content of previous analysis to develop tools that may support future QRA.

Table 3 – Example of data contained in PrHA documents.

<table>
<tr><td colspan="2"><b>Unit</b></td><td colspan="4">Industrial wastewater treatment</td></tr>
<tr><td colspan="2"><b>System</b></td><td colspan="4">Flow regularization system</td></tr>
<tr><td colspan="2"><b>Subsystem description</b></td><td colspan="4">Basin with possible presence of toxic substance hydrocarbon from another unit</td></tr>
<tr><td rowspan="3"><b>Pipeline</b></td><td rowspan="3"><b>Material</b></td><td rowspan="3">Carbon steel</td><td colspan="3"><b>Operating conditions</b></td></tr>
<tr><td><b>Temperature</b> $(^{\circ}C)$</td><td><b>Pressure</b> $(kgf.cm^{-2})$</td><td><b>Flow rate</b> $(kg.h^{-1})$</td></tr>
<tr><td>25</td><td>1.03</td><td>3,000</td></tr>
<tr><td colspan="2"><b>Equipment</b></td><td colspan="4">Sump pump</td></tr>
<tr><td colspan="2"><b>Chemical Product</b></td><td colspan="4">Contaminated and oily water</td></tr>
<tr><td colspan="2"><b>Initiating event</b></td><td colspan="2"><b>Potential consequences</b></td><td><b>Severity</b></td><td><b>Likelihood</b></td></tr>
<tr><td colspan="2" rowspan="2">Small leakage</td><td colspan="2">Irritation</td><td>II</td><td>D</td></tr>
<tr><td colspan="2">Toxic vapour cloud</td><td>II</td><td>D</td></tr>
<tr><td colspan="2" rowspan="2">Large leakage</td><td colspan="2">Irritation</td><td>III</td><td>A</td></tr>
<tr><td colspan="2">Toxic vapour cloud</td><td>III</td><td>A</td></tr>
</table>

Source: The author (2022).

2.2 NATURAL LANGAGE PROCESSING

NLP describes a field of artificial intelligence that uses computational algorithms to learn, understand, and produce human language content. The application areas in NLP include topics, such as extraction of useful information from text (e.g., named entities and topics), translation of text, summarization of written works, automatic answering of questions by inferring answers, and classification of texts (OTTER, MEDINA, *et al.*, 2021).

NLP has been successfully applied to extract knowledge of large amount of textual data and has proven to be useful in many fields, reducing the time and human effort required for content analysis of documents. For instance, text classification is one NLP task with several real-word applications, such as sentiment analysis of movies review and spam, bots, and fraud detection (MINAEE, KALCHBRENNER, *et al.*, 2021). This task involves extracting rules from a set of labelled documents/texts (also called as annotated corpus), and once the classifier is trained, it can classify new textual data based on the patterns detected (BENGFORT, BILBRO, *et al.*, 2018).

Basic NLP procedures include processing text data, converting text to features, and identifying semantic relationships (CAI, 2021). Moreover, language modeling is an essential piece of almost any application of NLP. Language modeling is the process of creating a model to predict words or simple linguistic components given previous words/components and/or deriving their full value from their interactions with other words (OTTER, MEDINA, *et al.*, 2021). Section 2.2.1 explains text preprocessing, which is a fundamental part of the development of NLP systems; Section 2.2.2 introduces topic modeling; Section 2.2.3 traditional representation models; Section 2.2.4 shows advances in NLP and the evolution of language models; Sections 2.3 and 2.4 describe Transformers architecture and BERT model respectively. These sections provide key concepts to understand this study.

2.2.1 Text Preprocessing

The presence of meaningless information, often found in raw text data, considerably affects the performance of predictive models. Therefore, preprocessing provides a significant contribution to improving data quality, by reducing computational costs, homogenizing the documents, and removing noisy or unwanted information (MADEIRA, MELÍCIO, *et al.*, 2021).

Text preprocessing is a set of operation applied on the textual data to eliminate noise from text, since text data often contains special characters, special formats (e.g., numbers and dates) and the most common words such as prepositions, articles, and pronouns are unlikely to provide useful knowledge. For this reason, text preprocessing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages (VIJAYARANI, ILAMATHI, *et al.*, 2015).

The most commonly applied preprocessing operations are illustrated in Table 4: stopwords removal, where irrelevant words (e.g., ‘a’, ‘it’ and ‘to’) are cut out; punctuation and noise removal; upper-to-lower case conversion to ensure that same words will be equivalent in different cases (e.g., ‘Hello’ and ‘hello’); stemming to reduce words to their root form (e.g., ‘processing’ is reduced to ‘process’); tokenization, to separate pieces of text into smaller units called tokens (e.g., ‘Hello word’ is converted to ‘Hello’, ‘word’) (TE, ADHITYA, *et al.*, 2014).

Table 4 – Example of preprocessing operations.

Original sentence ‘A simple# example of Preprocessing Operations’				
Preprocessing operation				
Noise removal	**Stopwords removal**	**Lowercasing**	**Stemming**	**Tokenization**
‘A simple example of Preprocessing Operations’	‘simple example Preprocessing Operations’	‘simple example preprocessing operations’	‘simple example preprocess operation’	‘simple’, ‘example’, ‘preprocess’, ‘operation’

Source: The author (2022).

The resulting tokens are then used to prepare a vocabulary, which refers to the set of unique tokens in the corpus. Tokenization is a key step while modeling textual data, it can be classified into 3 types according to the type of tokens obtained: word, character, and sub-word tokenization (CHOWDHARY, 2020, MORENO, REDONDO, 2016).

2.2.2 Topic Modeling

From the NLP domain, topic modeling consists of several unsupervised approaches to discover the latent semantic structure (i.e., topics) amongst words in a collection of textual documents. Topic modeling can be applied to summarize the main themes of a collection of documents (e.g., accident investigation reports) within a vector space with a reduced number of dimensions or topics since in traditional vector spaces each term or word corresponds to a single dimension (YUN, GEUM, 2020).

Popular topic modeling models include LDA, and its variants, that is a topic modeling algorithm that generates a probabilistic model for a collection of documents (texts), D, to be used as a text summary of a large set of files (EL AKROUCHI, BENBRAHIM, *et al.*, 2021). LDA is based on the assumption that a document is generated by first picking a set of topics, and then for each topic, a set of words is chosen. Thus, each topic is characterized by selecting a very suitable word distribution (SRIVASTAVA, SINGH, *et al.*, 2022).

In summary, LDA uses a predefined number of topics, K, and calculates two probabilities: (i) the probability of words in a specific document, d, assigned to topic τ, and (ii) the probability of topic τ in all the documents for the specific word w. It assumes that the topics and the words in these documents follow a Dirichlet distribution that are used to estimate latent topics (MIN, SONG, *et al.*, 2020). LDA model tries to infer the topics in a set of documents and the joint probability distribution for this model can be expressed as $p(w,\tau) = p(w|\tau) \times p(\tau|d)$ (OSMANI, MOHASEFI, *et al.*, 2020).

Each document is characterized by a topic distribution $\theta_1, \dots ,\theta_D$, while each topic is described by a word distribution $\varphi_1, \dots ,\varphi_K$. Given θ and φ, LDA assumes that the text is generated by the following processs: First, a word-probability distribution, $\varphi_w \sim Dir(\alpha)$, is chosen, where $Dir(\beta)$ is a Dirichlet distribution with parameter β. Second, for each document d, a topic-probability distribution, $\theta_d \sim Dir(\alpha),$ is specified. Then, for each n word w in d a topic assignment, $\tau_{d,n} \sim Multinomial(\theta_w)$, is drawn and a word $w_{d,n}$ is chosen from $p\left(w_{d,n} \middle| \varphi_{\tau_{d,n}}\right)$ (SCHWARZ, 2018).Thus, each topic, or latent dimension, is calculated without any kind of supervision, based only on the distribution of words in the reports (XU, GUO, *et al.*, 2020).

2.2.3 Representation Models

Since raw texts are useless for algorithms that work on numeric feature spaces, they must be converted into a numerical representation, a feature vector that can be used as inputs for supervised and/or unsupervised algorithms. This task is known as language modeling and these representations are the basis for knowledge distillation. Moreover, using mathematical representation for words allow us to perform operations with the resultant vectors, such as estimating ‘semantic similarity’ between words by computing, for instance, the cosine distance between their vectors (BIANCHI, BENGOLEA MONZÓN, *et al.*, 2020). However, obtaining high quality word representations is quite challenging because they should represent the syntax, semantics and context of a word (FELDMAN, SANGER, 2007). Thus, several modeling approaches have been designed from BoW representations to complex neural network language models based, for example, on recurrent neural networks and transformers (WOLF, DEBUT, *et al.*, 2020).

- BoW - BoW model requires two essential pieces of information: (i) a vocabulary of previously known words, and (ii) a measure for the occurrence of the words. Traditional BoW neglects contextual relationships (i.e., information about the order or structure of words), focusing only on the occurrence of words within the document (LI, Teng, MEI, *et al.*, 2011). Specifically, the documents are represented as a vector containing the complete vocabulary size, where all dimensions are null (i.e., 0) except for the ones corresponding to the words in the specific document. For instance, Table 5 shows a BoW representation of three sentences: (a) ‘the worker fell and hit the head on the ladder’, (b) ‘the worker fell from the ladder’, and (c) ‘the ladder fell and hit the worker on the head’. In this example, the vocabulary size is 9, and thus the sentences are 9-dimensional vectors.

Table 5 – BoW representation for three different sentences.

					Vocabulary				
Instances	**the**	**worker**	**fell**	**and**	**hit**	**head**	**on**	**ladder**	**from**
(a)	3	1	1	1	1	1	1	1	0
(b)	2	1	1	0	0	0	0	1	1
(c)	3	1	1	1	1	1	1	1	0

Source: The author (2022).

- TF-IDF - each dimension of the vector representation here corresponds to a word w in the vocabulary. Instead of considering the number of occurrences, TF-IDF takes a weight that increases proportionally to the frequency that the word shows up in a document and decreases proportionally to the number of documents that contain that word. Thus, the greater the frequency of a word in a large number of documents, the less emphasized it is. On the other hand, the less frequent words are more specific and, then, associated with greater weights (HAVRLANT, KREINOVICH, 2017), which are computed as:

$$weight(w, d) = tf(w, d) \times idf(w, D) \tag{1}$$

where $tf(w, d)$ is the number of times term w appears in document d, $idf(w, D) = log\left(\frac{m+1}{df(w)}\right)$, $df(w)$ is the number of documents in the collection D that contain w, and m is the size of D.

Moreover, a traditional task in statistical language modeling is to model the probability that a given word appears next after a given sequence of words. The idea is that there is a probability distribution on word sequences that govern natural language. Thus, a statistical language model can be represented as the conditional probability of the next word given all the previous ones, Equation (2, where w_t is the t-th word (BENGIO, DUCHARME, *et al.*, 2003).

$$\hat{P}(w_1^T) = \prod_{t=1}^{T} \hat{P}(w_t | w_1^{t-1}) \tag{2}$$

A problem with statistical language models is the curse of dimensionality to model the joint distribution between discrete random variables (such as words in a sentence). For instance, to model the joint distribution of 10 consecutive words in a corpus (i.e., set of texts) with a vocabulary size of 100 (i.e., with 100 unique words) there are $100^{10} - 1 = 10^{20} - 1$ free parameters. The introduction of n-gram models alleviate this issue by considering that the conditional probability for the next word depends only on the last $\mathrm{n} - 1$ words Equation (3 (DE MULDER, BETHARD, *et al.*, 2015).

$$\hat{P}(w_t | w_1^{t-1}) \approx \hat{P}(w_t | w_{t-n+1}^{t-1}) \tag{3}$$

However, this approach brings other issues, such as the inability to deal well with synonyms or out-of-vocabulary words that were not present in the training corpus, since the similarity between words is not taking into account and there is much more information in the sequence than just the identity of the previous couple of words (BELLEGARDA, 2004, BENGIO, DUCHARME, *et al.*, 2003).

2.2.4 Advancements in NLP

Progress was made in solving the issues mentioned with the introduction of the neural language model, i.e., language models based on neural networks. Bengio et al. (2003) trained a Feedforward Neural Network (FNN) with sequence of words, showing that a neural model can both learn the probability of a given word appearing next after a given sequence of words and a real-valued vector representation for each word in a predefined vocabulary. In addition to representing a corresponding word, such vector representations encode important linguistic information, e.g., vector representations capture relations like synonymy, antonymy and regional spelling variations.

For instance, Doc2Vec is a model based on neural networks, it introduces the concept of ‘paragraph vectors’, representing devices that retain the main topic of paragraphs. Doc2Vec intuition is that the representations of the document must be good enough at predicting the words or the context of the report. Doc2Vec simultaneously uses two different architectures: Distributed BoW, in which the input is considered as a special vector representing the document and the output is a context word, ignoring the order of the words; and Distributed Memory that introduces an additional document vector for the input along with the word vector representations that are shared among all documents. These vectors are concatenated and, then, the numeric representation of the words and documents are learned during the training process (LAU, BALDWIN, 2016).

Particularly, a continuous representation is derived by utilizing a neural network. Overall, the word vectors of the input sentence are processed in the layers of the architecture. Each layer yields a more abstract representation of the input sentence until a single vector representing the entire input text is obtained. Stated more generally, neural networks can project the vocabulary into hidden layers, so that semantically similar words are clustered. For this reason, neural networks can provide better estimates for words which have never been seen during training (DE MULDER, BETHARD, *et al.*, 2015). Neural networks perform well on several NLP tasks in the absence of any other features. Then, different output layers can be adopted, depending on the task performed by the network. For example, a $sigmoid$ or $softmax$ can be considered for classification or a decoder (sequence-to-sequence configuration) for translation (BAKER, HALLOWELL, *et al.*, 2020, GEORGE K, JOSEPH, 2014, KIM, YOON, *et al.*, 2020).

The main factor that distinguishes different types of networks from each other is their architecture, i.e., how the neurons are connected and the number of layers. Deep Learning (DL) architectures are currently the most popular in NLP research and applications, since they have achieved state-of-the-art results on different NLP tasks (e.g., machine translation, email spam detection, information extraction, and text summarization) (HOWARD, RUDER, 2018).

One critical drawback of the FNNs proposed by Bengio et al. (2003) is that only a fixed number of previous words can be considered to predict the next word. The architecture of FNNs lack any form of ‘memory’; thus, only the words that are presented via the fixed number of input neurons can be used to predict the next word and all words that were presented during earlier iterations are ‘forgotten’. However, NLP is dependent on the order of words or other elements such as phonemes or sentences. The context length was extended to indefinite by using a recurrent version of neural networks, Recurrent Neural Networks (RNN), which can handle arbitrary context lengths.

In RNN-based models the input data must be provided sequentially. In other words, we need to enter the previous state to make any operation in the current state. Sequential neural networks can be used to solve different problems such as:

- Vector-to-sequence: takes a single input, such as an image, and produces a sequence of data, such as a description;
- Sequence-to-vector: takes a sequence as input, such as a product review or a social media post, and outputs a single value, such as a sentiment score;
- Sequence-to-sequence: takes a sequence as input, such as an English sentence, and outputs another sequence, such as the Portuguese translation of the sentence.

However, reading one word at a time, forces RNNs to perform multiple steps to make decisions that depend on words far away from each other. Thus, these architectures are slow to train, and they do not seem to learn long-term dependencies, because of the vanishing gradient problem that occurs due to the multiplicative gradient, which can increase/decrease exponentially according to the number of layers (DE MULDER, BETHARD, *et al.*, 2015).

Transformer architecture was proposed by Vaswani et al. (2017) and has since been replacing RNNs and their variations. Transformer architecture lets go of the recurrence relations used in previous models and, instead, depends entirely on an attention mechanism for modeling dependencies. One of the biggest differences between is that the input sequences are passed in parallel to Transformer architecture. For instance, an input sentence in Portuguese passes through an RNN, one word after another. The hidden state of the current word depends on the hidden state of the previous word; thus, the word embedding is generated one at a time. With the Transformer encoder there is not the concept of timestep, the entire input sequence is provided at once and the word embeddings are determined simultaneously. In general, Transformer architecture abandons the recursion relations and instead relies entirely on an attention mechanism to model global dependencies between input and output. Details about the Transformer architecture are presented in Section 2.3.

2.3 TRANSFORMERS OVERVIEW

Transformers, introduced by Vaswani et al. (2017), was originally designed to solve sequence-to-sequence problems and it was the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence aligned RNNs or convolution. Thus, Transformers made two key contributions: 1) enabled processing entire sequences in parallel, making it possible to scale the speed and capacity of sequential DL models; 2) introduced 'attention mechanisms' that allowed tracking the relations between words across very long text sequences in both forward and reverse directions.

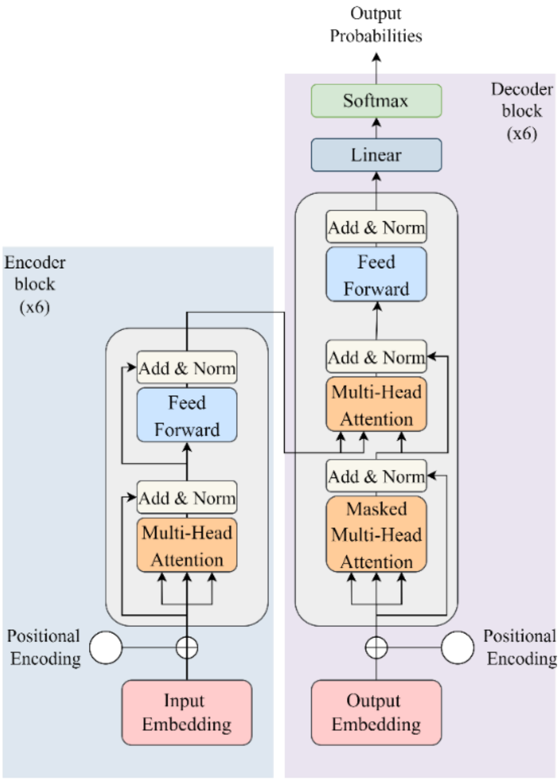
2.3.1 Architecture

Transformers have an encoder-decoder structure as depict in Figure 3 the encoder maps an input sequence of words representations $(x_1, \ldots, x_n)$ to a sequence of continuous representations $z = (z_1, \ldots, z_n)$. Given z, the decoder then generates an output sequence $(y_1, \ldots, y_m)$ of words one element at a time. At each step the model processes the previously generated words as additional input when generating the next (KALYAN, RAJASEKHARAN, *et al.*, 2021). Transformer follows this overall architecture using stacked self-attention and fully connected layers for both the encoder and decoder, shown in the left and right halves of Figure 3, respectively.

The encoder module is composed of a stack of 6 identical layers. Each layer has two sub-layers: 1) a multi-head self-attention mechanism; 2) a fully connected FNN. In addition, there is a residual connection around each of the two sub-layers, followed by layer normalization. In other words, the output of each sub-layer is $LayerNorm(\boldsymbol{x} + Sublayer(\boldsymbol{x}))$, where $Sublayer(\boldsymbol{x})$ is the function implemented by the sub-layer itself. Thus, the input is passed to each encoder block, which processes it through attention and feed forward layers to gradually capture more complicated relationships between the words in the sentence (XU, Y. et al., 2022).

Next, the resulting encoder's attention vector passes through the decoder blocks that translate it into output data (e.g., the translated version of the input text). The decoder module is also composed of a stack of 6 identical layers. In addition to the two sub-layers in each encoder layer, the decoder inserts a third sub-layer, which performs multi-head attention over the output of the encoder stack. Like the encoder, there are residual connections around each of the sub-layers, followed by layer normalization.

Figure 3 – Transformers architecture.



Source: Adapted from Vaswani et al. (2017).

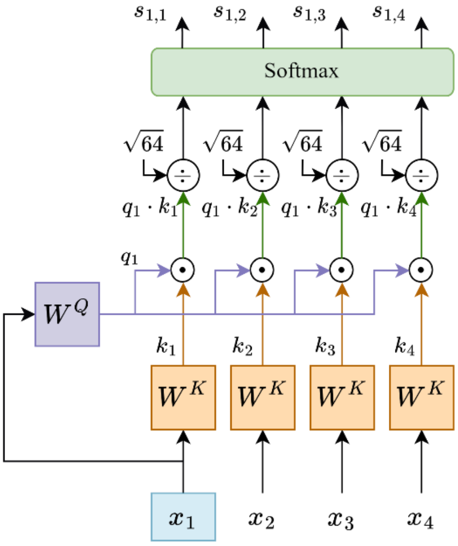
The self-attention sub-layer is modified in the decoder stack to prevent positions from attending to subsequent positions. This masking, combined with fact that the output embeddings are compensated by one position, ensures that the predictions for position i can depend only on the known outputs at positions less than i (VASWANI et al., 2017). The next section provides more details about the attention mechanism.

2.3.2 Attention Mechanism

Transformers architecture adopts self-attention mechanisms to generate word embeddings which take the context of the nearby words. Bahdanau, Cho and Bengio (2015) introduced the idea of ‘attention’ in DL networks, where not only all the input words are considered in the context vector of a RNN, but also the relative importance to each context vector. More specifically, the self-attention mechanism compares every word x_i in the sentence $\boldsymbol{x} = (x_1, \dots, x_n)$ to every other word in the sentence $\boldsymbol{x}$ and, then, combines contextually related words together.

The first step of the self-attention mechanism involves computing the query, key and value vectors for each word x_i in the input sentence, q_i, k_i and v_i respectively. These vectors are by multiplying the embeddings by matrices W^Q, W^K and W^V. Next, we take the dot products of the query with all keys in the sentence to find contextually related words for a chosen word. The results of the dot products are used as weight factors, that indicates how much two words (x_i and x_j) depend on each other. After that, the dot products are scaled with $\sqrt{64}$ (square root of the dimension of the key vectors) and passes through a softmax function to map the values to [0,1] and to ensure that they sum to 1 over the whole sequence, resulting in scaled weight factors for word i ($s_{i,j}$). Figure 4 shows this step for the first word of the sentence x_1 (OK, LEE, *et al.*, 2022).

Figure 4 – Computing the scaled weight factors for the first word x_1 of an input sentence $\boldsymbol{x} = (x_1, x_2, x_3, x_4)$.

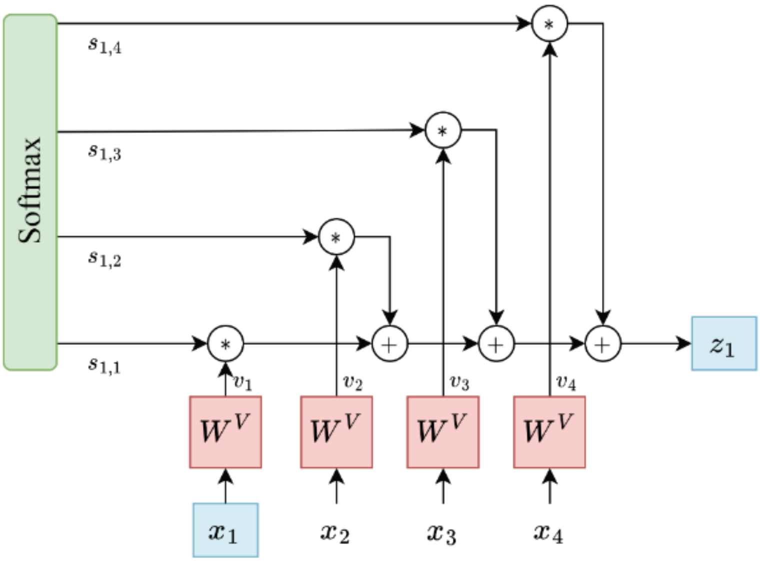


Source: adapted from Sutskever; Vinyals; Le (2014); Vaswani et al. (2017); Wu, Y. et al. (2016).

After computing the scaled weight factors for all the words in the sentence $s_{i,j}\ \forall\ i, j \in \boldsymbol{x} = (x_1, \dots, x_n)$, each value vector is multiplied by its corresponding scaled weight factor. Then, the weighted value vectors are summed resulting on the output of the self-attention layer, i.e., the enriched word embedding z_i for the word x_i. Figure 5 illustrates the computation for the first word of a sentence.

As shown in Figure 3, the attention mechanism is applied as self-attention in the encoder and decoder blocks and as encoder-decoder attention. The attention scores in the encoder-decoder attention are computed as described before, but the queries vectors come from the previous decoder layer, and the keys and values come from the output of the encoder. This allows the model to obtain word representations that captures the relation between each target word with each input word.

Figure 5 – Sum of the weighted vectors with the softmax layer outputs for the word x_1 resulting in new embeddings



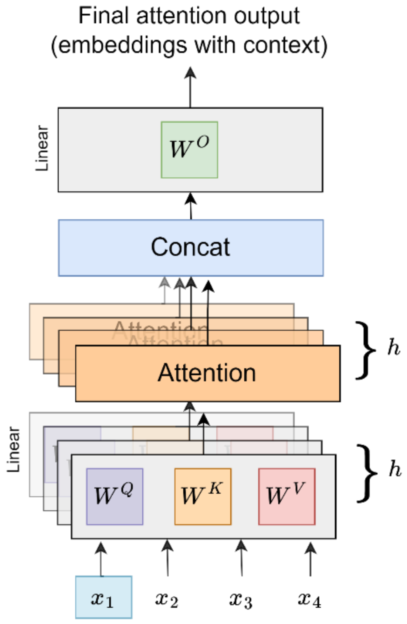
Source: adapted from Sutskever; Vinyals; Le (2014); Vaswani et al. (2017); Wu, Y. et al. (2016).

Moreover, in transformers, the attention function with the queries, keys and values vectors, are linearly projected h times in parallel. The attention sublayer splits its query, key, and value vectors across h heads and each head process the data independently. All the attention outputs are concatenated, and the final embedding is obtained by multiplying the final attention output by matrix W^O (Figure 6).

Transformers based models, have been successfully applied in NLP due to their ability to learn universal language representations, when trained with large volume of text data. However, training these models from scratch is computational costly and time-consuming; thus, transfer learning allows the reuse of the knowledge learned in source tasks (i.e., pretraining tasks) to perform downstream tasks target tasks. Another possible issue to learn valuable language information is the difficulty to obtained labeled dataset to perform NLP tasks. However, self-supervised learning allows transformers models to learn based on the pseudo supervision provided by one or more pretraining tasks (KALYAN, RAJASEKHARAN, *et al.*, 2021).

BERT (DEVLIN, CHANG, *et al.*, 2018) and GPT (RADFORD, NARASIMHAN, *et al.*, 2018) were the first pretrained language models based on transformers encoder and decoder respectively. Firstly, GPT achieved state-of-the-art results on 9 NLP tasks. BERT obtained new state-of-the-art results on 11 NLP tasks. Currently, several authors have been exploring and modifying BERT and GPT to derive models with better performance, such as XLNet (YANG, DAI, *et al.*, 2019), RoBERTa (LIU, Yinhan, OTT, *et al.*, 2019), ELECTRA (CLARK, LUONG, *et al.*, 2020), ALBERT (LAN, CHEN, *et al.*, 2019), Gato (REED, ZOLNA, *et al.*, 2022) and Chinchilla (HOFFMANN, BORGEAUD, *et al.*, 2022). The proposed modifications consist mainly in increasing the number of parameters (e.g., number of layers), increasing the number of training data, and training the models in different unsupervised source tasks (GAO, Leo, BIDERMAN, *et al.*, 2020).

Figure 6 – Computing multi-head self-attention for the word x_1 resulting in new embeddings



Source: adapted from Vaswani et al. (2017).

In the context of risk and reliability analysis, it is still very common to find simpler models such as BoW, TF-IDF and Doc2Vec applied, despite the advances when considering studies related to core NLP tasks. BERT implementations in Pytorch and Tensorflow have been available for more than a year (WOLF, DEBUT, *et al.*, 2020), in different languages, stably, with no long-term compatibility problems between libraries.

Moreover, the inference and fine-tuning of models with more parameters, such as the ones mentioned in the previous paragraph, has a higher computational cost. Taking these factors into consideration, we developed our methodology based on BERT, as it gives us flexibility and robustness.

2.4 BERT

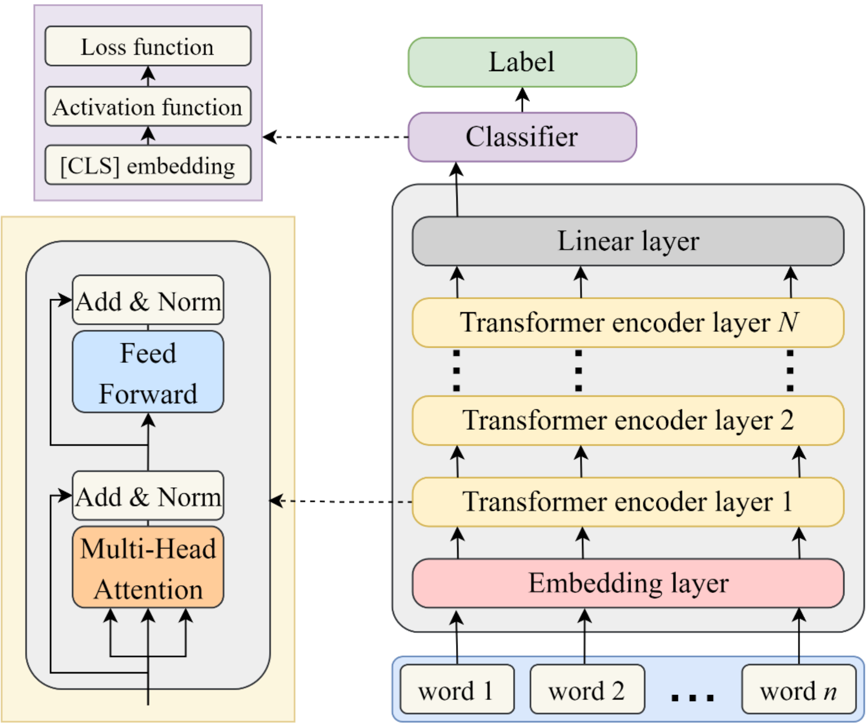
BERT is a novel method of pre-training language representations, and its architecture consists of a stack of N, in the original paper $N = 12$, Transformer encoders. BERT was proposed and pre-trained by Google on an extremely large corpus (BooksCorpus (ZHU, KIROS, *et al.*, 2015) and Wikipedia). During pre-training, the model learns the relation between words within a sentence and between sentences by training on two unsupervised tasks: ‘mask language modeling’ and ‘next sentence prediction’. For the ‘masked language modeling’, BERT takes in a sentence with 15% of the words being randomly masked originally; then, the objective is to predict the original word of the masked token based only on its context. This objective allows the representation to combine the right and the left contexts. For the ‘next sentence prediction’, BERT takes in two sentences and the aim is to predict if the second one follows the first. This task allows the model to understand the relation across sentences (DEVLIN, CHANG, *et al.*, 2018).

In a nutshell, during pre-training, a set of two sentences, with some words being masked, is fed to the model and each word is converted into feature vector representations. These input data are passed to the transformer encoder layers; each encoder is broken down into two sub-layers: a multi-head attention and a feed forward neural network. The encoder’s inputs first flow through the attention layers that help the model focus on other words in the input sentence as it encodes a specific word. Next, the outputs of the multi-head attention layer are concatenated and fed to a neural network. Finally, each word vector is passed into a fully connected layer in the output layer; for more details see Devlin et al. (2018) and Vaswani et al. (2017).

Nevertheless, training these models from scratch would require large datasets and a long time to converge. Thus, pre-trained word representations have been the key component for improving different NLP tasks. Several pre-trained versions of the model are available for download; thus, we can further train BERT to perform a supervised-learning task by adding an untrained layer of neurons on top of the pre-trained model. Overall, during fine-tuning, the pre-trained parameters are adopted to initialize the model and, then, they are fine-tuned using specific labelled data for solving the supervised task.

To fine-tune pre-trained BERT, we need to adjust its last layer according to the supervised task of interest. For instance, we can adjust BERT's architecture by adding one output layer on top of the pre-trained model to adapt it for performing a classification task (Figure 7).

Figure 7 – Modified BERT architecture to perform classification task.



Source: adapted from (DEVLIN et al., 2018).

The representation of the last token [CLS] of the input sentence is fed to the output layer, i.e., the final hidden state $\boldsymbol{h}$ of the token [CLS] is used to represent the sentence. Then, an activation function computes $\boldsymbol{h}$ and converts it into probabilities Equation (4.

$$p(c|\boldsymbol{h}) = activation(\boldsymbol{h}^T \boldsymbol{W} + b) \tag{4}$$

where c is the class of the input sentence, b is the bias, and $\boldsymbol{W}$ is the weights matrix of the added output layer.

In this case, the loss is propagated through the entire architecture and all BERT pretrained parameters as well as $\boldsymbol{W}$ and b are fine-tuned and updated based on the new dataset. However, $\boldsymbol{W}$ and b are the only parameters that need to be randomly initialized and learned from scratch. Although the developed models are initialized with the same pre-trained parameters, training for distinct tasks provides different fine-tuned models in an efficient way. Indeed, this approach allows us to build models with state-of-the-art architectures within a reasonable time, since training these architectures from scratch can take days (HOWARD, RUDER, 2018).

3 RELATED WORKS

Zio (2018) pointed that the advance in computing power and growing data availability counts in favor for the development of models for mining of knowledge acquired for RA. Indeed, automatic mining patterns from massive amount of textual data is attractive as the text is a source of knowledge to support safety professionals and to improve the spread of safety-related culture. Indeed, TM and NLP techniques aim to understand, process, and interpret human language allowing to train intelligent models (SINGLE, SCHMIDT, *et al.*, 2020), which provides a rapid and trustworthy analysis of large, textual databases comprised of risk study documents, such as accident investigation reports and/or PrHA (BALLESTEROS, SUMNER, *et al.*, 2020). ML and NLP have been successfully applied to different research areas. In this section, we focus on the different issues that both fields can handle in the risk analysis context.

For instance, Rachman and Ratnayake (2019) developed an ML based-model to conduct risk-based inspection screening assessment that is used to identify equipment that makes major contribution to the system's total risk of failure, thus, allowing to prioritize high-risk systems. These authors have to perform feature selection to build a dataset from previous risk-based inspections conducted for offshore oil and gas production and processing units, where the output was the risk category. Kurian et al., (2020) applied ML to analyze process and occupational-type incidents reports from five oil sand industries. These reports were manually classified and provided to different ML algorithms (AdaBoost, Decision Trees, K-Nearest Neighbors, Random Forest, Support Vector Machines (SVM), Multilayer Perceptron, Multinomial Naive Bayes, and Logistic Regression) in order to predict labels for incident type, consequence type, actual risk score, and potential risk score. Moreover, we developed in (MACÊDO, MOURA, RAMOS, *et al.*, 2022) a ML-based methodology to identify potential consequences related to the operation of an atmospheric distillation unit and to classify the expected frequency and severity of consequences. To that end, we compared the performance of different ML algorithms (SVM, Multilayer Perceptron, AdaBoost, Random Forest, K-Nearest Neighbors, and Gradient Boosting Decision Trees). However, these approaches require the feature engineering step, where the specialist must manually process the database and pick up which features will be used for feeding the model.

TM and NLP has been adopted to infer information from accident investigation reports of aviation (ANDRZEJCZAK, KARWOWSKI, *et al.*, 2014), civil construction (BAKER, HALLOWELL, *et al.*, 2020), road traffic (MUGURO, SASAKI, *et al.*, 2020), railway industry (HUGHES, SHIPP, *et al.*, 2018), and oil industry (AHMADPOUR-GESHLAGI, GILLANI, *et al.*, 2020). In addition, NLP can be found in the analysis of reports of accidents that almost occurred (ANSALDI, SIMEONI, *et al.*, 2020), verification of safety rules (GULIJK, HOLMES, 2020), and automated classification of injury (GUARAV, KIRSTEN, *et al.*, 2020) and injury leave (MAIOR, SANTANA, *et al.*, 2020) from accident investigation reports. For example, Sarkar et al. (2019) combined categorical features and text data to predict the accident outcomes such as injury, near miss, and property damage using topic modeling and ML. Madeira et al. (2021) performs text classification to categorize human factors from aviation incident reports using semi-supervised and supervised models. It is worth noting that these authors focused on the model's performance to automatically identify patterns and classify causes since this is paramount for accident investigation.

Moreover, Heidarysafa et al. (2018), Nayak et al. (2009) and Zhang et al. (2019) applied NLP methods on accident's narratives to comprehend contextual relationships inherent to road accidents, identify causes and predict secondary crashes. Passmore et al. (2018) applied topic modeling to summarize narrative reports of injuries that occurred in coal mines to identify the main theme of the documents. Boggs, Wali and Khattak (2020) applied NLP to perform an exploratory analysis of automated vehicle crash reports to quantify the pre-crashes and location factors. Kuhn (2019), Robinson (2019) and Sjöblom (2014) developed approaches based on NLP in the context of aviation accident analysis to find similarities between the accident investigation reports applying clustering, and to automatically identify topics within reports by topic modeling in order to pinpoint trends to prioritize safety activities (VAYANSKY, KUMAR, 2020).

Liu et al. (2021) employed K-means clustering to assess incident narratives from Pipeline and Hazardous Materials Safety Administration database in order to identify contributing factors and latent causality. Suh (2021) applied TM and topic modeling to extract topics from narrative texts contained in accidents reports of Occupational Safety and Health Administration database to identify sectoral patterns, which is defined by categorizing the common nature of accidents shared across industries. Wang and Mai (2016) proposed a system based on TM to extract risk elements from risk matrices. The authors extracted the risk origin, component, causes, probability, and severity from those documents and annotated the identified words. Then, they adopted SVM to perform a binary classification (high or low risk).

Sarkar (2016) proposed a TM and ML-based model to identify basic events that influence the primary causes of occupational accidents (reactive data) in a steel plant. Then, the author predicted the probability of the occurrence of a given cause through a Bayesian network (LEU, CHANG, 2013). Next, Sarkar, Verma and Maiti (2018) included proactive data, which consist of observations by safety inspectors that indicate a certain level of potential hazard, and used decision tree classifiers to predict the occurrence of accidents.

Singh, Maiti and Dalmahapatra (2019) used reactive (accident report) and proactive data (workplace conditions during an accident-free period) to identify chain of events in accident paths. Zhang and Mahadevan (2019) developed an SVM and DL-based model to examine previous accidents. The authors applied the ensemble model to extract features from reports of aviation accidents and assign the risk level to a corresponding aviation incident. Their model classified the risk associated with the consequences of accidental events as high, moderately high, medium, moderately medium, and low.

In order to support the development and periodic review of QRA, several studies have proposed advanced approaches to address different challenges usually faced in the analysis. For example, Bernechea, Vílchez and Arnaldos (2013) proposed a methodology to consider domino effects into QRA, by estimating the frequency with which new accidents will occur, while Kamil et al. (2019); Lisi et al. (2015); Zhou and Reniers (2018) focused on modeling such effects. Other studies have been done to estimate more realistic accident frequency (BADRI, NOURAI, *et al.*, 2013, LANDUCCI, PALTRINIERI, 2016) and to quantify and/or update probability of failure/ accidents (GUO, JI, *et al.*, 2021, LI, Yang, WANG, *et al.*, 2021, MENG, ZHU, *et al.*, 2021, SARVESTANI, AHMADI, *et al.*, 2021). However, these studies were not concerned with the efforts required in the early stages of QRA. Indeed, they adopted traditional and time-consuming approaches to identify hazards – namely, PrHA, HAZOP, bow-tie (constructed after assessing accident databases), assessment of historical data, literature review, and others.

Generally, these techniques involve examining different engineering documents that describe the installation (e.g., flowcharts, equipment, and material lists) and attending numerous meetings to postulate possible leakages, identify hazards and their possible causes and consequences and, finally, evaluate and classify risks (CARRASQUILLA, MELKO, 2017).

Regarding the initial steps of QRA, for instance, Aziz, Ahmed and Khan (2019) built an ontology for knowledge modeling and to design an expert database system from the hazard scenarios. The preliminary step of the proposed approach consists of outlining the hazard scenarios and gathering relevant information, which requires tedious procedures and several brainstorming sessions. Furthermore, Ahmad et al. (2019) incorporated the thematic analysis for hazard prevention strategies, which was applied to extract information from accident databases; this analysis involves transcribing and/or re-reading the data repeatedly to obtain the key points of the hazard prevention suggestions from the accident databases.

Although the mentioned studies significantly contributed to the advance of RA, there is still a lack of studies applying TM, NLP and DL techniques to support preventive risk studies in process industries. Most studies that apply these techniques TM, NLP and DL to RA focus mainly on automatic identification of patterns from reactive data, such as accident reports, and classification of accidents' cause. Indeed, one can look into reports both quantitatively, producing statistics and trends, and qualitatively, where prevention strategies can be drawn up based on different causes. However, real investigation reports oftentimes present drawbacks related to the quality of information filled out. For example, incomplete or missing data, lack of standardization, and conflicting information are commonly found in these reports. In practice, these issues are a significant hurdle for analysts to extract useful insights, and then propose effective preventive measures to improve safety. Moreover, valuable information is stored in different forms of text data, especially proactive data, and their adoption could support preventive risk studies. Therefore, this thesis aims to fill the mentioned gaps by:

Applying NLP techniques to recognize, address, and point out possible inconsistencies in the accident investigation reports. Here, the database is composed of written reports about previous accidental events that occurred in a real hydroelectric power company. Using a 6-year historical database, the proposed methodology investigates and discusses the current characterization of the accident investigation reports, which were structured based on the Brazilian Standard ABNT NBR 14280 - Workplace accident record. To the best of our knowledge, this is the first study proposing a methodology for identifying low-quality patterns in investigation reports through text mining and NLP and multiclassification tasks considering hydroelectric power company context.

Proposing an approach based on TM, NLP and DL models to extract information from proactive risk analysis of an oil refinery in order to specify the potential consequences and classify each accidental scenario in terms of severity of their consequence and likelihood of occurrence. To that end, we developed an approach based on pre-trained BERT model to extract relevant information from PrHA documents. The text data was used to feed and fine-tune pre-trained BERT. In this way, the model is capable of learning patterns that allow it to characterize risk scenarios (predict potential consequences, severity of consequences and likelihood of occurrence) given the occurrence of an uncontrollable release of hazardous material. Next, we present our proposed approach.

4 METHODOLOGY TO ASSESS ACCIDENT INVESTIGATION REPORTS OF A HYDROPOWER COMPANY

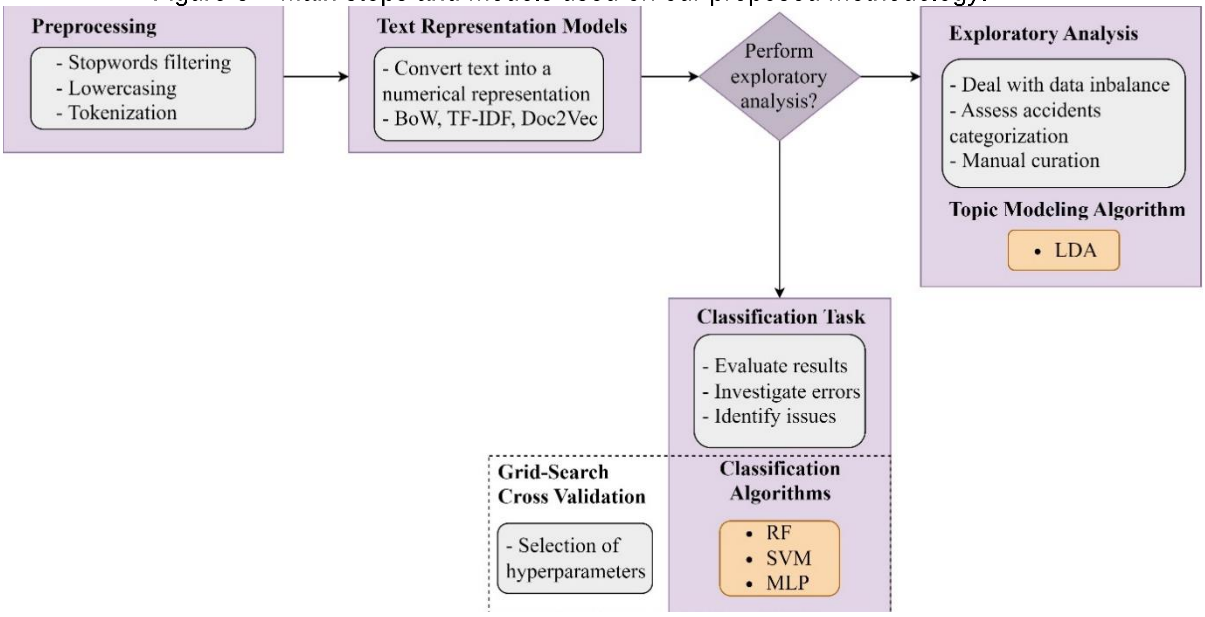
Part of this chapter have been have published in International Journal of Occupational Safety and Ergonomics (MACEDO, July B. et al., 2022). This part of the study aims to use NLP techniques to recognize, address, and point out possible inconsistencies in the accident investigation reports. Here, the database is composed of written reports about previous accidental events that occurred in a real hydropower company. Using a 6-year historical database, the proposed methodology investigates and discusses the current characterization of the accident investigation reports, which were structured based on the Brazilian Standard ABNT NBR 14280 - Workplace accident record (NBR, 2001). To the best of our knowledge, this is the first study proposing a methodology for identifying low-quality patterns in investigation reports through NLP and multiclassification tasks considering hydroelectric power company context.

4.1 METHODS

A schematic overview of our proposed methodology to extract knowledge from text is shown in Figure 8. The main idea is to support the diagnosis of the quality content and understanding of raw texts of accident investigation reports using NLP and ML methods. This can allow experts to identify inconsistencies and/or propose corrections and/or changes to improve the reports. After implementing the suggestions, the resulting reports would provide more useful and reliable information for the safety technicians. Although we processed texts in Portuguese, we followed generic steps.

Firstly, raw reports are pre-processed using operations to remove noise, then are converted into feature vectors through different text representation models. Next, we performed an exploratory analysis, using the resulting representations to summarize their main contents and identify patterns, searching for useful information to perform a further assessment. Then, we trained several classifiers using different combinations of ML algorithms and feature vector representations to categorize the reports into different groups. Finally, we evaluated their predictions and compared their performance. Each step was developed in Python computational language. The steps are detailed in the following sections.

Figure 8 – Main steps and models used on our proposed methodology.


Preprocessing
- Stopwords filtering
- Lowercasing
- Tokenization
Text Representation Models
- Convert text into a numerical representation
- BoW, TF-IDF, Doc2Vec
Perform exploratory analysis?
Exploratory Analysis
- Deal with data inbalance
- Assess accidents categorization
- Manual curation
Topic Modeling Algorithm
• LDA
Classification Task
- Evaluate results
- Investigate errors
- Identify issues
Grid-Search Cross Validation
- Selection of hyperparameters
Classification Algorithms
• RF
• SVM
• MLP

Source: The author (2022).

4.1.1. Text Preprocessing

We performed three preprocessing operations: (i) stop words filtering, (ii) lowercasing, and (iii) tokenization. Stop words filtering is used to identify the content information, in which noninformative terms are removed (e.g., ‘the’, ‘it’, and ‘is’). Terms are also converted to lowercase (uppercase conversion). Finally, tokenization is applied by dividing the text into terms in which their unit is defined as words (also known as tokens). For all these preprocessing steps, we used Python string methods, as well as functions from the NLTK Library (BIRD, KLEIN, *et al.*, 2009).

4.1.2. Modeling Process

As mentioned, computers do not understand words, only numbers; thus, instead of directly providing text to algorithms, we convert it into feature vector representations. Here, we analyzed and compared the results by using three different models: BoW, TF-IDF, and Doc2Vec, using the Gensim library (REHUREK, SOJKA, 2010).

4.1.3. Exploratory Analysis

Firstly, we performed Data Augmentation (DA) as an attempt to overcome the database imbalance. In NLP context, DA is a natural choice to replace words or phrases with their synonyms (WEI, ZOU, 2020). We used the nlpgaug (MA, 2021), which is a library dedicated to textual augmentation in ML experiments. Simply put, the replacement of some words by synonyms, such as ‘machinery’ for ‘engine’, preserves the same content but generates a new sentence. More specifically, we used word.synonym function for the Portuguese language, which performs word replacement from the large lexical database WordNet.

Thus, the synonym substitution procedure was performed on the training data generating new samples for categories with fewer instances and creating a more balanced database. To train and test the classifiers, we split each category in an 80/20 ratio for training/testing.

Moreover, in this step we used topic modeling to summarize the main themes of a collection of documents (e.g., reports) within a vector space with a reduced number of dimensions or topics since in traditional vector spaces each term or word corresponds to a single dimension (YUN, GEUM, 2020). We sought the best number of topics by using the coherence score within a pre-defined search space. Topic coherence computes a value for a single topic by measuring the degree of semantic similarity between high-scoring words in the topic, summing pairwise distributional similarity scores over all words (W). We here consider the UMass metric (ASNANI, PAWAR, 2018) Equation (5, which was designed for LDA; it uses the conditional log-probability smoothed by adding ε; $D(w_i, w_j)$ is the number of documents containing the words w_i and w_j, and $D(w_j)$ is the number of documents containing w_j:

$$coherence(W) = \sum_{w_i, w_j \in W} \frac{\log\left(D\left(w_i, w_j\right) + \varepsilon\right)}{D\left(w_j\right)} \tag{5}$$

4.1.4. Classification Task

In the occupational safety context, NLP is frequently used to build predictive systems by using a corpus of already coded accident descriptions to learn the relationship between the terms in the narratives and a target category (PIMM, RAYNAL, *et al.*, 2014). Thus, to perform the classification task we here consider three well-known ML techniques implemented in Python language with the scikit-learn package (PEDREGOSA, VAROQUAUX, *et al.*, 2011):

- SVM - Support Vector Machine: it is based on the inductive principle of structural risk minimization, which aims to minimize an upper limit of the generalization error, considering the sum of the training error and a loss function (MAIOR, MOURA, *et al.*, 2019). SVM maps nonlinearly separable data in a small-sized space to a large-sized feature space, where the data can be linearly separated (VAPNIK, IZMAILOV, 2019).
- MLP - Multilayer Perceptron: class of artificial, feedforward neural network, which presents at least three layers of neurons: an input layer, a hidden layer, and an output layer (MAIOR, MOURA, *et al.*, 2018). Each hidden node includes a non-linear activation function; MLP uses a supervised learning technique for training that is called backpropagation (WANG, Shi-Wei, YU, 2005).
- RF - Random Forest: it is an ensemble method that extends the idea of recursive partitioning, where it cultivates several decision trees. Each tree is trained using a sample of the training data set and, then, grows using a subset of predictors randomly selected from each node. After generating a large number of trees, they vote for the most popular class; more details can be found in (BREIMAN, 2001).

Here, the classifiers were trained over 80% of the dataset. Then, the performance of the classifiers was evaluated on test data (the remainder 20%) through the accuracy, A, as seen in Equation (**6**:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

where TP, TN, FP and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. Additionally, we computed precision, recall and F_1-score, where precision is the number of true positives (instances correctly predicted as 1) overall positive predictions (all instances predicted as 1), recall is the number of true positives over all instances with 1 as the true label, and F_1-score is the harmonic mean between precision and recall.

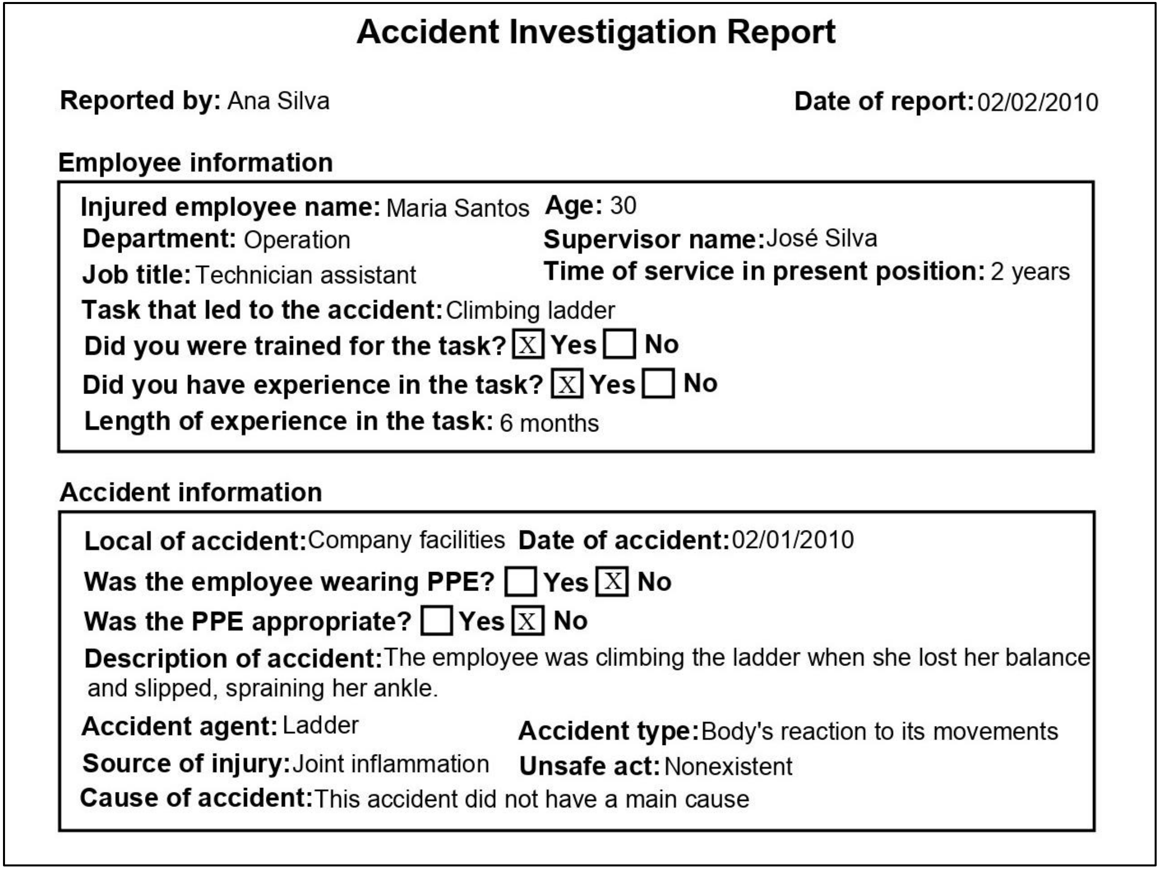
4.2 DATA ANALYSIS AND BASELINE RESULTS

The database we analyzed is presented in the form of a spreadsheet, where rows correspond to an accident investigation report and columns are characteristics (factors) about the event itself (e.g., location, causes, damage to people, injury leave (days), financial impact) and employee involved in the event (e.g., job position, experience in the activity, training). The dataset contains 626 reports that describe, in Portuguese, accidental events that occurred in a 6-year period. Each report contains several factors; thus, the safety technician must fill out almost 60 fields with either a 'Yes' or 'No' answer or provide a short text. Figure 9 shows a simplified version, with fictitious employee personal information, of the report to illustrate its structure and how the main information is provided.

The high number of factors may lead one to believe the event descriptions are thoroughly detailed. However, the filling of the long report makes the process of investigating, recording, and documenting the accidental event boresome. In fact, difficulties in filling out the reports are a common problem already mentioned by the safety technician. Moreover, the lack of standardization for describing the factors may hinder the efficient use of the accident investigation reports database for supporting decision-making for risk management. Indeed, as the report's descriptions are based on NBR14280 (NBR, 2001), it makes one wonder if the factors are clearly understood and well-presented by the safety technician.

After scrutinizing the raw data, we identified that much information is not filled out, presenting a poor standardization for completing the reports, which is probably related to a lack of understanding about the fields and/or questions. This is even more critical when factors need to be described as free text in open fields, which is the case of 'accident type', 'accident agent', 'source of injury', 'unsafe act', and others. For instance, based on NBR 14280 (NBR, 2001), the 'accident agent' factor represents the object, substance, or environment which the unsafe condition is related to and that caused the accident, whereas 'source of injury' corresponds to the object, substance, energy, or movement of the body that directly caused the injury. Their meanings are slightly different and may be easily mistaken. Thus, the safety technician needs to deeply understand the factors' concept to correctly fill out the reports and have them memorized (or check the standard); otherwise, an incorrect description is going to be provided.

Figure 9 – Simplified version of the report, filled with fictitious employee information and an example accident.



Accident Investigation Report

Reported by: Ana Silva **Date of report:** 02/02/2010

Employee information

Injured employee name: Maria Santos **Age:** 30
Department: Operation **Supervisor name:** José Silva
Job title: Technician assistant **Time of service in present position:** 2 years
Task that led to the accident: Climbing ladder
Did you were trained for the task? ☒ **Yes** □ **No**
Did you have experience in the task? ☒ **Yes** □ **No**
Length of experience in the task: 6 months

Accident information

Local of accident: Company facilities **Date of accident:** 02/01/2010
Was the employee wearing PPE? □ **Yes** ☒ **No**
Was the PPE appropriate? □ **Yes** ☒ **No**
Description of accident: The employee was climbing the ladder when she lost her balance and slipped, spraining her ankle.
Accident agent: Ladder **Accident type:** Body's reaction to its movements
Source of injury: Joint inflammation **Unsafe act:** Nonexistent
Cause of accident: This accident did not have a main cause

Source: The author (2022).

Figure 9 exemplifies a common situation found in the database, where the 'source of injury' factor was filled in as 'joint inflammation', which is in fact the 'accident consequence'. According to NBR 14280 (NBR, 2001), in this case, the correct 'source of injury' factor would be 'ladder'; thus, the 'accident agent' factor, which was correctly filled out in Figure 9, is categorized as 'building, structure, pole, tower, rope, cable, electrical cable, chair, drums, pulleys, tanks, cylinders, tank protection'. Yet, in the analyzed database, there are several redundant, but required fields that could be grouped together to make the report simpler, more straightforward. For example, the fields 'did you have experience in the task?' and 'how much time of experience in the task?' could be merged, because if the employee does not have experience in the task, the time of experience would be zero. Moreover, the fields 'was the employee wearing

PPE?' and 'was the PPE appropriate' could be simplified to 'was the employee wearing appropriate PPE?', where PPE stands for Personal Protective Equipment. Such simplifications could make the filling out process less exhausting, which would increase the quality of reports.

Here, we focus our analysis on the 'accident agent' as it represents a valuable source of information to identify common elements about the cause of accidents and, then, propose preventive measures. The safety technician standardized the 'accident agent' categories into nine different classes, which are presented in Table 6.

Table 6 – Accident agent' categories.

Label	Category
0	Scaffolding
1	Duct, ditches, pipes, tunnels, pressure vessels
2	Building, structure, pole, tower, rope, cable, electrical cable, chair, drums, pulleys, tanks, cylinders, tank protection
3	Manual and automatic tools, drilling machines, sander, polisher, grinder, drill, lathe, electrical discharge machine, electrical equipment, electric arc, hydraulic or pneumatic
4	Engine, pump, turbine
5	Trip or slip
6	Chemical substance and industrialized metal, lead, mercury, zinc, cadmium, chromium, rebar, ferrous alloy
7	Commuting accidents
8	Motor vehicle, motorcycle, tractor scooter, on track, hoisting equipment

Source: The author (2022).

However, we identified that, no matter the background and expertise, the safety technician was often confused by the categories when reporting the accidental event. To depict this situation, Table 7 presents actual cases of the 'accident agent' category we found in the database. In Table 7, one can see that the 'accident agent' of the first instance should be categorized as 'trip or slip' (category 5), because the employee tripped over a box. In addition, the second instance represents an accident that occurred when the employee was arriving at the company's facilities, i.e., a 'commuting accident' (category 7), but it was misclassified.

Moreover, the third instance should be classified as category 2, since the accident consisted of the employee slipping while climbing the structure. This may indicate that these categories are not well-understood. In the context of this database, the accident investigation reports were also analyzed by few other studies in statistical context and in text mining field. For instance, Moura et al. (2016) proposed a Bayesian population variability method for estimating accident and recovery rate distributions, while Maior et al. (2020) performed binary text classification (accidents with or without injury leave) based on the description written by the safety technician. Guimarães et al., (2020) grouped the same accident investigation reports, considering the safety technician description, and indicated five as the ideal number of clusters, pointing out that it may exist a smaller number of groups of accidents that are more generic and easier to distinguish.

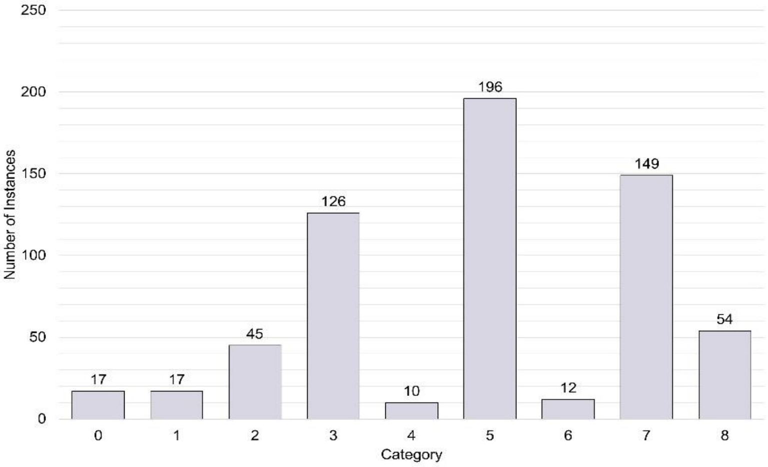
Table 7 – Example of instances assigned with the wrong 'accident agent' category.

Instance	Accident description	Accident agent
1	*When the employee stood up from her chair, she hit her foot on a box that was under the worktable. She lost her balance, falling and hurting her thumb.*	Duct, ditches, pipes, tunnels, pressure vessels
2	*The employee, when arrived at the sidewalk that gives access to the company's facilities, bumped into a stone and fell, suffering injuries to her leg and hand*	Building, structure, pole, tower, rope, cable, electrical cable, chair, drums, pulleys, tanks, cylinders, tank protection
3	*When climbing on the structure, the employee slipped, and the impact dislocated his shoulder.*	Manual and automatic tools, drilling machines, sander, polisher, grinder, drill, lathe, electrical discharge machine, electrical equipment, electric arc, hydraulic or pneumatic

Source: The author (2022).

Conversely to these aforementioned papers, we here perform a multiclassification task, since we categorized the accidents agents. As previously mentioned, the original database has nine categories of 'accident agents', and the number of instances in each is presented in Figure 10.

Figure 10 – Number of instances per 'accident agent' factor considering the original categories.



Source: The author (2022).

It is noteworthy that there are many more descriptions in categories 3 (‘Manual and automatic tools, drilling machines, sander, polisher, grinder, drill, lathe, electrical discharge machine, electrical equipment, electric arc, hydraulic or pneumatic’), 5 (‘trip or slip’), and 7 (‘commuting accidents’) than to the others.

Initially, to get a base result to compare with the modifications that will be proposed, we considered these classes to train the classification models without performing the exploratory analysis (see Figure 8). In order to avoid biases and to account for the variability, the process of training and test was repeated 10 times (CV – Cross Validation) and we computed the median performance metrics (Table 8) to classify the test data.

Table 8 – Median accuracy (%) of the classification task using nine ‘accident agent’ categories as the target.

Classifier	BoW	TF-IDF	Doc2Vec
SVM	51.98	53.96	31.34
RF	58.33	57.14	31.34
MLP	59.92	59.52	28.17

Source: The author (2022).

As we expected, the performance of the models is quite low and further investigation is required. We assume a few hypotheses that may have caused this: (i) the data imbalance (Figure 10), (ii) the low quality of data/grouping, and/or (iii) poor configuration of ML algorithms. Then, we will work on these assumptions in next the sections.

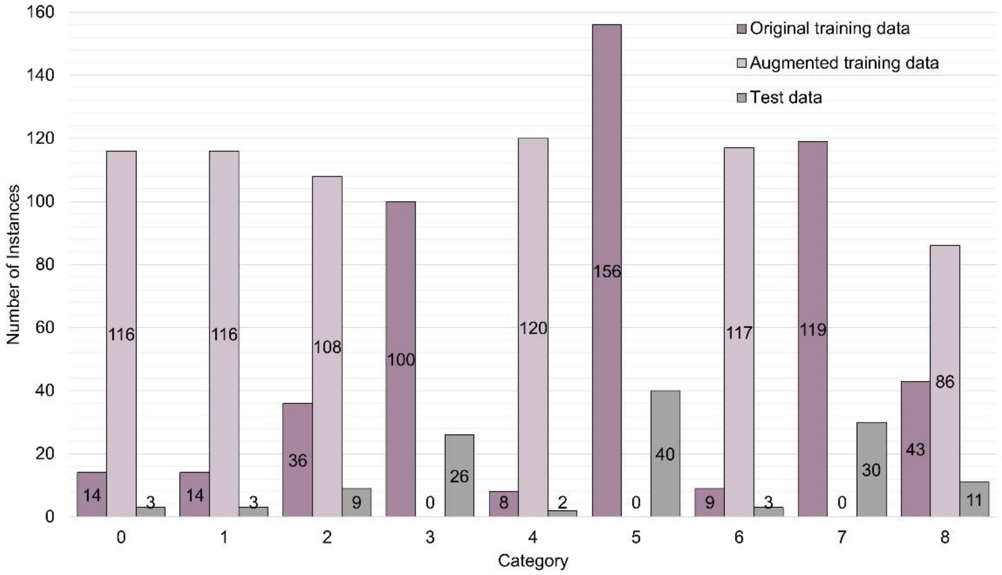
4.3 APPLICATION AND RESULTS

4.3.1 Dealing with Data Imbalance

The synonym substitution procedure was performed on the training data generating new samples for categories with fewer instances and creating a more balanced database. To train and test the classifiers, we split each category in an 80/20 ratio for training/testing. Therefore, the training set for categories 3, 5, and 7 had 100 samples, while the remaining categories had less than 45 samples (e.g., there were only 8 training samples for category 4). Hence, the word.synonym was used for categories 0, 1, 2, 4, 6 and 8 to randomly select and generate new training samples.

Note that we did not generate a database with an equal amount of data for all categories, we only ensured that the word.synonym provides at least 100 training samples, including the original training data. Figure 11 presents the number of instances in each category on the augmented training set and on the test portion.

Figure 11 – Number of training (original and augmented) and test instances per ‘accident agent’ factor considering the original categories found in the reports.



Source: The author (2022).

Next, the same steps of processing the augmented data, training, and testing were repeated 10 times (CV) to account for the variability. Nevertheless, as one can see in Table 9, even with a more balanced dataset, all classifiers showed an inferior median accuracy compared to the baseline performance (Table 8).

Table 9 – Median accuracy (%) of the classification task using nine categories as the target after DA.

Classifier	BoW	TF-IDF	Doc2Vec
SVM	51.94	51.94	11.24
RF	54.65	56.20	13.18
MLP	58.91	57.76	10.86

Source: The author (2022).

In addition, Table 10 provides the median performance metrics of the best classifier (i.e., MLP-BoW) for test data and for each category. Even with augmented data, the model was not able to correctly classify the instances belonging to augmented categories, which supports the idea that the data found in the reports are not of good quality. In other words, create artificial instances from poor-defined data jeopardizes the model performance as the technique increased the number of low-quality instances in the dataset. As this behavior is presented in all models of Table 9, the following analyses are not performed in the augmented database.

Table 10 – Median performance metrics (%) of on test data with MLP-BoW (trained with augmented data).

Label	Precision (%)	Recall (%)	F_1-score (%)
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	50.00	18.18	26.66
3	55.55	69.23	61.64
4	0.00	0.00	0.00
5	56.25	36.36	45.28
6	0.00	0.00	0.00
7	65.63	73.33	69.27
8	60.00	36.36	45.28
9	55.55	69.23	61.64

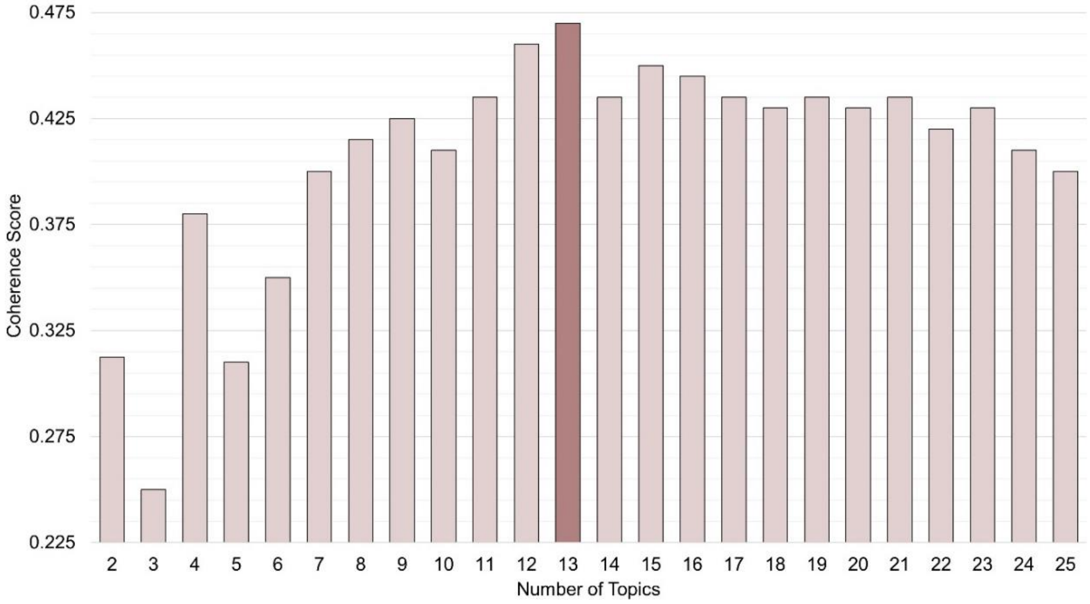
Source: The author (2022).

Moreover, despite being a more elaborated model, the performance of the Doc2Vec model is much worse in both analyses compared to the others (Table 8 and Table 9). Indeed, Lee and Yoon (2018) argued that although Doc2Vec proved to be effective in binary classification, it is not clear whether Doc2Vec can perform well multiclassification tasks. The authors mentioned that since multiclassification task can be challenging for Doc2Vec, depending mainly on the number of categories, to get satisfactory results it may be necessary to augment the Doc2Vec-based features. Therefore, in the remainder of Section 4.3, we will discard Doc2Vec and only evaluate the TF-IDF and BoW models.

4.3.2 Topic Modeling

Next, aiming to identify potential inherent groups of accidents and evaluate the categorization of the accidents, topic modeling task is performed using LDA algorithm to summarize the descriptions and key topic groups. Then, we here defined a search space for the number of topics (2 to 25 topics) and adopted the number of topics that yielded the best average topic coherence score, which shows the weak or strong topic correlation as described in Section 2.2.2. Figure 12 depicts the average coherence score as we vary the number of topics.

Figure 12 – Average coherence score for a different number of topics.



Source: The author (2022).

As illustrated in Figure 12, the best score was achieved when thirteen topics were considered. However, thirteen topics is a rather large number considering the total of accidents in our database. This result is probably caused by the variability on the accident descriptions and possible overlapping topics. Table 11 summarizes some topics achieved (translated into English), providing the four main concepts (i.e., terms / words) that represent each topic.

Table 11 – Concepts describing each topic.

	Concept			
Topic	**1st**	**2nd**	**3rd**	**4th**
1	*Bus*	*Same*	*Get off*	*Employee*
2	*Stairs*	*Slipped*	*Wet*	*Get down*
3	*Coming*	*Suffered*	*Employee*	*Driving*
4	*Employee*	*Coming*	*Foot*	*Injury*
5	*Knee*	*Left*	*Day*	*Vehicle*
6	*Stepped*	*Right*	*Building*	*Eye*
7	*Same*	*Key*	*Moto*	*Work*
8	*Vehicle*	*Causing*	*Left*	*Hand*
9	*Entrance*	*Access*	*Sidewalk*	*Step*
10	*Chair*	*Right*	*After*	*Step*
11	*Floor*	*Went down*	*Slipped*	*Catch*
12	*Contusion*	*Left*	*Shoulder*	*Rise*
13	*Room*	*Ankle*	*Ramp*	*Coming*

Source: The author (2022).

Indeed, it is possible to notice that many topics overlap. For instance, both topics 2 and 11 involve ‘slipping’. However, they differ in terms of the ‘accident agent’: ‘stairs’ in topic 2, while ‘floor’ for topic 11. Moreover, topics 4 and 13 handle with events in which an employee walks (see the action verb ‘coming’ associated with the employee action) and, then, his/her ‘ankle’ or ‘foot’ are injured. Note that topics 1, 3, 4, 7, 9, and 13 seemingly represent accidents that occurred when the employee was arriving/going to work; topics 2, 10, and 11 correspond to trip and fall accidents; while topics 5 and 8 may characterize accidents caused by vehicles and topics 6 and 12 seems to be lone topics. Based on the outcome of topic modeling, the evaluation of the results after DA, as well as on the results found in (GUIMARÃES, ARAÚJO, *et al.*, 2020), the accidents can be put together in larger groups, resulting in less categories.

Thus, we decided to perform a laborious, yet necessary, database manual curation. Note that even if we performed topic modelling on the clusters identified by Guimarães et al. (2020), a manual curation would probably still be necessary to label each accident agent, because the descriptions used for clustering/topic modeling encompass not only the accident agents, but the accident conditions, consequence and other factors. We discuss the manual curation in next Section.

4.3.3 Restructuring the Reports’ Database

We evaluated the original pre-defined categories, merging the ones that seem to be similar. The restructuring resulted in six categories as shown in Table 12. This step was essential as it allowed us to identify accident agents misclassified by the safety technician (see Table 7).

Table 12 – Six Categories of accidental events.

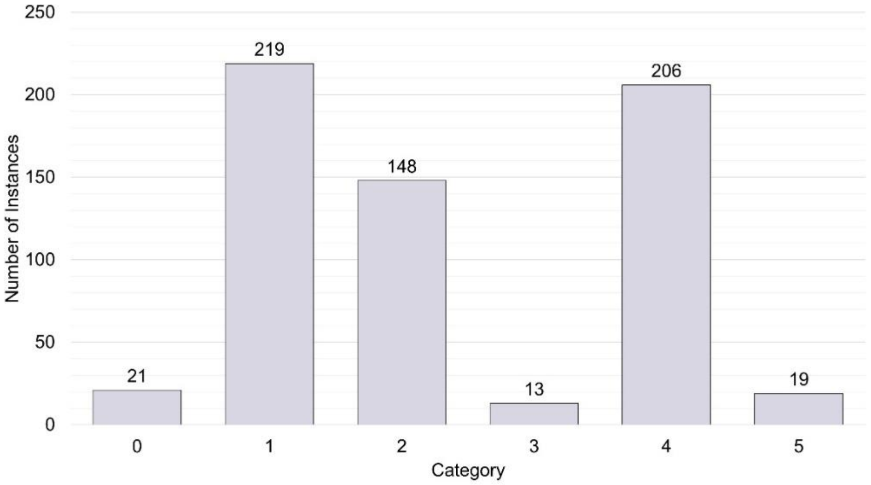
Label	Categories
0	Scaffolding
1	Administrative fall/ injury
2	Equipment/ tools
3	Chemical products
4	Commuting
5	Others

Source: The author (2022).

After reorganizing the database, we proposed new labels, and the number of instances in each class is presented in Figure 13. Five groups (0-4) are formed by accident investigation reports with common causes within each group, and the remaining set (5: ‘others’) is composed by events that have different causes and are not assigned to any specific category. The categories ‘administrative fall/ injury’ refers to accidents in the administration building due to person movement, while ‘commuting’ represents accidents due to employee displacement, with or without a motorized vehicle, outside the company’s building.

Despite the restructuring, there are still categories (0: ‘scaffolding’, 3: ‘chemical products’, and 5: ‘others’) with rather few instances. One may expect that these accidents were related of more severe consequences, which has a lower frequency according to the Heinrich’s triangle (BATTIATO, FARINELLA, *et al.*, 2018). However, considering the information of ‘injury leave’ provided in the original database, the accidents related to these less frequent agents are not necessarily associated with more severe consequences: 61.9%, 53.8%, and 63.1% of the accidents categorized as 0, 3, and 5 respectively led to injury leave compared to 57.5%, 55.4%, and 68.9% of classes 1, 2 and 4. In other words, a frequent ‘accident agents’ (e.g., 4: ‘commuting’) led to severe consequences. These results reinforce the hypothesis that the investigation reports were not filled in correctly.

Figure 13 – Number of instances per ‘accident agent’ considering the new categorization after the curation



Source: The author (2022).

Next, we performed the same steps of the methodology (i.e., preprocessing and conversion into feature vector representations) before training and, then, we tested each model 10 times. Table 13 presents the classification median accuracies. The manual labeling increased the accuracy by about 15 percentage point in some cases (e.g., SVM-TF-IDF) compared to Table 8. The significant improvement in performance confirms the assumption of mislabeling in filling out accidents’ investigation report based on the original categories. Indeed, the new categorization apparently is more coherent, aligned with the safety technician view; thus, being beneficial for the identification of patterns.

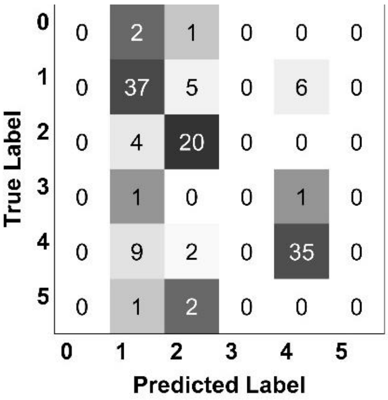
Table 13 – Median accuracy (%) of the classifiers on test data adopting six categories of accidental events

Classifier	**BoW**	**TF-IDF**
SVM	64.29	68.26
RF	67.86	67.06
MLP	70.63	69.44

Source: The author (2022).

Moreover, Figure 14 shows the confusion matrix for the results of one round of CV on the test set for the MLP-BoW, which presented the best median accuracy. In the confusion matrices, the element in the r-th row and j-th column, $c_{r,j}$, indicates the number of observations j predicted as r. For example, the element $c_{0,0}$ of the confusion matrix indicates that 0 of the instances were corrected classified as 0.

Figure 14 – Confusion matrix for classification of test data with MLP-BoW model



	0	1	2	3	4	5
0	0	2	1	0	0	0
1	0	37	5	0	6	0
2	0	4	20	0	0	0
3	0	1	0	0	1	0
4	0	9	2	0	35	0
5	0	1	2	0	0	0

Source: The author (2022).

Despite the overall improvement in the performance, the results still indicate a poor representation of different categories with fewer data (i.e., 0, 3, and 5). In fact, the median performances for these categories were all zero; the model only correctly classified the instances labeled as 1, 2, and 4.

Yet, 9 of 11 errors for classifying label 4 (‘commuting’) consist of predicting label 1 (‘administrative fall/injury’) and 6 of 11 errors for classifying label 1 consist of predicting label 4. This is probably because these ‘accident agent’ categories are inherently similar and, then, it negatively interfered in the models’ learning process. In order to illustrate these errors, Table 14 shows some examples of misclassified accidents' descriptions labeled as 1 and 4.

The first instance (label 4) misclassified as label 1 contains a consequence (underlined in the description) that is common to label 1 (e.g., the employee slipped in the office and sprained his/her ankle). The second instance also comprises a consequence common to accidents categorized as 1; moreover, the accident involves slipping in high heels, which is a frequent cause of administrative falls/injuries. The description of instances 2 and 3 also encompasses the terms 'slip' and 'trip'. In addition, instances 1, 2 and 3 involve employee displacement to work; according to standard NBR 14280 (NBR, 2001), accidents labeled as 1 involve displacement within the company's facilities. Since the employees were walking to work, the accidents occurred due to similar causes ('slip' and 'trip') and had similar consequences ('sprained his/her ankle') as 'administrative fall/injury' accidents. The fourth instance describes a 'stepping on a stone' situation, which is an odd condition for an administrative accident. In addition, the accident described in the fifth instance occurred when the employee entered a facility by walking. The terms 'entering' and 'arriving' are oftentimes used to describe commuting accidents. Finally, in the examples provided one can see that the descriptions use very similar terms regardless the categories. Thus, it is plausible that the model had difficulties in differentiating label 1 and label 4.

Table 14 – Instances misclassified by SVM-TF-IDF model regarding label 1 ('administrative fall/injury') and label 4 ('commuting') accident agents

Instance	Accident description	Accident agent	
		Label	Predicted
1	The employee went to work at around 8:20 a.m. when he *sprained* his *ankle* when *stepping* on the uneven side of the sidewalk.	4	1
2	The employee walked from home to work and when passing on a sloping sidewalk she *slipped and fell* to the ground. She was not in a hurry at the time, she wore a clog and suffered a *fractured* *left* *ankle*.	4	1
3	The employee *tripped* over a loose stone on the ground when walking down the sidewalk on his way to work, *falling on his knee and suffering bruises*.	4	1
4	While performing an activity the employee *stepped on a stone*, fell with the right side of his chest on the stone, and suffered a chest injury.	1	4
5	When *entering* the accommodation, the employee did not notice the glass door and bump into it, suffering a cut on the left eyebrow.	1	4

Source: The author (2022).

The result for the class ‘others’ is somehow expected because it is made up of reports of events that have different causes and are not common to the other categories. Thus, from the few instances available, the classification methods were not able to identify features to represent this class. Indeed, the classifiers try to assign these accidents to classes that have the maximum similarity, which once again, might hinder their performance. For example, the accident description ‘during a competition held by the team, one of the recreational activities was tug-of-war. When pulling the cable, the employee felt pain in the lower back’ was labeled as 2 (‘equipment/ tools’) instead of 5 (‘others’) probably because of the expression ‘pulling the cable’ makes that accident to be incorrectly related to the maintenance/operation of some equipment.

4.3.4 Parameter Tuning

Each ML algorithm requires a set of hyper-parameters in its formulation, and their proper estimation demands attention to best adjusts the mapping function (MAIOR, SANTANA, *et al.*, 2018). Thus, we here applied Grid-Search (GS) and CV algorithms to fine-tune their hyper-parameters. GS looks into a specified searching space (subset of hyper-parameters), while k-fold CV is applied to the training dataset to find the hyper-parameters resulting in the best performance measured on the validation set. Additionally, k-fold CV provides less-biased and less-optimistic results equally dividing dataset into k parts, where $k - 1$ are used for training and 1 for validation purposes.

This process is executed k times, changing the validation portion until all training data have been used. This procedure was here performed over training data, with $k = 10$. Table 15 presents the search spaces for GS-CV approach for each ML model, see (PEDREGOSA, VAROQUAUX, *et al.*, 2011) for further details.

Table 15 – Search space for GS for each classifier and description of the hyperparameters

Classifier	Hyper-parameters	Value / Type range
SVM	gamma	[10-5, 10-4, 10-3, 10-2]
	C	[10, 102, 103, 104]
MLP	hidden layers size	[(100), (100, 100), (100, 100, 100)]
	activation	[‘tanh’, ‘relu’]
	solver	[‘sgd’, ‘adam’]
RF	max depth	[20, 30, 40, 50]
	n estimators	[50, 83, 116, 150, ‘None’]

Source: The author (2022).

The results on a test set with fine-tuned hyper-parameters are summarized in Table 16. The results slightly improved for most classifiers. The greatest enhancement (5 percentage point) was obtained for SVM-TF-IDF, which is particularly interesting for discarding the hypothesis that a greater performance is not achieved due to the poor selection of the hyper-parameters of the classifiers.

Table 16 – Accuracy (%) of the classifiers with fine-tuned hyper-parameters on test data adopting six categories of accidental events.

Classifier	BoW	TF-IDF
SVM	65.08	73.02
RF	69.84	67.46
MLP	69.05	72.22

Source: The author (2022).

As we mentioned, it is noteworthy that, despite the reduction in the number of categories from nine to six, there are still many of them with a limited amount of data (i.e., 0, 3, and 5). Thus, even with parameter tunning, the models may not have been able to properly learn the patterns to identify these categories due to the small number of samples.

4.3.5 Sensitivity Analysis

Finally, we performed a sensitivity analysis regarding the three classes with the least frequent data (i.e., 0, 3, and 5) to assess the impact of these categories on the overall performance. Specifically, we considered the four cases in which we removed the instances categorized as ‘scaffolding’, ‘chemical products’, and ‘others’ individually, and the case where they are jointly crossed out. Then, we performed the hyper-parameters selection via GS-CV and retrained the models still using the 80/20 ratio for training/testing. Table 17 presents the accuracy obtained of the models on test data. Compared to the performance of Table 13 removing one category at a time generally led to better results. For example, taking out ‘scaffolding’ or ‘chemical products’ provided an increase of 8 percentage point in the RF-BoW performance. Moreover, the removal of ‘others’ raised 5, 4, and 4 percentage point in the SVM-TD-IDF, RF-BoW, and MLP-TF-IDF accuracy respectively. Yet, removing all samples that belong to classes with fewer data (i.e., labeled with 0, 3 and 5) clearly improved the performance for all classification and representation models (compare Table 13 and the last line of Table 17).

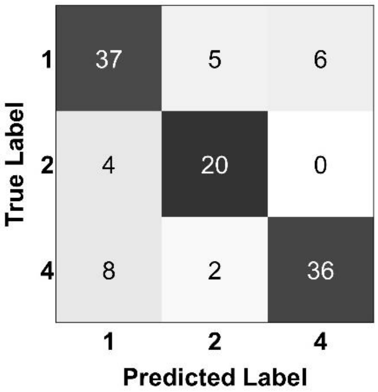
Table 17 – Accuracy (%) of classifiers on test data after removing classes 0, 3 or/and 5.

Class Removed	BoW			TF-IDF		
	SVM	RF	MLP	SVM	RF	MLP
0: ‘scaffolding’	66.67	75.61	69.92	73.27	70.73	73.98
3: ‘chemical products’	66.13	75.00	74.19	75.00	70.97	75.00
5: ‘others’	66.67	71.74	71.74	73.98	69.11	73.98
0, 3 and 5	69.49	76.27	76.27	77.97	74.58	78.81

Source: The author (2022).

To illustrate the classification performance regarding each category, Figure 15 presents the confusion matrix for the results of the best model without the three categories mentioned (i.e., MLP-TF-IDF), while Table 18 shows the performance metrics for each remaining category. Even with the concerns regarding the safety technician labeling procedure, the account of classes with a greater number of reports led to balanced and reasonable results.

Figure 15 – Confusion matrix for the test classification with MLP-TF-IDF model after removing classes 0, 3 and 5



Source: The author (2022).

Table 18 – Performance metrics on test data of MLP-TF-IDF model after removing 'scaffolding', 'chemical products', and 'others' for each accident label

Label	Precision (%)	Recall (%)	F_1-score (%)
1	75.51	77.08	76.29
2	74.02	83.33	78.43
4	85.71	78.26	81.82

Source: The author (2022).

The model could satisfactorily learn patterns of these categories, with F_1-scores close to 80% for all classes. The precision for label 2 is slightly lower than for categories 1 and 4; probably because category 2 has the smallest number of samples and, then, the model did not learn all the subtleties to identify this ‘accident agent’ factor. Note that these errors also affected the recall for labels 1 and 4 (instances labeled as 2 were misclassified as 1 and 4). Thus, the consideration of a truly balanced dataset (i.e., with original reports, not augmented ones) may be helpful in analyzing the performance metrics.

4.3.6 Conclusion

In Section 4.2 we presented the results of the NLP-based classifiers with the original categorization of the accidents reports that are used as a starting point of the analysis. Then, the results of the exploratory analysis, obtained through an unsupervised learning algorithm, provided us with an overview of the accident investigation reports, serving as a tool to identify possible systematic issues in these reports. Indeed, in Section 4.3.2 we identified a high number of topics and also observed the overlap among them. These results allowed us to postulate a possible problem related to the categorization of the accidents, which led us to perform a manual curation of the database where we proposed a new categorization of ‘accident agent’ and reclassified the accidents. Next, in Section 4.3.3 we present a new categorization and the performance of the NLP-classifiers on the classification task using the updated database (accident investigation reports with their new labels). Finally, the improvement observed in the performance of the classifiers indicates that the proposed categories are more representative and contain better quality information for training the classifiers.

Here we employ NLP as a tool to improve the quality of the information contained in the reports. Thus, the result obtained through the proposed methodology would be more useful accident investigation reports that could be adopted by methodologies already proposed by other authors. To demonstrate this, in the next section we adopt the resulting reports for predicting injury leave.

4.4 INJURY LEAVE CLASSIFICATION

This part of the study aims to analyze the accident investigation reports resulting from the proposed methodology, i.e., with accidents relabeled with the proposed accident agents, to illustrate the practical usefulness of the modified reports to support decision making. The goal is to determine whether there will be an injury leave, in which this information is fundamental to effectively defining strategies for the worker's absence. The results and methodology described here were accepted in the Journal of Risk and Reliability (RAMOS, Plinio, MACÊDO, *et al.*, 2022).

Investigation of injury leaves caused from accidents is important when the evaluation of unavailability and worktime loss is of interest. The employee’s absence can lead the company to spend financial resources. Indeed, according to the Brazilian National Institute of Social Security (INSS, 2022), the organization pays the employee’s salary within the first 15 days after the accident, and then the government pays the employee's social security benefit for the rest of the leave period. However, for the company, it is still mandatory to collect the worker's guarantee fund, besides possibly hiring a new employee for the task. Indeed, significant research effort has been put into the analysis of occupational accidents regarding the assessment of rates of accident and recovery (PAREJO-MOSCOSO, RUBIO-ROMERO, *et al.*, 2012).

Pre-trained language models such as BERT (described in Section 2.4) can be fine-tuned with specific domain data and provide state-of-the-art results even with only a few hundred samples (SI, WANG, *et al.*, 2020). Indeed, some successful applications in risk and reliability have been published (MACÊDO, MOURA, AICHELE, *et al.*, 2022, MOON, CHI, *et al.*, 2022, ZHANG, Lite, WANG, *et al.*, 2022, ZHOU, Bing, ZOU, *et al.*, 2022), but there is still a gap in the application of these powerful models regarding the use numerical variables, which are fundamental to our context. The proposal is to use BERT model for the textual variable, which, when concatenated with the non-textual variables, feeds a classifier model. To analyze the results, we considered one specific class of the category ‘accident agent’, that was proposed after the application of the proposed methodology (Section 4.3.3) and represents the element related to the unsafe condition that caused the accident.

Thus, the trained classifier would be used in practice to process new accident reports. As new accidents would occur, the safety technician would fill out an accident report and the model would provide as an output whether the injury would result as a leave or not. Managers can use the model to reorganize their workforce due to the shortage of human resources, or even assist the management of financial resources such as the need to cover employee leave or hire temporary employees. Here, we consider not only the accident description, which is written by the company safety technician, but also other numerical and categorical information presented in the database such as the use or not of PPE and the employee’s years of work.

4.4.1 BERT-based Classifier

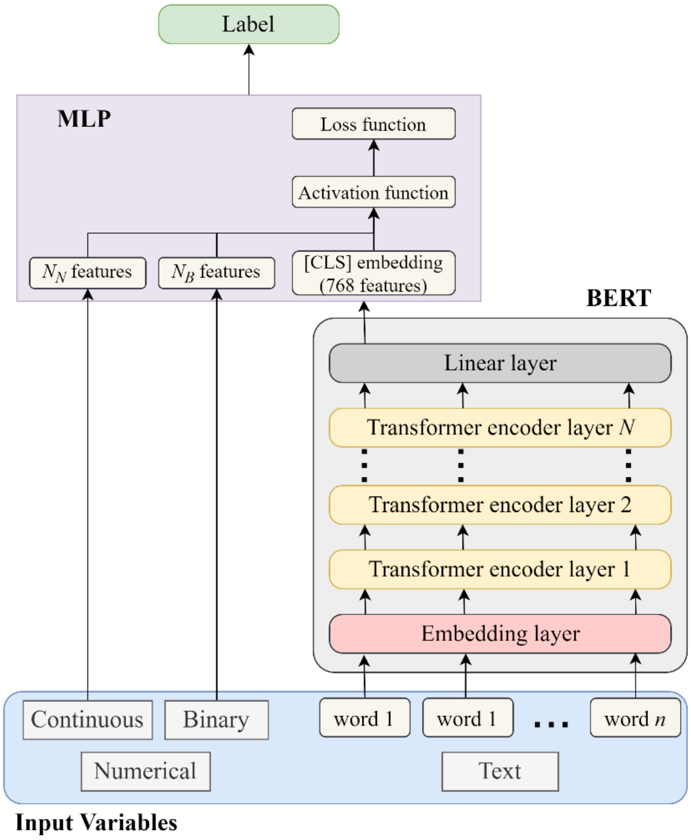
Accident reports are not usually restricted to the event description as they may contain fields filled in with numerical, categorical and/or binary information to detail the accident. This information can be used to identify common causes, consequences and sources of accidents, as well as propose preventive and mitigative measures. One common approach, when applying models to process natural language, such as BERT, is to combine numerical and categorical variable with textual variable by converting all variables to text (MACÊDO, MOURA, RAMOS, *et al.*, 2022). Although BERT can understand natural language, it may not be able to capture all information contained in values relative to continuous numbers, or even a large number of discrete numbers (WALLACE, WANG, *et al.*, 2020). Thus, an alternative approach is to concatenate the non-textual variables for BERT embedding and classify the resulting vector through a neural network.

Here we changed BERT's architecture (Figure 7) to incorporate the numerical data into the word representations. In other words, during fine-tuning, we update the entire architecture not only with text data but also with the numerical data. Thus, the generated embeddings take into consideration not only the specific context as technical terms, but also quantitative information about the analyzed scenarios. A schematic overview of our proposed BERT-based classifier is shown in Figure 16.

From the modified accident investigation reports we select textual variables (e.g., accident descriptions), numerical variables (e.g., years of experience of the injured person, age of the worker) and binary variables (e.g., the use of PPE, training in occupational safety) that can be used as predictor variables in a classification model.

The numerical (continuous and binary) variables will be concatenated from the 768 standard BERT features that will be used to represent the textual variables. These combined features are input to a MLP for the final classification. Each step was developed in the Python computational language and is detailed below. We used the Pytorch implementation of pre-trained BERT available at transformers library (WOLF, DEBUT, *et al.*, 2020). We here built our model on top of the ‘bert-base-multilingual-cased’ version of the ‘BertForSequenceClassification’ model which is a version pre-trained in 104 languages (including Portuguese, in which our text data are originally written).

Figure 16 – Overview of the proposed BERT-based classifier.



Source: The author (2022).

The MLP implemented has 770 nodes in the input layer (dimension of our combined input vector) and the dimension of the MLP layers were defined as $\left(\frac{1}{4}\right)$ of the nodes of the previous layer. Thus, the resulting MLP has 4 hidden layers with 192, 48, 12 and 3 nodes, and a final layer with 2 nodes (since we are performing a binary classification). See Table 19 for a detailed description of the system parameters.

We apply a CV algorithm based on the Leave-P-Out cross-validator (LPO-CV) (PEDREGOSA, VAROQUAUX, *et al.*, 2011), to evaluate the performance of our model. Using the CV has a high chance of detecting whether our model is being overfitted. In this way, the LPO-CV method creates all the different training/validation sets revoking p samples from the entire set. For n samples, this method produces $\left(\frac{n}{p}\right)$ train-test pairs.

From this, a confusion matrix with the prediction of the test data for each run was formed, to assist in the performance evaluation of the models. The classifier's performance has been evaluated considering the classification accuracy, precision, recall, and F_1-score for each round.

Table 19 – System trainable parameters.

Block	Type	Shape	Activation	Amount	Parameters
Embedding	word_embeddings	(11954, 768)		1	9,180,672
Embedding	position_embeddings	(512, 768)		1	393,216
Embedding	token_type_embeddings	(2, 768)		1	1,536
Embedding	LayerNorm.weight	(768)		1	768
Embedding	LayerNorm.bias	(768)		1	768
Dropout	dropout			1	
BertSelfAttention	query.weight	(768, 768)	Linear	12	
BertSelfAttention	query.bias	(768)	Linear	12	
BertSelfAttention	key.weight	(768, 768)	Linear	12	
BertSelfAttention	key.bias	(768)	Linear	12	
BertSelfAttention	value.weight	(768, 768)	Linear	12	
BertSelfAttention	value.bias	(768)	Linear	12	
Dropout	dropout			12	
BertOutput	dense.weight	(768, 768)	Linear	12	
BertOutput	dense.bias	(768)	Linear	12	
BertOutput	LayerNom.weight	(768)		12	
BertOutput	LayerNorm.bias	(768)		12	
BertIntermediate	dense.weight	(3072,768)	GELUActivation	12	
BertIntermediate	dense.bias	(3072)	GELUActivation	12	
BertOutput	dense.weight	(768, 3072)	Linear	12	
BertOutput	dense.bias	(768)	Linear	12	
BertOutput	LayerNom.weight	(768)		12	
BertOutput	LayerNorm.bias	(768)		12	
Dropout	dropout			12	
BertPooler	dense.weight	(768,768)	Linear	12	
BertPooler	dense.bias	(768)	Linear	12	
MLP	init	(770,3)	tanh	1	
MLP	final_layer		Linear	1	

Source: The author (2022).

4.4.2 Data Analysis

As mentioned, here, we focused on predicting if an occupational accident led to injury leave or not. In the modified accident investigation reports, the accidents were classified into six different ‘accident agent’ categories. Here, we focus our analysis on the class with more samples, ‘administrative fall/injury’, which contains 219 reports. Other different classes, which have fewer accident reports e.g., 13 reports in the ‘chemical products’ class), would increase the list of words (vocabulary) for the model but would not provide enough data for the model to extract patterns.

After evaluating all the characteristics present in the reports, we consider that three of them represent a relevant source of information for the classifier: (i) accident description (provided as free text), (ii) use of PPE during the accident (binary variable), and (iii) the employee’s length of experience (continuous variable). Thus, we removed the accident samples that did not present these variables filled in the report. This reduced 38 samples of the specific category. The binary variable assumes 0 value if the employee was not using PPE and 1 otherwise. The continuous variable was scaled between 0 and 1.

Finally, the texts are preprocessed to remove noise as described in Section 4.1.1. In addition, to mark the beginning and the end of each sentence, it was necessary to add [CLS] and [SEP] respectively; this is because BERT was pre-trained using the format [CLS] sentence [SEP]. Moreover, it is essential to use the same tokenization to fine-tune a pre-trained model; for this reason, we used the ‘BertTokenizer’ backed by transformers library, which splits the sentences into a sequence of tokens according to punctuation and sub-word units, converts raw text to sparse index encodings, and stores the vocabulary token-to-index map. Thus, the cleaned sentences were processed by the tokenizer. In addition, the tokenizer transforms all sequences to a maximum length by adding zeros, since the model requires inputs that have the same shape and size.

4.4.3 Data Augmentation

Here, after the 80/20 split, the DA procedure was applied to the test set to: (i) balance the ‘administrative fall/injury’ category of accident agent; (ii) to increase the dataset. Next, in Table 20, an example of how the DA procedure works.

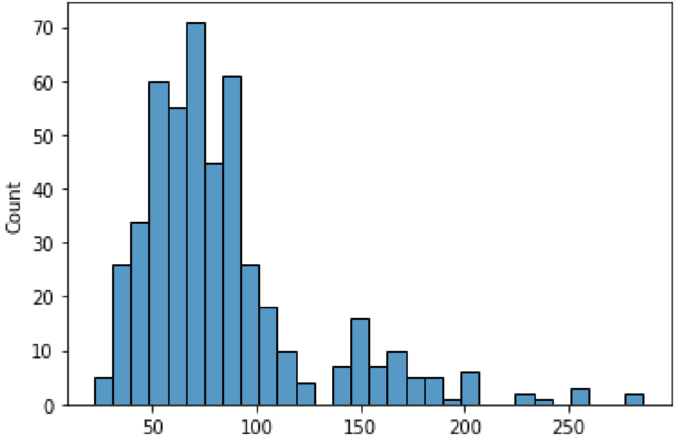
Table 20 – Example of DA procedures in an adapted description.

Procedures	**Description**
Preprocessed original	*after parking truck getting out of he did not realize that there was gap ground slipping hurting his right knee*
DA1	*after parking **car** getting out of he did not realize that there was gap ground slipping hurting his right knee*
DA2	*after parking truck getting out of he did not realize that there was gap ground slipping **harming** his right knee*

Source: The author (2022).

Therefore, the training set consists of 480 samples among augmented and original descriptions (145 original), whereas the test set contains 36 non-augmented descriptions (19 reports of injury leave and 17 reports of non-injury leave). The sentence length of these reports varies according to the detail of each accident, providing different terms for vocabulary training. The sentence length histogram is shown in Figure 17.

Figure 17 – Histogram of sentence lengths.



Source: The author (2022).

We noticed that all sentences have a length of less than 300. which is smaller than BERT's sequence length limit of 512 tokens. Next, during the tokenization process, we added [CLS] and [SEP] to mark the beginning and end of each sentence, making the sentences usable by BERT who have been pre-trained using this format. Table 21 shows the post tokenization sentence.

Table 21 – Example of tokenization procedures in an adapted description.

Procedures	**Description**
Tokenization	*[CLS] [after] [parking] [truck] [getting] [out] [of] [he] [did] [not] [realize] [that] [there] [was] [gap] [ground] [slipping] [hurting] [his] [right] [knee] [SEP]*

Source: The author (2022).

Characterizations of the dataset before and after the pre-processing steps described above are reported in Table 22. After the sentences are cleaned up and tokenized, we transform them to a maximum length of 300 tokens for all sentences by adding zeros so that all inputs have the same shape and size. From there, the textual variable will go to the BERT which will be concatenated with the other variables, and finally feed the MLP classifier.

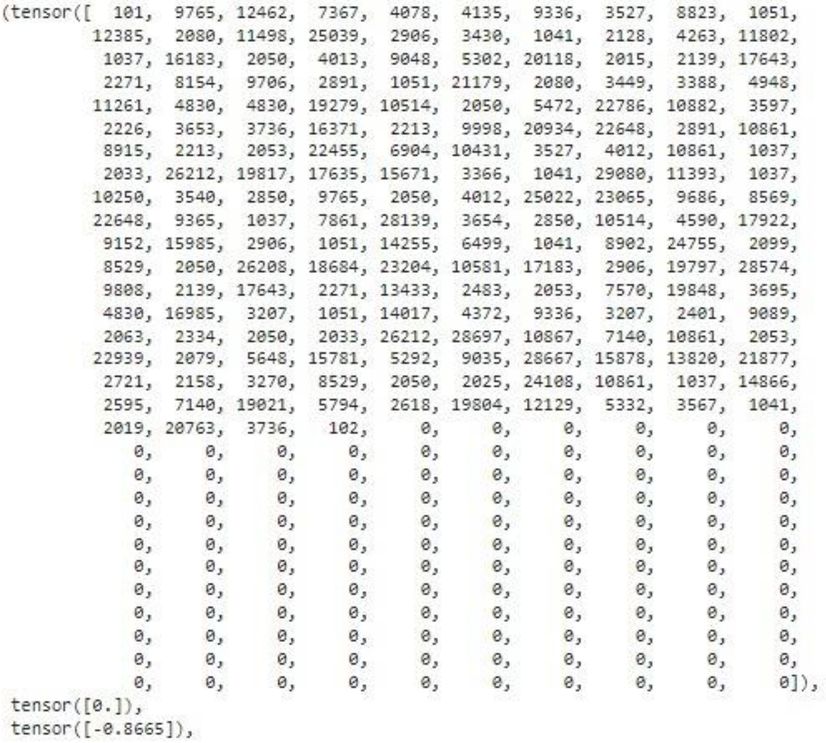
Table 22 – Summary descriptive statistics of documents, before and after pre-processing steps.

Measure	Before Preprocessing	After Preprocessing
Number of observations	291	181
Vocabulary size (distinct word count)	2025	1703
Mean word count per document	43.96	28.62
Standard deviation word count per document	25.95	16.83

Source: The author (2022).

Finally, the processed texts are converted into a numerical representation using BERT. Thus, one can feed the ML model with the concatenated vector, as shown in Figure 18.

Figure 18 – Tokenization and encoding concatenated vector.



```
(tensor([  101,  9765, 12462,  7367,  4078,  4135,  9336,  3527,  8823,  1051,
         12385,  2080, 11498, 25039,  2906,  3430,  1041,  2128,  4263, 11802,
          1037, 16183,  2050,  4013,  9048,  5302, 20118,  2015,  2139, 17643,
          2271,  8154,  9706,  2891,  1051, 21179,  2080,  3449,  3388,  4948,
         11261,  4830,  4830, 19279, 10514,  2050,  5472, 22786, 10882,  3597,
          2226,  3653,  3736, 16371,  2213,  9998, 20934, 22648,  2891, 10861,
          8915,  2213,  2053, 22455,  6904, 10431,  3527,  4012, 10861,  1037,
          2033, 26212, 19817, 17635, 15671,  3366,  1041, 29080, 11393,  1037,
         10250,  3540,  2850,  9765,  2050,  4012, 25022, 23065,  9686,  8569,
         22648,  9365,  1037,  7861, 28139,  3654,  2850, 10514,  4590, 17922,
          9152, 15985,  2906,  1051, 14255,  6499,  1041,  8902, 24755,  2099,
          8529,  2050, 26208, 18684, 23204, 10581, 17183,  2906, 19797, 28574,
          9808,  2139, 17643,  2271, 13433,  2483,  2053,  7570, 19848,  3695,
          4830, 16985,  3207,  1051, 14017,  4372,  9336,  3207,  2401,  9089,
          2063,  2334,  2050,  2033, 26212, 28697, 10867,  7140, 10861,  2053,
         22939,  2079,  5648, 15781,  5292,  9035, 28667, 15878, 13820, 21877,
          2721,  2158,  3270,  8529,  2050,  2025, 24108, 10861,  1037, 14866,
          2595,  7140, 19021,  5794,  2618, 19804, 12129,  5332,  3567,  1041,
          2019, 20763,  3736,   102,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0]),
 tensor([0.]),
 tensor([-0.8665]),
```

Source: The author (2022).

The first tensor represents the textual feature after tokenization and encoding, the second tensor refers to the binary variable, and finally, the third tensor represents the continuous variable after normalization. These combined variables are input to an MLP that predicts the occurrence of the accident leave for a given accident, 0 for an accident without injury leave and 1 otherwise. Instead of adopting a simple strategy of combining all variables as text and feeding it through BERT, the numerical/binary variables are treated separately to improve the performance of the classifier.

4.4.4 Results and Discussion

We considered 10 different training/validation sets, randomly split in a proportion of 90/10, to avoid biases and to account for the variability. It is noteworthy that, unlike k-fold CV, the validation set samples may overlap considering the different splits of the complete set. Considering this, we computed the median and standard deviation for the accuracy of the classification on test set, seen in Table 23. The median results for the other metrics for each label are shown in Table 24.

Table 23 – Median and standard deviation for the accuracy test set.

Measures	Median	Standard deviation
Accuracy	73.50 %	3.18

Source: The author (2022).

Despite the median being around 73%, one of the models (rounds) achieved the best test accuracy of 78%. In fact, seven of the ten (7/10) rounds had test results greater than 70%, with the smallest three achieving close results (69%). We noticed that all the medians of the metrics reached results above 70%.

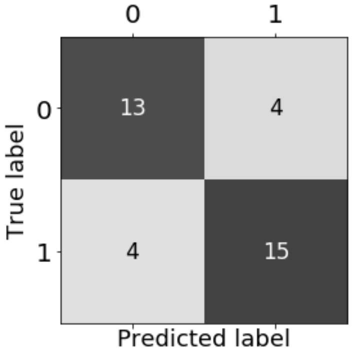
Table 24 – Median of test data classification results for each consequence.

Consequence/Mensure	Precision	Recall	F_1-score
0	71.5%	76%	71.5%
1	76.5%	74%	74.5%

Source: The author (2022).

In particular, the injury leave consequence had a median precision above 76%, which is interesting for our application. Figure 19 shows the confusion matrix for the results of one round of CV on the test set, where 0 represents no injury leave and 1 represents that there was a leave.

Figure 19 – Confusion matrix for classification of test data with the best result.



Source: The author (2022).

As one can see, there were 4 FP and 4 FN in the test set classification, which means that 4 accidents that did not lead to injury leave were erroneously classified as accidents that led to injury leave; meanwhile 4 accidents that led to injury were misclassified as accidents that did not lead to injury leave.

Finally, we present reports correctly classified by most of the models and others frequently incorrectly classified. Table 25 and Table 26 present the textual, binary, and numerical variables of these reports, in addition to the true and predicted values of the injury leave label. The hit rate, as the name implies, is the proportion of times that report was correctly predicted in the models.

Table 25 – Adapted reports with the higher hit rates.

Model input			**Ground truth**	**Models output**	**Summary**
Description	**PPE**	**Years**	**True label**	**Majority predicted label**	**Hit rate**
*the employee was working in the warehouse when he tripped and fell **fracturing** his right arm*	True	53	1	1	10/10
*the employee lost his balance and fell from a height of about 6 steps to reach the ground floor suffering **light injuries** to his shoulder leg and right elbow*	True	26	0	0	10/10
*when the employee was executing a maneuver to remove the mechanical lock there was a sudden displacement taking the right **hand** with it and causing injury to the distal phalanx of the little **finger***	True	45	0	0	10/10

Source: The author (2022).

We noticed that in the first report of Table 25 the element ‘fracturing’ in the description, despite the use of PPE, is pertinent to indicate the occurrence of injury leave. While in the second document the elements as ‘light injuries’, and the use of PPE can indicate the non-leave. In the third document, despite the use of PPE, elements such as ‘hand’, ‘finger’ and the absence of any term to describe the seriousness of the situation, can mean indicating the non-leave.

In the reports that had a low hit rate, some elements may have contributed to misleading the classifier. For example, in the first one of Table 26, the use of PPE in addition to description elements such as ‘hand’ and ‘finger’, has led most models to classify it as non-leave. In the second document, elements such as ‘bruise’, despite not using PPE, can mean a minor consequence of the accident, which caused the model to misclassify as well.

Table 26 – Adapted reports with the lower hit rate.

Model input			**Ground truth**	**Models output**	**Summary**
Description	**PPE**	**Years**	**True label**	**Majority predicted label**	**Hit rate**
*the employee was storing the material for removing the ladder this one got its hooks stuck in the structure the employee went up to release it one of the hooks came loose and the ladder descended hitting the 4th **finger** of the left **hand** causing dislocation*	True	39	1	0	1/10
*the employee was transporting equipment to the warehouse and then when he pulled one of the ropes it came loose causing him to twist his body causing a **bruise** on his right leg*	False	50	1	0	3/10

Source: The author (2022).

Thus, the absence of terms to describe the severity of the situation and accidents with injury leave, even with the use of PPE, may have contributed to misleading the classifier. In fact, the reports that had a low hit rate had elements that did not characterize the accidents to justify or not the injury leave.

4.4.5 Conclusion

Using the modified accident reports we were able to build a classifier to predict the occurrence of injury leave with an accuracy of 78%. A model that correctly predicts worker leave can support managers to propose an organizational plan to effectively deal with the worker's absence from the job before the leave is officially communicated. An interesting point to be highlighted is that because our analysis consisted of a very small subset of the accident reports (i.e., we analyzed ‘administrative fall/injury’ accidents), the use of a pre-trained model as BERT is a smart solution, since the amount of data available would not be enough to train a NLP model from scratch and achieve satisfactory results. In addition, another advantage of using BERT is that are several pre-trained versions available in other languages, including Portuguese.

4.5 GENERAL CONCLUSION

The results obtained shows that NLP provides auspicious techniques to identify poor/imperfect datasets once their performance greatly relies on the quality and integrity of the database. The proposed approach was able to identify issues on the filling out of the reports as well as in the safety technician grasp on the standard NBR 14280 (NBR, 2001). Thus, adjustments are necessary to provide documents that are more capable of retaining the knowledge acquired from the accident events, and then could be reused by the company and improve the current safety environment.

The downside of the proposed methodology is that it is not yet completely automatic, still relying on manual curation. Moreover, the analyzed database has only 626 reports, while other studies worked on databases with more than 1,000 reports (BAKER, HALLOWELL, *et al.*, 2020, CHENG, LEU, *et al.*, 2012), not allowing building and training DL architectures from scratch. In addition, the database was unbalanced, and even using the DA procedure, no significant improvement was achieved. Finally, the quality of filling in the reports was poor, hindered the classifiers' performance, which explains the improvement observed after reclassification.

Using the improved reports, we inferred the occurrence of injury leaves for occupational accidents. In addition, we propose an approach that considers not only the accident description but also binary information such as the use or not of PPE at the time of the accidents and numerical information such as the years of work of the injured employee. These textual, binary, and numerical variables were combined, and the resulting vector was trained by the MLP classifier. Thus, the trained classifier can provide useful information that helps managers effectively deal with the worker's absence and its consequences (costs, work replanning).

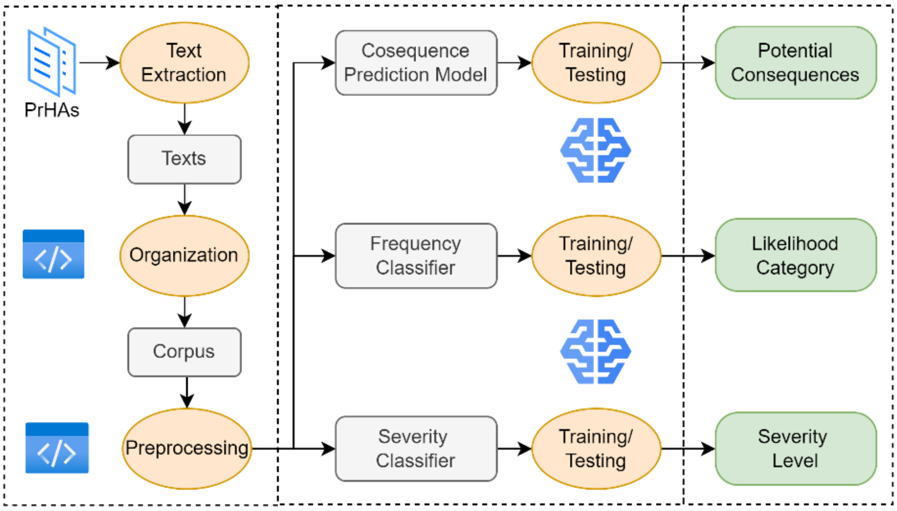
5 METHODOLOGY TO SUPPORT QRA IN O&G INDUSTRY

This part of the thesis aims at supporting identification and assessment of hazards with severe consequences related to the operation of an O&G industry, reducing efforts required to develop risk studies. Our idea is to develop NLP-models capable of learning and recognizing risk features, and thus extract useful knowledge about accidental scenarios. Some results discussed on this chapter was published in Process Safety and Environmental Protection (MACÊDO, MOURA, AICHELE, *et al.*, 2022) and registered in the national institute of industrial property, process number BR512022000211-6, (MACÊDO, MOURA, LINS, *et al.*, 2021).

5.1 METHODS

We adopted TM techniques to extract text data from PrHA documents, and then perform text classification tasks to identify risk features in oil refinery's subsystems. Then, we developed three models. Figure 20 provides an overview of the proposed methodology. First, we developed two scripts: one that automatically extracts text from a collection of PrHA spreadsheets, and another to organize and build an annotated corpus for each supervised-learning task, also referred to as dataset in this study. Next, the corpus was preprocessed and converted into a manageable format for feeding the learning algorithms.

Figure 20 – General overview of the proposed methodology.



Source: The author (2022).

Next, the classifiers were developed by fine-tuning pre-trained BERT model with the extracted data to perform three classification tasks: i) identification of possible consequences, given an occurrence of a leakage; ii) classification of the severity of the consequences; iii) classification of the likelihood of occurrence of the accidental scenario. Each model was trained with a specific annotated corpus that was built from the PrHA sheets. Indeed, the corpus contains the data extracted from PrHA documents (e.g., Table 3) and the target related to its corresponding task. Below, we describe these steps in more details.

5.1.1 Text Extraction

The first script accesses each PrHA document (available as DOC files), and the textual data are extracted by searching for each header: unit, system, subsystem description, equipment, chemical product, pipeline/equipment material, temperature, pressure, flow rate, initiating event, potential consequences, likelihood, and severity. Then, all texts were automatically extracted and stored into a CSV file.

Since some headers have multiple text data (e.g., initiating event, see Table 3), various instances were generated per document. Each row of the resulting CSV file corresponds to an instance, and each text associated to a header is separated by commas. Table 27 illustrates the instances resulting from Table 3 on the CSV file.

Table 27 – Data from Table 3 converted into CSV format.

Instance	**Data** (unit, system, subsystem description, chemical product, temperature, pressure, flow rate, material, equipment, initiating event, potential consequence, severity, likelihood)
1	Industrial wastewater treatment, Flow regularization system, Basin with possible presence of toxic substance hydrocarbon from another unit, Contaminated and oily water, 25, 1.033 3000, Carbon steel, Sump pump, Small leakage, Irritation, II, D
2	Industrial wastewater treatment, Flow regularization system, Basin with possible presence of toxic substance hydrocarbon from another unit, Contaminated and oily water, 25, 1.033 3000, Carbon steel, Sump pump, Small leakage, Toxic vapour cloud, II, D
3	Industrial wastewater treatment, Flow regularization system, Basin with possible presence of toxic substance hydrocarbon from another unit, Contaminated and oily water, 25, 1.033 3000, Carbon steel, Sump pump, Large leakage, Irritation, II, A
4	Industrial wastewater treatment, Flow regularization system, Basin with possible presence of toxic substance hydrocarbon from another unit, Contaminated and oily water, 25, 1.033 3000, Carbon steel, Sump pump, Large leakage, Toxic vapour cloud, III, A

Source: The author (2022).

5.1.2 Text Organization

As Figure 20 depicts, the target depends on the task performed. For this reason, an annotated corpus (i.e., labelled dataset) was built for each of the three tasks. Thus, the second script selects the textual data to construct the input sentences and the target from the CSV file according to the header. The script also cuts out the rows of the CSV without data related to the ‘initiating event, ‘potential consequence’, ‘severity category’ or ‘likelihood category’.

Each row of the CSV file may provide multiple input sentences. For example, a set of potential consequences was specified for two initiating events (small leakage and large leakage, see Table 3) and, thus, we can build a sentence for each set. It is also possible to construct different input sentences using the potential consequences, where the output pair can be either the severity or the likelihood category. Thus, we built corpus 1 with 1,391 instances, i.e., input sentence and label pairs, for Consequence Prediction Model and dataset 2 and dataset 3 with 2,974 instances for Severity and Frequency Classifiers. Table 28 summarizes the input and output that compose the datasets and presents the number of instances of each corpus.

Table 28 – Definition of the input and output of each corpus.

Corpus	Input data	Output	Number of instances
1	Unit, system, subsystem description, chemical product, initiating event, equipment, equipment specifications, temperature, pressure and flow rate	Potential consequence	1,391
2	Unit, system, subsystem description, chemical product, initiating event, equipment, equipment specifications, temperature, pressure, flow rate, and potential consequence	Severity category	2,974
3	Unit, system, subsystem description, chemical product, initiating event, equipment, equipment specifications, temperature, pressure, flow rate, and potential consequence	Likelihood category	2,974

Source: The author (2022).

Each input sequence provided to Consequence Prediction Model characterizes a possible leakage in a specific subsystem of the oil refinery. Since two initiating events (small leakage and large leakage) were considered by the experts during the development of the PrHA, every subsystem is represented twice in this dataset. For the first task, we defined our input as a sentence constructed by joining the following text: unit, system, subsystem description, chemical product, operating conditions, equipment, equipment specifications, and initiating event. Considering the example in Table 27, one of the raw sentences i used as input for task 1, $x_{1,i}$, is given in Equation (**7**:

$$x_{1,i} = \textit{Industrial wastewater treatment Flow regularization system Basin with possible presence of toxic substance hydrocarbon from other unit Contaminated and oily water 25 1.033 3,000 Carbon steel Sump pump Small leakage} \qquad (7)$$

This sentence represents that a small leakage of petroleum might cause toxic vapour cloud and/or irritation; thus, the output is a vector that contains the combination of the potential consequences. In this example, the output $y_{1,i}$ corresponding to $x_{1,i}$ is given in Equation (**8**:

$$\mathrm{y_{1,i}} = \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{matrix} \qquad (8)$$

Note that this vector contains 7 positions, which represent the number of all potential consequences that can damage human life found in the PrHA documents. Thus, each position $y_{1,i}^{n}$, $n = 1, 2, 3, 4, 5, 6,$ or 7 (burn injury, vapour cloud explosion, flash fire, irritation, pool fire, toxic vapour cloud, or jet fire respectively), assumes the values 0 or 1, where 1 indicates the presence of the potential consequence n. These potential consequences were defined by the experts based on their knowledge, experience, and creativity. The use of textual data from PrHA may allow the model to extract and understand this knowledge. Thus, it would enable postulating an accidental scenario in an oil refinery subsystem.

For the second and third tasks, the input sentences were similarly constructed by joining the same textual data used for the first task with the addition of the potential consequence (see Table 3 and Table 28). Using the same example, two input sentences can be constructed: one by adding ‘toxic vapour cloud’ and the other by adding ‘irritation’. These input sentences are present in both corpus (i.e., for task 2 and for task 3). For example, one of these sentences used as input for task 2, $\mathrm{x}_{2,\mathrm{i}}$, and 3, $\mathrm{x}_{3,\mathrm{i}}$, is given in Equation (**9**.

$$x_{2,i} = x_{3,i} = \textit{Industrial wastewater treatment Flow regularization system Basin with possible presence of toxic substance hydrocarbon from other unit Contaminated and oily water 25 1.033 3,000 Carbon steel Sump pump Small leakage Toxic vapour cloud} \tag{9}$$

The output for the second task may be assigned to four possible values (0, 1, 2, or 3), which represent the severity categories (I to IV; see Table 1). For instance, the output of Severity Classifier for the i-th input sentence is $y_{2,i} = 1$ (II). Also, the output for the third task may be assigned to four possible values (0,1, 2, or 3), which represent the likelihood category (A to D; see Table 2), while the output of Frequency Classifier for the i-th input sentence is $y_{3,i} = 3$ (D). Thus, we created an appropriate dataset for each model.

5.1.3 Text Preprocessing

Textual data oftentimes present noise, such as different variations of capitalization for the same word, punctuation, special characters, etc. Given that, three preprocessing operations (lowercasing, noise removal and tokenization) were performed to transform the input sentences into a cleaner format that can help improve the learning process of the models. The lowercasing and noise removal were implemented in Python using regular expression operations and Pandas library (MCKINNEY, 2010) and the tokenization was performed using the tokenizer provided by transformers library (WOLF, DEBUT, *et al.*, 2020).

First, we converted upper-cased to lower-cased words. Lowercasing all data is simple and one of the most effective processes to solve data sparsity issues, and it should be applied to improve accuracy for all languages and domains (UYSAL, GUNAL, 2014). Next, noise removal includes deleting special characters, white spaces, and punctuation. In the input sentences, several special characters were present (e.g., ‘-’, ‘/’, ‘%’, ‘#’). This step also includes the expansion of abbreviations (e.g., ‘adu’ was converted to ‘atmospheric distillation unit’). Thus, noise removal was paramount to construct cleaner sentences. To mark the beginning and the end of each sentence, it was necessary to add [CLS] and [SEP] respectively; this is because BERT was pre-trained using the format [CLS] sentence [SEP].

To fine-tune a pre-trained model, it is necessary to apply the same tokenization applied during pre-training; thus, we used the ‘AutoTokenizer’ backed by transformers library, which splits the sentences into a sequence of tokens according to punctuation and word pieces (i.e., sub-word units), converts raw text to sparse index encodings, and stores the vocabulary token-to-index map. Thus, the cleaned sentences were processed by the tokenizer, and then Table 29 presents some examples. In addition, the tokenizer transforms all sequences to a maximum length (512 tokens) by adding zeros, since the model requires inputs that have the same shape and size.

Table 29 – Examples of tokenized sentences.

Corpus	Tokenized sentence
1	['[CLS]', 'industrial', 'waste', '##water', 'treatment', 'flow', 'regularization', 'system', 'basin', 'with', 'possible', 'presence' 'of', 'toxic', 'substances, 'hydrocarbon', 'from', 'other', 'unit', 'contaminated', 'and', 'oily', 'water', 'sump', 'pump', 'leak', '##age', '[SEP]']
2 and 3	['[CLS]', 'industrial', 'waste', '##water', 'treatment', 'flow', 'regularization', 'system', 'basin', 'with', 'possible', 'presence' 'of', 'toxic', 'substances, 'hydrocarbon', 'from', 'other', 'unit', 'contaminated', 'and', 'oily', 'water', 'sump', 'pump', 'leak', '##age', 'toxic', 'vapour', 'cloud', '[SEP]']

Source: The author (2022).

5.1.4 Modeling Process

We used the Pytorch implementation of pre-trained BERT available at transformers library (WOLF, DEBUT, *et al.*, 2020). We here adopted the ‘bert-base-multilingual-cased’ version of the ‘BertForSequenceClassification’ model. Thus, fine-tune them on our own datasets described in Section 5.1.2.

As mentioned, we modified the output layer of the pre-trained model to adapt it for performing a classification task. More specifically, we added three different output layers for each model because Consequence Prediction Model aims at predicting multiple classes that are not mutually exclusive, while Models 2 and 3 classify the input sentence into a single class among mutually exclusive classes. Then, we used a $sigmoid$ activation function for Consequence Prediction Model, and a $softmax$ for Models 2 and 3 to predict the probability of each label c. Simply put, the $sigmoid$ function returns a value in the range 0 to 1 for each label (i.e., independent probabilities); thus, the predicted labels are the ones, which have probability greater than 0.5. In turn, the $softmax$ activation function outputs are mutually exclusive, and then the sum of their probabilities is 1; thus, the predicted class is the one, which has the highest probability (FARHADI, NIA, *et al.*, 2019, GAO, Bolin, PAVEL, 2017). Consequence Prediction Model was fitted using the binary cross-entropy loss function (to penalize each output independently), while Models 2 and 3 adopted the categorical cross-entropy (further details can be found in (GOODFELLOW, BENGIO, *et al.*, 2016).

We fine-tuned the pre-trained model three times, each time using a specific dataset for a given classification task. Thus, using the GeForce RTX 2080 Ti, the fine-tuning of each model took about four/five hours. We adopted ‘BertAdam’ as optimizer, available on transformers library, and a learning rate of 10^{-6}, batch size of 16, and warm up of 0.1 to train all models.

Finally, each dataset was randomly split into 90% (10% of it were adopted for validation) for training and the remaining 10% for test (unseen data). Finally, we evaluated the model’s performance on test data. The results achieved with each model are discussed in the following section.

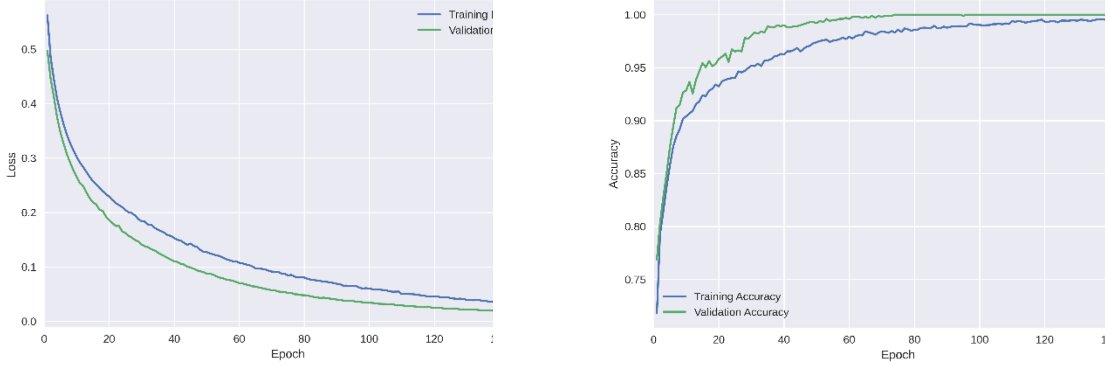
5.2 RESULTS AND DISCUSSION

Some results discussed on this chapter have been presented at ESREL (MACÊDO, AICHELE, *et al.*, 2020a) and ABRISCO (MACÊDO, MOURA, AICHELE, *et al.*, 2021) and published in Process Safety and Environmental Protection (MACÊDO, MOURA, AICHELE, *et al.*, 2022).

5.2.1 Consequence Prediction Model

To evaluate the learning and generalization of Consequence Prediction Model, the train and validation learning curves are presented in Figure 21a, and the accuracy (i.e., proportion of true positives and true negatives among the total number of observations) graphs are given in Figure 21b. The training and validation accuracies were remarkably high (99.97% and 99.9% respectively), which indicate a good fit of the learning algorithm. Then, we stopped the training after 150 epochs to avoid overfitting.

Figure 21 – a) Training and validation optimization learning curves for Consequence Prediction Model; b) Training and validation accuracy learning curves for Consequence Prediction Model.



a b

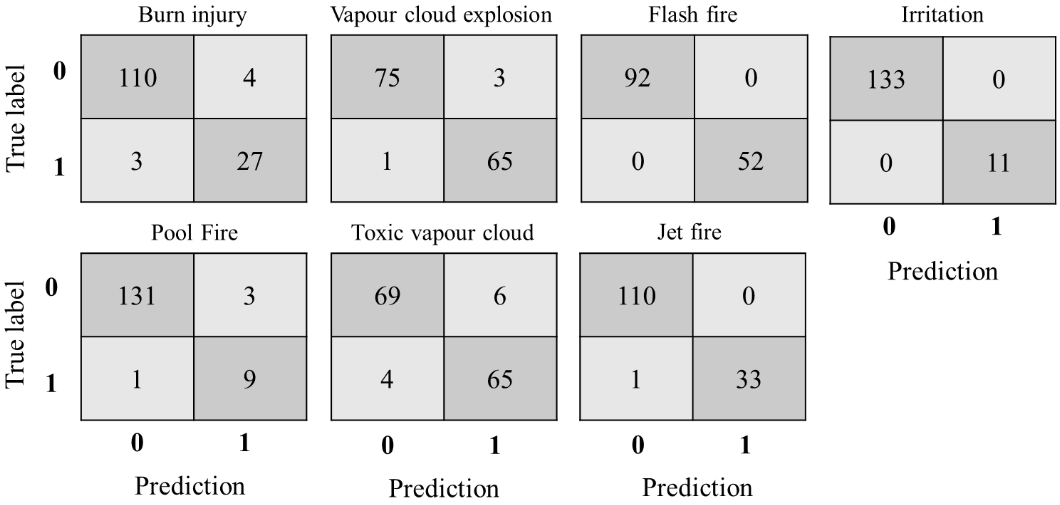
Source: The author (2022).

As explained, Consequence Prediction Model performs a multi-classification in which an instance can be assigned to different potential consequences simultaneously; thus, the output is a vector with 7 dimensions, which represent potential consequences, and the value assumed in each dimension is binary, indicating whether the example contains that consequence or not.

Figure 22 provides the confusion matrices with the prediction on test data for each consequence to evaluate the performance of Consequence Prediction Model. In the confusion matrices, the element in the r-th row and j-th column, $c_{r,j}$, indicates the number of observations j predicted as r. For example, $c_{0,0}^{(burn\,injury)}$ (element of the confusion matrix for burn injury) indicates that 110 of the instances that are not labelled as burn injury (0) were corrected classified as 0.

Indeed, Consequence Prediction Model achieved a mean accuracy of 97.42% to predict the potential consequences of test samples. From the confusion matrices, we computed precision, recall, and F_1-score for each class, where precision is the number of true positives (instances correctly predicted as 1) over all positive predictions (all instances predicted as 1), recall is the number of true positives over all instances with 1 as true label, and F_1-score is the harmonic mean between precision and recall.

Figure 22 – Confusion matrices for Consequence Prediction Model’s classification of test data



Burn injury

True label \ Prediction	0	1
0	110	4
1	3	27

Vapour cloud explosion

True label \ Prediction	0	1
0	75	3
1	1	65

Flash fire

True label \ Prediction	0	1
0	92	0
1	0	52

Irritation

True label \ Prediction	0	1
0	133	0
1	0	11

Pool Fire

True label \ Prediction	0	1
0	131	3
1	1	9

Toxic vapour cloud

True label \ Prediction	0	1
0	69	6
1	4	65

Jet fire

True label \ Prediction	0	1
0	110	0
1	1	33

Source: The author (2022).

Table 30 summarizes the scores for each category. One can see the inferior precision for pool fire (75%) in comparison to others. This can be explained by the relatively high number of false positives. In the early stages of QRA, the aim is to identify all the possible hazardous scenarios; thus, this type of error is deemed to be acceptable since the risk analyst may assess the coherence of the model’ results.

Table 30 – Results of the classification of test data for each potential consequence.

n	Potential consequence	Precision (%)	Recall (%)	F_1-score (%)
1	Burn injury	87.09	90	88.52
2	Vapour cloud explosion	95.59	98.48	97.01
3	Flash fire	100	100	100
4	Irritation	100	100	100
5	Pool fire	75	90	81.82
6	Toxic vapour cloud	91.55	94.20	92.75
7	Jet Fire	100	97.06	98.51

Source: The author (2022).

Moreover, Consequence Prediction Model yielded more false positives for burn injury and for toxic vapour cloud than for pool fire. However, the precision related to the prediction of these consequences were less affected by these. Consequence Prediction Model also yielded more false negatives for burn injury and toxic vapour cloud than for the other consequences. Nevertheless, the recall for burn injury and toxic vapour cloud is 90% and 94.2% respectively. Thus, Consequence Prediction

Model presented satisfactory results considering all potential consequences and achieved a mean F_1-score above 94.09%.

We also evaluated the model's performance to correctly predict the whole list of potential consequences of a given instance. This means that if one or more labels of a sample were misclassified, we considered the prediction as a model error. Then, Table 31 presents all misclassifications, the actual target, and a brief description of the error; 85.41% of the test data were assigned with the correct set of labels.

Table 31 – Wrong predictions made by Consequence Prediction Model.

	Error			
# of errors	**Type**	**Description**	**Prediction**	**Target**
1	False positive	Predicted toxic vapour cloud	0 0 0 0 0 1 1	0 0 0 0 0 0 1
		Predicted toxic vapour cloud	0 1 1 0 0 1 1	0 1 1 0 0 0 1
		Predicted vapour cloud explosion	0 1 0 0 1 1 0	0 0 0 0 1 1 0
		Predicted burn injury	1 1 1 0 0 1 0	0 1 1 0 0 1 0
		Predicted toxic vapour cloud	0 0 1 1 0 1 0	0 0 1 1 0 0 0
		Predicted burn injury	1 1 0 0 0 0 0	0 1 0 0 0 0 0
		Predicted toxic vapour cloud	1 1 0 0 0 1 0	1 1 0 0 0 0 0
		Predicted vapour cloud explosion	0 1 0 0 1 1 0	0 0 0 0 1 1 0
	False negative	Did not predict burn injury	0 1 1 0 0 1 0	1 1 1 0 0 1 0
		Did not predict toxic vapour cloud	0 0 1 1 0 0 0	0 0 1 1 0 1 0
		Did not predict vapour cloud explosion	0 0 0 0 0 1 0	0 1 0 0 0 1 0
		Did not predict toxic vapour cloud	1 1 0 0 0 0 0	1 1 0 0 0 1 0
		Did not predict pool fire	1 1 0 0 0 0 0	1 1 0 0 1 0 0
		Did not predict burn injury	0 1 0 0 0 0 0	1 1 0 0 0 0 0
		Did not predict jet fire	0 1 0 0 0 0 0	0 1 0 0 0 0 1
		Did not predict toxic vapour cloud	0 0 1 1 0 0 0	0 0 1 1 0 1 0
2	False positive	Predicted burn injury and pool fire	1 1 1 0 1 1 0	0 1 1 0 0 1 0
		Predicted burn injury and toxic vapour cloud	1 1 0 0 0 1 0	0 1 0 0 0 0 0
	False negative	Did not predict pool fire and toxic vapour cloud	1 1 0 0 1 1 0	1 1 0 0 0 0 0
	False negative and false positive	Did not predict burn injury; predicted pool fire	0 1 0 0 1 0 0	1 1 0 0 0 0 0
		Did not predict toxic vapour cloud; predicted vapour cloud explosion	0 1 0 0 1 0 0	0 0 0 0 1 1 0

Source: The author (2022).

Note that 21 out of 144 instances on test set were misclassified, 16 out of 21 instances had one incorrect label and 5 instances had two incorrect labels. However, in only 11 instances, there were potential consequences unpredicted; in these cases, experts should review the results and manually include them. It is noteworthy that most unpredicted potential consequences were burn injury and toxic vapour cloud.

A possible explanation for the inferior performance to predict some potential consequences might be the presence of similar scenarios in the PrHA documents that cannot led to such consequences. Table 32 shows some examples of input (sentences showed without preprocessing and tokenization to facilitate de analysis), output, and the prediction made by Consequence Prediction Model, $x_{1,i}$,$y_{1,i}$, and $\hat{y}_{1,i}$, respectively.

Table 32 – Assessing scenarios provided by Model and the potential consequences predicted.

i	$x_{1,i}$	$y_{1,i}$	$\hat{y}_{1,i}$	**Error**
1	**naphtha hydrotreating unit** reactor **system hydrogen and hydrogen sulfide section between exchanger** and **recycling gas injection** point naphtha and **naphtha stream** 57 200 91,461 **carbon steel** small leakage	1 1 0 0 0 0 0	0 1 0 0 0 0 0	Did not predict burn injury
2	**naphtha hydrotreating unit** cooling and separation system **hydrogen and hydrogen sulfide section between the** separation point of the output stream from the stabilization vessel and the **naphtha stream** for **recycling exchanger** 31 260 82,943 **carbon steel** large leakage	0 1 0 0 0 0 0	0 1 0 0 0 0 0	-
3	**naphtha hydrotreating unit** cooling and separation system **hydrogen and hydrogen sulfide section between exchanger** and condenser 33 300 20,1291 **carbon steel** large leakage	1 1 0 0 1 0 0	1 1 0 0 0 0 0	Did not predict pool fire

Source: The author (2022).

For instance, there are several similarities in the description of instances 1 and 2 (in bold); however, instance 2 does not generate burn injury despite the higher operating temperature and it was correctly classified by Consequence Prediction Model. Also, among 635 accidental scenarios related to naphtha hydrotreating unit only 8 of them can cause burn injury. Moreover, instances 2 and 3 even involve the same system (cooling and separation system), equipment (exchanger) and similar operating conditions; however, differently from instance 3, the scenario described in instance 2 cannot led to pool fire.

In addition, it is worth mentioning that burn injury, toxic vapour cloud, and pool fire (for which the model had the worst recall) are related to a wide variety of scenarios, which can make it difficult for it to learn/recognize all features that characterize these potential consequences. For instance, considering our database, pool fire and toxic vapour cloud occur in more than 15 different units of the oil refinery (i.e., almost all units in the refinery) and they are associated with the release of more than 100 chemical products, whereas jet fire occurs in only 7 units, due to the outflow of 50 substances.

How to improve the performance of the model will be object of future research. A possible solution to overcome this problem may be to perform DA to build a more homogeneous dataset in relation to the different features, such as unit and/or chemical (LIU, Sisi, LEE, *et al.*, 2020). It is worth mentioning that the model's outcomes represent a starting point for completing the QRA. Thus, analysts should critically evaluate these results in order to add/remove scenarios, and then build a more representative database. Thus, the updated database could be used to retrain the models as a way to improve their performances and system representativeness.

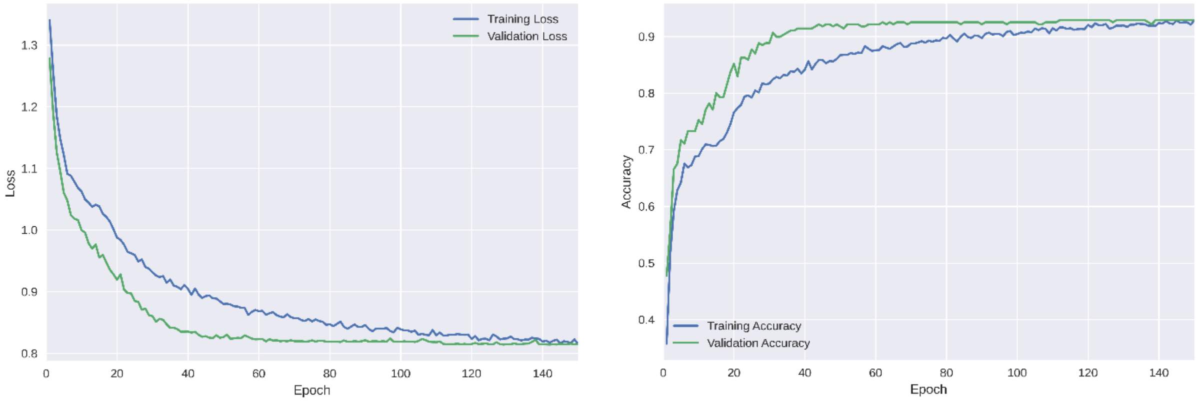
Overall, Consequence Prediction Model provides satisfactory predictions of the potential consequences for different subsystems of the oil refinery. The model predictions could be used as a starting point for RA by providing an initial set of hypothetical accidental scenarios and its severity and frequency categories. The team of risk analysts could evaluate whether the scenarios are coherent and could postulate new potential consequences. Thus, the trained models may support experts and consequently reduce time and human efforts to perform QRA.

5.2.2 Severity Classifier

The sentences (instances) provided to Severity Classifier are labelled with the severity of the potential consequences (see Table 1). The optimization and the accuracy learning curves for Severity Classifier are shown in Figure 23a and Figure 23b respectively. Severity Classifier was trained for 150 epochs until its loss curves reached some stability. The accuracy curves reached 92.83% on training and 92.92% on validation. Again, the proximity of the training and validation curves and the behavior of the learning plots indicate a good fit of the model.

Table 24 shows the confusion matrix on the test data. The model achieved an accuracy of 86.44% to classify the test data.The lowest precision was for the prediction of category I (75.86%) and III (78.35%). These outcomes for category I can be explained by the small number of instances; thus, the instances II misclassified as I had a big impact on the precision (note that the recall for category II was less affected by these errors). Yet, Severity Classifier predicted correctly 95.65% of the instances classified as I.

Figure 23 – a) Training and validation optimization learning curves for Severity Classifier; b) Training and validation performance learning curves for Severity Classifier.

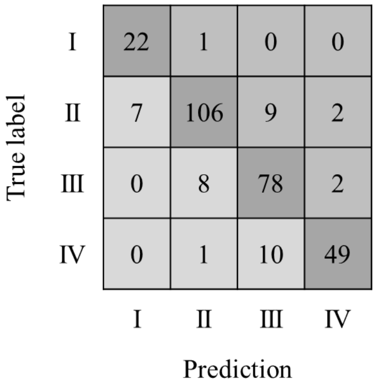


a

b

Source: The author (2022).

Figure 24 – Confusion matrix for Severity Classifier's classifications of test data.



True label	I	II	III	IV
I	22	1	0	0
II	7	106	9	2
III	0	8	78	2
IV	0	1	10	49
	Prediction			

Source: The author (2022).

Moreover, the worst recall was computed for category IV, where 10 out of 11 misclassifications were predicted as III. Nevertheless, the results obtained were satisfactory. Indeed, F_1-scores were above 80% for all categories. Table 33 summarizes the results for each category.

Table 33 – Results of the classification of test data for each severity category.

Severity	Precision (%)	Recall (%)	F_1-score (%)
I	75.86	95.65	84.61
II	92.17	85.48	88.69
III	80.41	88.63	81.28
IV	92.45	81.67	84.21

Source: The author (2022).

A possible explanation for the inferior performance to predict category III might be the presence of similar scenarios descriptions in the PrHA documents, which are classified with different severity levels. For instance, Table 34 shows some examples of input (sentences showed without preprocessing and tokenization to facilitate the analysis) and output pairs $(x_{2,i}, y_{2,i})$ used to train Severity Classifier and misclassified scenarios (i.e., prediction $\hat{y}_{2,i}$ different from $y_{2,i}$). Note that the instances without predictions correspond to the training instances.

Table 34 – Samples of input and output pairs in the data set.

i	$x_{2,i}$	$y_{2,i}$	$\hat{y}_{2,i}$
1	Hydrotreating unit loading and unloading of chemicals system dmds from **container** to **pump and from pump to unit flange** 1.33 25 container tank large leakage toxic vapour cloud	III	**-**
2	Hydrotreating unit loading and unloading of chemicals system dmds from **truck** to the **loading area and container in the waiting area** 1.33 25 container tank large leakage toxic vapour cloud	II	III
3	Powerhouse steam system medium pressure steam from **boiler** to the **medium pressure steam collection and distribution header** 17.5 260 **64.73** large leakage burn injury	II	-
4	Powerhouse steam system medium pressure steam from **medium pressure steam collection and distribution header** to the **battery limit** 17.5 260 **128.92** large leakage burn injury	III	II
5	Water treatment unit chemical product system gaseous chlorine from chlorine cylinders **to the chlorination system in the chlorinator house** carbon steel 25 **0 30** large leakage toxic vapour cloud	III	-
6	Water treatment unit chemical product system gaseous chlorine chlorine cylinders **inside the cylinder room** carbon steel 25 **21.1 40** large leakage toxic vapour cloud	IV	III

Source: The author (2022).

As we can see, $x_{2,1}$ (training input) and $x_{2,2}$ (test input) are very similar, both represent the transport of dmds (dimethyl disulphide) in the hydrotreating unit under similar operating conditions; then, it is reasonable that Severity Classifier classified $x_{2,2}$ as III, i.e., the same category as $x_{2,1}$. However, the system described in $x_{2,1}$ starts at the container tank and goes to pump; while the system described in $x_{2,2}$ starts at the truck and goes to the container tank. Likewise, Severity Classifier misclassified $x_{2,4}$ as II. Both $x_{2,3}$ and $x_{2,4}$ represent a burn injury due to large leakage of a pipeline containing medium pressure steam in the powerhouse under similar operating conditions; however, the system described in $x_{2,3}$ goes from the boiler to the header, while $x_{2,4}$ considers the header up to the battery limit and involves a higher flow rate.

Moreover, most of the scenarios of the water treatment unit in our database are classified as III, such as $x_{2,5}$, which involves the large leakage of pipeline from chlorine cylinders to the chlorination system. There are only three scenarios classified as IV, which may have compromised the learning process. One of them is $x_{2,6}$ (misclassified as III) that represents a toxic vapour cloud generated due to the large leakage of the chlorine cylinder. Additionally, $x_{2,5}$ and $x_{2,6}$ are also similar; both encompass the release of gaseous chlorine and the chlorine cylinder. Thus, the data provided might not be sufficient to make the model capture these subtleties between category III and others, which may have compromised the learning process. On the other hand, these errors for categories III and IV are tolerable because in practice scenarios classified as III or IV must be further analyzed quantitatively.

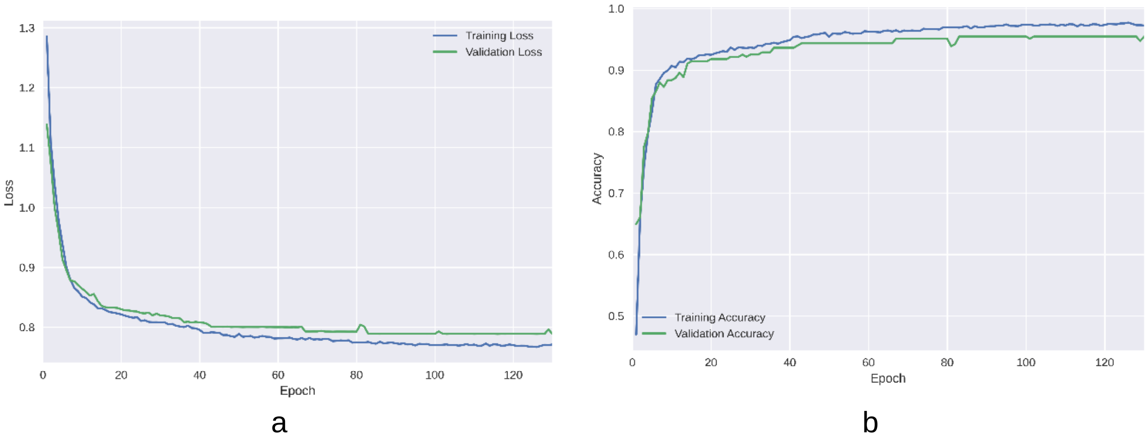
Thus, the classification provided by Severity Classifier would certainly be useful to identify the least severe scenarios, filtering the most critical cases for further quantitative analysis. Moreover, since the main purpose of QRA is to identify all accidental scenarios, underestimate scenarios are undesirable. Thus, to avoid this situation we plan to incorporate in the models' training a function to penalize non-conservative predictions. Furthermore, since we treated all variables as textual data, we will investigate ensemble models such as the combination of BERT with other ML classifiers to process the continuous variables separately, so that the model can better learn the relationship between these variables and the description of subsystems.

5.2.3 Frequency Classifier

Finally, the input sequence provided to Frequency Classifier is labelled with the likelihood category; see Table 2. The evaluation of the model training can be done through the optimization and the accuracy learning curves in Figure 25a and Figure 25b respectively. The model was trained for 130 epochs until its loss curve reached some stability and achieved an accuracy of 97.68% on training and 95.48% on validation; the curves indicate a good fit of the model. Moreover, Frequency Classifier achieved an accuracy of 94.34% on test. A detailed description of Frequency Classifier outcomes is given in

Figure 26 and the performance metrics are summarized in Table 35.

Figure 25 – a) Training and validation optimization learning curves for Frequency Classifier; b) Training and validation accuracy curves for Frequency Classifier.



Source: The author (2022).

Figure 26 – Confusion matrix of Frequency Classifier's classifications of test data.

True label	A	B	C	D
A	25	8	0	0
B	3	112	0	0
C	0	1	56	4
D	0	0	0	89

Prediction

Source: The author (2022).

Table 35 – Results of the classification of test data for each likelihood category.

Likelihood	**Precision (%)**	**Recall (%)**	F_1**-score (%)**
A	89.28	75.76	81.97
B	92.56	97.39	94.91
C	100	91.80	95.72
D	95.7	100	97.8

Source: The author (2022).

Frequency Classifier predicted all classes with great precision. The worst performance was in the prediction of category A, which represents the least frequent events (Table 2) that are indeed more difficult to envision and usually leads to more uncertain estimates (JIN, WANG, *et al.*, 2020, MARCHIORI, GUIDA, 2015). In fact, 24.24% of the instances of category A were misclassified as B. These errors have a greater impact on the metrics for category A, since it corresponds to the smallest group on test. Note that these errors do not have a major impact on the metrics for category B. Moreover, one interesting finding is that most of the model's errors were in predicting instances into categories that represent more likely events. These errors may lead to more critical risk classification of the scenarios (ISO, 2018). Finally, all performance metrics for category B, C, and D were above 90%. These results of Frequency Classifier suggest a reasonable ability to learn and recognize patterns about all likelihood categories.

5.2.4 Concatenation of Errors

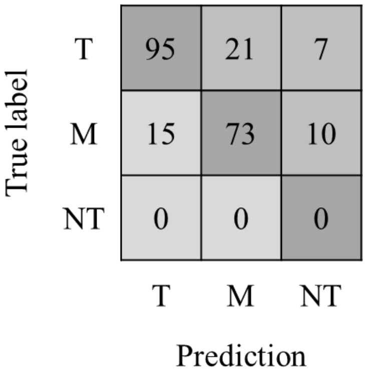
To analyze the concatenation of errors, we combined the predictions made by Severity Classifier and 3 on test data. To that end, we considered the risk matrix (Table 36), according to which risks are classified as Tolerable (T), Moderate (M), or Non-Tolerable (NT) (ISO, 2018). The risk categories provided by the models are summarized in the confusion matrix (Figure 27). One can see that more than 17% of the test samples were assigned into a more critical risk category; thus, the results provided are more conservative.

Table 36 – Risk matrix.

Risk Matrix				
Consequence	**Likelihood**			
	A	**B**	**C**	**D**
IV	M	M	M	NT
III	T	M	M	M
II	T	T	M	M
I	T	T	T	T

Source: The author (2022).

Figure 27 – Confusion matrix with the result on test data of the combination of Severity and Frequency Classifiers.



True label \ Prediction	T	M	NT
T	95	21	7
M	15	73	10
NT	0	0	0

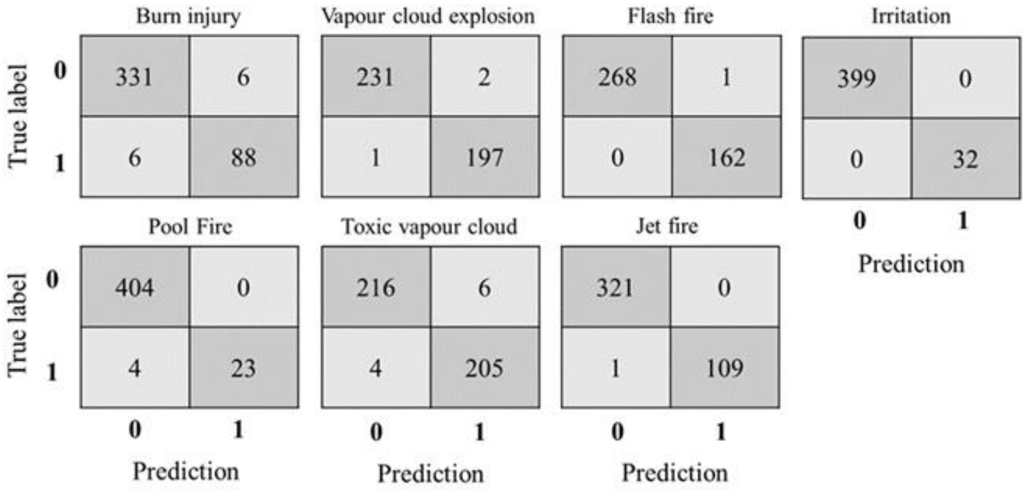
Source: The author (2022).

5.2.5 Sensitivity Analysis

In order to further evaluate the performance of the algorithms we retrained the models using 70% of the dataset; then, we evaluated their performances on the remaining data. Figure 28 provides the confusion matrices with the prediction on test data for each consequence to evaluate the performance of Consequence Prediction Model.

From the confusion matrices, we computed the performance metrics summarized in Table 37. Comparing Table 30 and Table 37, one can see that the performance of Consequence Prediction Model improved and the false negative rate dropped for all potential consequences. This result is very positive, since the main purpose of the early stages of QRA is to predict all possible consequences. This result is very positive, since the main purpose of the early stages of QRA is to predict all possible consequences.

Figure 28 – Confusion matrices for Consequence Prediction Model's classification of test data using 70/30 split.



Source: The author (2022).

Table 37 – Results of the classification of test data using 70/30 split for each potential consequence.

n	Potential consequence	Precision (%)	Recall (%)	F_1-score (%)
1	Burn injury	93.62	93.62	93.62
2	Vapour cloud explosion	99.49	98.99	99.24
3	Flash fire	99.39	100	99.69
4	Irritation	100	100	100
5	Pool fire	85.19	100	92.00
6	Toxic vapour cloud	97.16	98.08	97.62
7	Jet Fire	100	99.09	99.54

Source: The author (2022).

Moreover, with the increase from 144 to 431 test instances, Consequence Prediction Model made only 8 more errors considering the assignment of the correct set of potential consequences, i.e., 29 out of 431 instances on test set were misclassified (Table 38). Among them, 27 instances had one incorrect label and 2 instances had two incorrect labels. As mentioned, the main purpose of the early stages of QRA is to identify all possible hazards; thus, only 16 errors (unpredicted consequences) are critical regarding the following steps of QRA. Yet, the most unpredicted was burn injury (6 out of 16), which is not quantitively evaluated in the following steps of QRA, since it is not often related to casualties.

Table 38 – Wrong predictions made by Consequence Prediction Model using 70/30 split.

	Error			
# of errors	**Type**	**Description**	**Prediction**	**Target**
1	False positive	Predicted toxic vapour cloud	1 1 0 0 0 1 0	1 1 0 0 0 0 0
		Predicted toxic vapour cloud	0 0 0 0 0 1 1	0 0 0 0 0 0 1
		Predicted vapor cloud explosion	0 0 1 0 1 1 0	0 0 0 0 1 1 0
		Predicted toxic vapour cloud	0 0 1 1 0 1 0	0 0 1 1 0 0 0
		Predicted toxic vapour cloud	0 1 1 0 0 1 1	0 1 1 0 0 0 1
		Predicted toxic vapour cloud	0 0 1 1 0 1 0	0 0 1 1 0 0 0
		Predicted burn injury	1 1 1 0 0 1 0	0 1 1 0 0 1 0
		Predicted burn injury	1 1 0 0 0 0 0	0 1 0 0 0 0 0
		Predicted burn injury	1 1 1 0 0 1 0	0 1 1 0 0 1 0
		Predicted vapour cloud explosion	0 1 0 0 1 1 0	0 0 0 0 1 1 0
		Predicted burn injury	1 1 1 0 0 1 0	0 1 1 0 0 1 0
		Predicted burn injury	1 1 0 0 0 1 0	0 1 0 0 0 1 0
	False negative	Did not predict pool fire	1 1 0 0 0 1 0	1 1 0 0 1 1 0
		Did not predict toxic vapour cloud	1 1 0 0 0 0 0	1 1 0 0 0 1 0
		Did not predict burn injury	0 1 1 0 0 0 0	1 1 1 0 0 0 0
		Did not predict burn injury	0 1 0 0 0 1 0	1 1 0 0 0 1 0
		Did not predict burn injury	0 1 0 0 0 0 0	1 1 0 0 0 0 0
		Did not predict toxic vapour cloud	0 0 1 1 0 0 0	0 0 1 1 0 1 0
		Did not predict vapour cloud explosion	0 0 0 0 0 1 0	0 1 0 0 0 1 0
		Did not predict pool fire	0 0 1 0 0 0 0	0 0 1 0 1 0 0
		Did not predict vapour cloud explosion	0 0 1 0 0 0 0	0 1 1 0 0 0 0
		Did not predict pool fire	1 1 1 0 0 0 0	1 1 1 0 1 0 0
		Did not predict toxic vapour cloud	0 0 0 0 1 0 0	0 0 0 0 1 1 0
		Did not predict burn injury	0 1 0 0 0 0 0	1 1 0 0 0 0 0
		Did not predict toxic vapour cloud	1 1 1 0 0 0 0	1 1 1 0 0 1 0
		Did not predict jet fire	0 1 0 0 0 0 0	0 1 0 0 0 0 1
		Did not predict pool fire	1 1 0 0 0 0 0	1 1 0 0 1 0 0
2	False negative and false positive	Did not predict burn injury; predicted toxic vapour cloud	0 1 0 0 0 1 0	1 1 0 0 0 0 0
	False positive	Predicted burn injury and toxic vapour cloud	1 1 0 0 0 1 0	0 1 0 0 0 0 0

Source: The author (2022).

Regarding the performance of Severity Classifier, some performance metrics improved (see Table 33 and Table 39); however, more than 12% of the instances III were classified as a less severe category. As we mentioned, in practice, scenarios classified as III and IV must be further analyzed in the QRA, while the less severe scenarios are screened out from the analysis. Thus, we believe that such errors made by Severity Classifier with the 70/30 split would result in the removal of many scenarios from the following stages of a QRA, diminishing the usefulness of the proposed methodology.

Table 39 – Performance metrics for Severity Classifier using 70/30 split.

Severity	Precision (%)	Recall (%)	F_1-score (%)
I	80.39	94.25	86.77
II	88.18	86.87	87.52
III	86.50	85.87	86.18
IV	94.12	90.26	92.15

Source: The author (2022).

Moreover, for Frequency Classifier, the performances for classifying the likelihood of occurrence with the 70/30 split resulted in worse performances (see Table 35 and Table 40). This can be explained because we have an unbalanced dataset; then, 70% can represent a small amount of training data considering some classes. For instance, there are 330 instances labeled as A, while there are 1150 instances of category B.

Table 40 – Performance metrics for Frequency Classifier using 70/30 split.

Likelihood	Precision (%)	Recall (%)	F_1-score (%)
A	93.24	67.65	78.41
B	90.82	96.74	93.48
C	82.80	81.25	82.02
D	88.15	90.50	89.31

Source: The author (2022).

5.2.6 Conclusion

Consequence Prediction Model presented a great performance both in the individual prediction of each potential consequence, as well as in the prediction of the set of consequences associated with the different subsystems. Severity Classifier also showed satisfactory results, since only the precision to classify the severity level III was less than 80%. In addition, part of the contribution to decrease the precision for this category is due to the classification of less severe instances (I and II) as III; thus, the experts can focus their efforts on the assessment of the most severe accident scenarios. Finally, most of Frequency Classifier's errors were more conservative (i.e., predicted an instance into a more likely category). Moreover, the Frequency Classifier presented promising results, achieving high performance in the prediction of most likelihood categories.

Thus, TM and NLP can be adopted to support risk analysts in identifying the potential consequences of different scenarios, related to 'loss of containment' of hazardous material, and to describe qualitatively risks in terms of expected likelihood and severity of consequences. In addition, we believe that the predictions provided by the models would also allow the experts to correct the PrHAs. Experts can critically evaluate scenarios that have been misclassified by the trained models and decide whether the model's prediction makes sense, i.e., indicating an error made by experts during the PrHA.

5.3 EXTRACTING FEATURES OF RARE RISK EVENTS

Some results discussed on this section have been presented at ESREL (MACÊDO, MOURA, LINS, *et al.*, 2022). It is important to keep in mind that step 2 of QRA (see Figure 2) may be quite challenging when dealing with rare accident events with extreme consequences, since limited knowledge exists for these events (SPADA, BURGHERR, *et al.*, 2019, ZIO, AVEN, 2013). On the other hand, developing techniques to identify features about catastrophic events is quite useful for QRA.

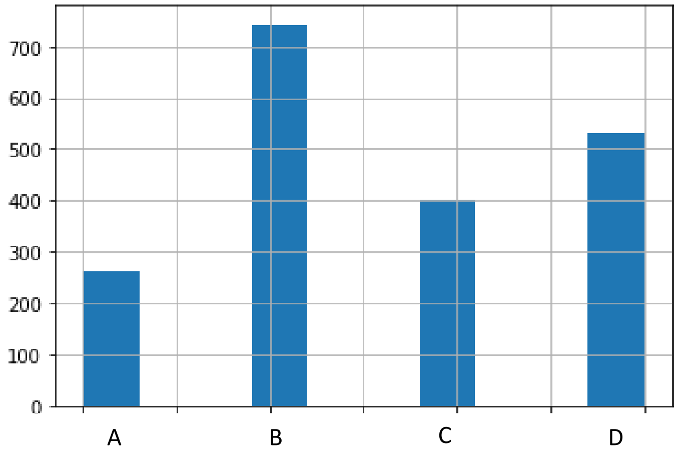
Here, we investigate how to handle accident datasets to improve BERT-based classifier learning about rare event. To do that we compared different DA and undersampling configurations to obtain a balanced and sufficiently large training set. The final aim of this analysis is developing capable of characterizing relevant features about rare accidental scenarios in support to HAZOP and PrHA.

As mentioned, pre-trained language models have been directly fine-tuned, entirely removing the need for task-specific architectures. In addition, the substantial increase in the transformer language models, from 100 million to 17 billion parameters, has brought improvements in several downstream NLP tasks. However, to achieve strong performance on a desired task typically requires fine-tuning on a dataset of thousands to hundreds of thousands of examples specific to that task (BROWN, MANN, *et al.*, 2020). Thus, a challenge arises when we are interested in rare events, since only limited information is available. For instance, some accidents are postulated as possible in principle, but there are no historical occurrence records.

As described in Section 2.1, each document contains the description of potential accident events from different processing units of an oil refinery and their qualitative assessment of frequency of occurrence and severity of consequences. The expected frequency of the events is classified into four categories (Table 2) according to (ISO, 2018).

As one can see in Figure 29, the labelled dataset is quite unbalanced. There are less than 300 potential accidents classified as Remote, representing under 15% of the dataset. As pointed by Brown et al. (2020), the difficult to build labelled datasets may limit the applicability of NLP to perform text classification tasks. Indeed, this dataset was used in to build a classifier able to predict the expected frequency (Section 5.2.3), and the worst performance of the NLP classifiers was to categorize remote events (i.e., labelled as A).

Figure 29 – Number of instances per likelihood category.



Source: The author (2022).

DA is a useful tool to improve the performance of a model; however, in NLP datasets, DA is difficult due to the high complexity of language (WEI, ZOU, 2020). Thus, we explored different DA configurations to balance our dataset in order to improve the performance of the NLP model in classifying remote events.

The approach adopted consists in using contextualized word embeddings to generate a vector under a different context. DA was implemented using *nlpaug*, a library dedicated to textual augmentation in ML experiments (MA, 2021). More specifically, the *ContextualWordEmbsAug* function, designed to perform insertion and substitution, was adopted to generate new samples based on randomly selected input sentences, maintaining the original meaning. Simply put, the *ContextualWordEmbsAug* function performs substitution by searching the most suitable word using the surrounding words as a feature to predict the target word.

DA was performed on the training data (80% of the dataset) for the classes with less samples (A and C) in order to balance the number of samples. In addition, we assumed that categories B and A have characteristics in common, which may hinder the model's ability to separate these events. Also, the high number of B instances can lead to a faster overfitting for this class. Thus, undersampling was performed to reduce the number of samples from major category (B), since we are interest in extracting features of rare events that presents few samples available. The following configurations were adopted:

i. Generate 200 training samples labelled as A;
ii. Generate 200 training samples labelled as A and 100 labelled as C;
iii. Undersampling B to reduce to 399 samples;
iv. Removing all instances labelled as B;
v. Generate 100 training samples labelled as A and remove all instances labelled as B.

Then, the training sets obtained according to each configuration are used to fine-tune BERT model resulting on different classifiers. The impact of each configuration on the classifiers' performance is evaluated by comparing the results on the test set to the baseline result (Table 41), which is brought in as the performance on test data of a classifier obtained by fine-tuning of BERT using the original training set.

Table 41 – Baseline results; performance on test data of the classifier using the original training set.

Likelihood	**Recall**	**Precision**	F_1**-score**
A	70.67	81.54	75.72
B	88.89	86.02	87.43
C	90.10	79.82	84.95
D	85.04	91.53	88.19

Source: The author (2022).

Moreover, all classifiers were trained for 25 epochs, with batch size of 16 and learning rate of 5×10^{-5}. The results and discussion of all experiments are presented in the next section.

5.3.1 Results

The performance on test data of the classifier obtained using configuration (i) (200 augmented samples labelled as A) is presented in Table 42.

Table 42 – Performance of the classifier using the training set obtained with configuration (i).

Likelihood	Recall	Precision	F_1-score
A	67.53	75.36	71.23
B	88.32	85.25	86.76
C	82.76	88.89	85.72
D	95.08	89.92	92.43

Source: The author (2022).

Overall, comparing Table 41 to Table 42, the performance improved considering categories C and D. There was a 10% increase on the recall for class D and the precision for class C. However, as one can see, the performances for category A and B decreased. This result may indicate that the description of accident scenarios related to A and B are similar. Thus, simply augmenting the data related to remote events is not enough to tackle this issue, because it makes the classifier label scenarios B as A.

Table 43 shows the classifier's performance on test data with configuration (ii). One interesting result was the increase of the recall metric regarding classes B and D. This result may indicate that configuration (ii) improved the features extracted for accident scenarios B and D, since the number of instances from these classes that were misclassified decreased.

Table 43 – Performance of the classifier using the training set obtained with configuration (ii).

Likelihood	Recall	Precision	F_1-score
A	60.26	95.92	74.02
B	94.89	83.5	88.83
C	83.91	76.84	80.22
D	89.44	91.37	90.39

Source: The author (2022).

On the other hand, although there was a significant improvement on the precision metric for class A, there was also a decrease on the recall. This is particularly undesirable since we want a model to extract valuable features about such rare events. With configuration (ii) the extracted features were worse than the baseline classifier to represent class A since more accident scenarios were misclassified into a different class.

In addition, we randomly removed some instances belonging to class B. Table 44 shows the classifier's performance on test data with configuration (iii). After under sampling class B, most metrics worsened, as expected since we are reducing the number of samples used to train the model. One can see that the model began to classify more instances as A and consequently the recall for class A increased and the precision decreased. There was also an improvement regarding the recall for class D, because the classifier did not misclassify accidents from class D as B.

Table 44 – Performance of the classifier using the training set obtained with configuration (iii).

Likelihood	Recall	Precision	F_1-score
A	73.75	56.19	63.78
B	68.98	84.87	76.10
C	82.42	77.32	79.79
D	92.68	88.37	91.00

Source: The author (2022).

Configuration (iv) was adopted considering that category B must have many characteristics in common with category A, since class B also represents rare (unlikely) events. Indeed, the results obtained with configurations (i), (ii) and (iii) showed the classifiers often have difficulties differentiating these classes. Table 45 shows the classifier's performance on test data with configuration (iv).

Table 45 – Performance of the classifier using the training set obtained with configuration (iv).

Likelihood	Recall	Precision	F_1-score
A	93.06	97.10	95.04
C	77.23	89.66	82.98
D	92.13	81.25	86.35

Source: The author (2022).

Indeed, after training the model only with common accident events (class C and D) and class A (remote events), there was a significant improvement in the performance of the model regarding A. Also, it is possible to see a slight reduction in some metrics related to the classification of C and D, due to the removal of a large number of samples; considering the F_1-score for C and D the performance decreased 2%. In addition to the removal of the samples labelled as B we generated 100 training samples for class A (configuration v). Table 46 shows the classifier's performance on test data with configuration (v).

Table 46 – Performance of the classifier using the training set obtained with configuration (v).

Likelihood	Recall	Precision	F_1-score
A	100	100	100
C	86.35	87.65	86.99
D	93.10	90.00	91.52

Source: The author (2022).

After removing the samples that has common characteristics as the remote events and performing DA, we were able to correctly predict all instances labelled as A of this specific test set. Thus, it seems promising for dealing with rare scenarios that usually are underrepresented in accident and reliability databases.

5.3.2 Conclusion

The early stages of QRA involves identifying and assess accident events. Developing techniques to identify features about catastrophic events with extreme consequences is quite useful for QRA. The application of NLP may be quite challenging when dealing with rare accident events, since limited knowledge exists for them. Moreover, in NLP datasets, DA is difficult due to language complexity. Thus, we explored different DA configurations to balance our dataset in order to improve the performance of the NLP model in classifying remote events.

The results showed that simply augmenting the training data for the remote accident class is not enough to improve the model's learning for this class. However, fine-tuning only with samples of frequent and likely events (C and D) significantly improved the performance of the model. Thus, combining this approach and performing DA for class A, configuration (v), resulted in a model capable to correctly predict all remote events, which is a promising result.

5.4 HALO (Hazard Analysis based on Language processing for Oil refineries)

Overall, the proposed methodology for identifying risk features presented satisfactory results; thus, the trained classifiers could be a useful tool to support the

QRA early qualitative stages. To that end, we developed a web app, known as HALO2, registered in the national institute of industrial property (MACÊDO, MOURA, LINS, *et al.*, 2021), and embedded the trained classifiers into the app. The app to support risk analysts identifying and assessing accidental scenarios related to the operation of oil refineries. In summary, the user (risk analyst) provides information about the system and the trained classifiers take the user's input.

Given a system described by the user, the app 1) identifies the potential consequences of accidents related to the operation of the system, classifies each accidental scenario in terms of 2) severity of the consequence and 3) likelihood of occurrence. Moreover, the app provides an overview of similar scenarios and allows risk analysts to perform an exploratory analysis through the scenarios. Yet, HALO also displays visual outputs, includes word clouds for the description of similar systems found in our database and a bar chart to illustrate the proportion of the potential consequences associated with these similar systems. These visual outputs summarize the knowledge contained in previous risk studies and allow the user (risk analyst) to gain insight into the analyzed system.

HALO can meaningfully support the early stages of QRA; instead of starting the risk study from scratch, risk analysts can use the app outcomes as a starting point to identify and qualitatively characterize the accidental scenarios. This may be useful to prioritize the most critical scenarios that should be analyzed quantitatively. For instance, in the case of new plants, where there is almost no specific information available, the experts usually rely on partially relevant risk studies performed for similar plants. Thus, the app, which contains valuable information garnered from past risk studies, allows the experts to use that entire source of knowledge to reduce the uncertainty for performing the early stages of QRA.

5.4.1 What is HALO?

HALO was developed in Python, using the Streamlit, Transformers, and Gensim libraries (REHUREK, SOJKA, 2010, WOLF, DEBUT, *et al.*, 2020). The content of the app is divided into four parts: i) What is Hazard Analysis based on Language

[2] http://nlprisk.ceerma.com/

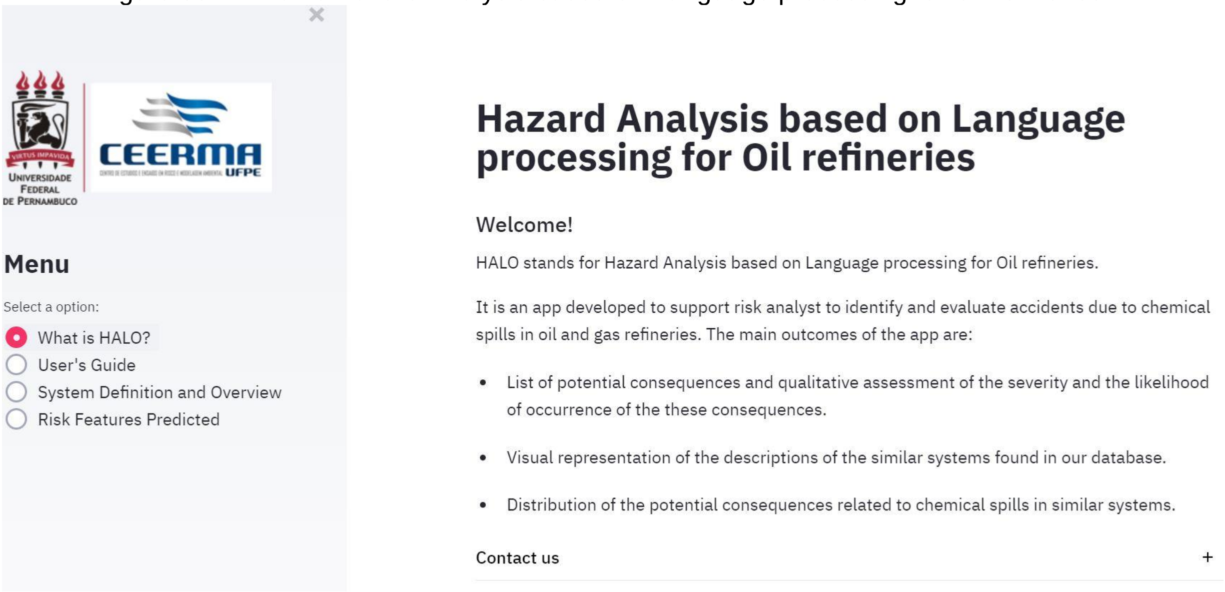
processing for Oil refineries?, ii) User's Guide, iii) System Definition and Overview, and iv) Accident Scenario Prediction. As shown in

Figure 30, there is a menu in the sidebar, displayed on the left side of the screen, with the title of each part. In this menu, the user selects the desired part. In the following figures, this menu will be omitted for a better adjustment of the images.

The first part (

Figure 30) presents general information about the app and its outcomes. The app was developed to identify different risk features and provide an overview of the hazards in oil refineries, and support risk analysts to complete the early stages of a QRA. One of the outcomes provided by the app consists of a list of potential consequences due to hypothetical chemical spills in an oil refinery system. In addition, the app qualitatively estimates the severity and the likelihood of occurrence of each predicted potential consequence. Moreover, the app presents a visual representation through word cloud for the descriptions of the similar systems found in our database. Also, the app provides a bar chart to represent the distribution of the potential consequences related to chemical spills in these similar systems.

Figure 30 – What is Hazard Analysis based on Language processing for Oil refineries?


×
VIRTUS IMPAVIDA
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CEERMA UFPE
Menu
Select a option:
● What is HALO?
○ User's Guide
○ System Definition and Overview
○ Risk Features Predicted

Hazard Analysis based on Language processing for Oil refineries

Welcome!

HALO stands for Hazard Analysis based on Language processing for Oil refineries.

It is an app developed to support risk analyst to identify and evaluate accidents due to chemical spills in oil and gas refineries. The main outcomes of the app are:

- List of potential consequences and qualitative assessment of the severity and the likelihood of occurrence of the these consequences.
- Visual representation of the descriptions of the similar systems found in our database.
- Distribution of the potential consequences related to chemical spills in similar systems.

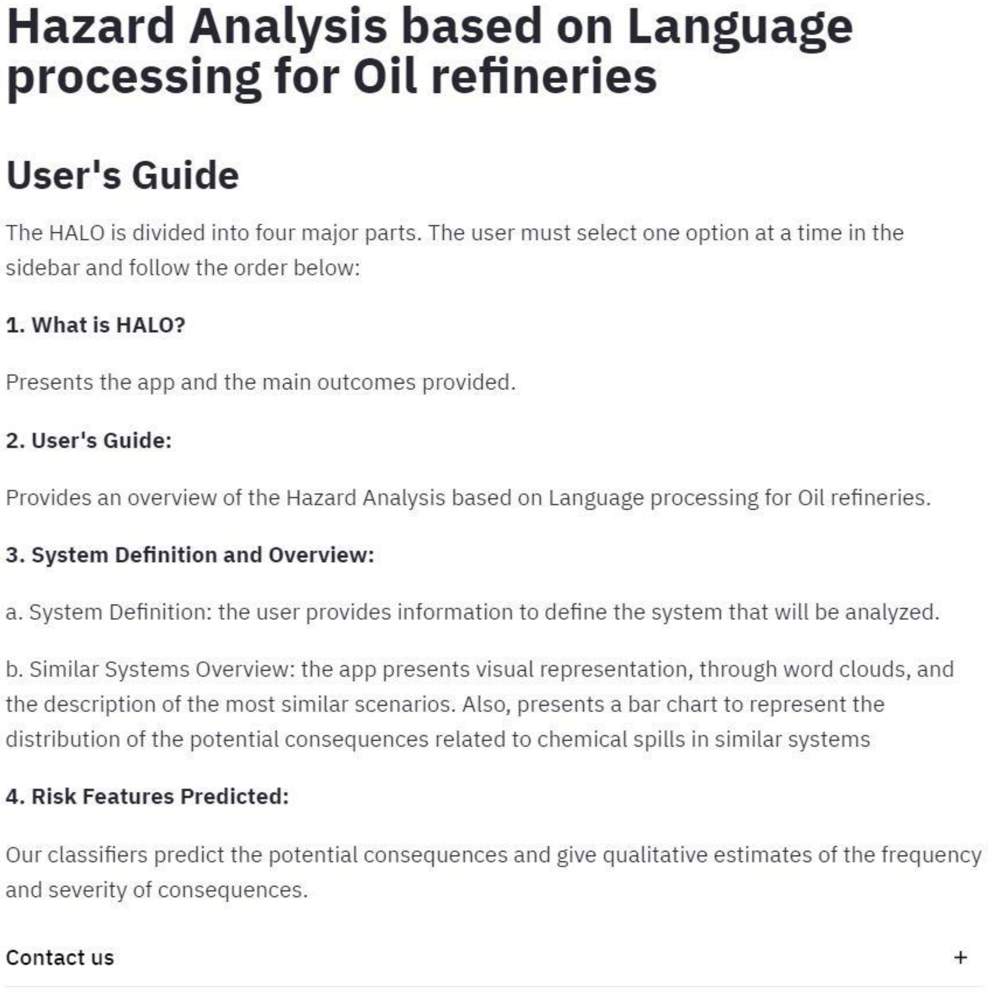
Contact us +

Source: The author (2022).

5.4.2 User's Guide

The second part (Figure 31) provides instructions for the user, explains how the app is organized, and briefly describes the parts of the app. As mentioned, HALO has four parts, and the user can select one of them at a time in the sidebar. First, the user must go to the third part to define the system of the oil refinery that will be analyzed. Also, in the third part visual outputs related to similar systems found in our database are displayed, the visual outputs include word clouds to summarize the descriptions of the similar systems and a bar chart to illustrate the potential consequences related to chemical spills in these similar systems.

Figure 31 – User's Guide.



Hazard Analysis based on Language processing for Oil refineries

User's Guide

The HALO is divided into four major parts. The user must select one option at a time in the sidebar and follow the order below:

1. What is HALO?

Presents the app and the main outcomes provided.

2. User's Guide:

Provides an overview of the Hazard Analysis based on Language processing for Oil refineries.

3. System Definition and Overview:

a. System Definition: the user provides information to define the system that will be analyzed.

b. Similar Systems Overview: the app presents visual representation, through word clouds, and the description of the most similar scenarios. Also, presents a bar chart to represent the distribution of the potential consequences related to chemical spills in similar systems

4. Risk Features Predicted:

Our classifiers predict the potential consequences and give qualitative estimates of the frequency and severity of consequences.

Source: The author (2022).

Then, the user can go to the fourth part, where the possible consequences predicted by our classifiers and their respective frequency and severity estimates is provided. The trained classifiers take the user's input features and each classifier considers two failure modes: small leakage and large leakage. The first classifier predicts the potential consequences related to both failure modes. The second and third classifiers take the user's input, the failure mode, and the list of potential consequences to predict the category that represents the likelihood of occurrence and the severity level of all potential consequences. The app is able to predict 11 potential consequences: burn injury, vapor cloud explosion, flash fire, irritation, soil contamination, pool fire, groundwater contamination, atmospheric contamination, toxic vapor cloud, and jet fire. Moreover, the consequences are classified into four severity categories and four likelihood categories, described in Table 1 and Table 2 respectively.

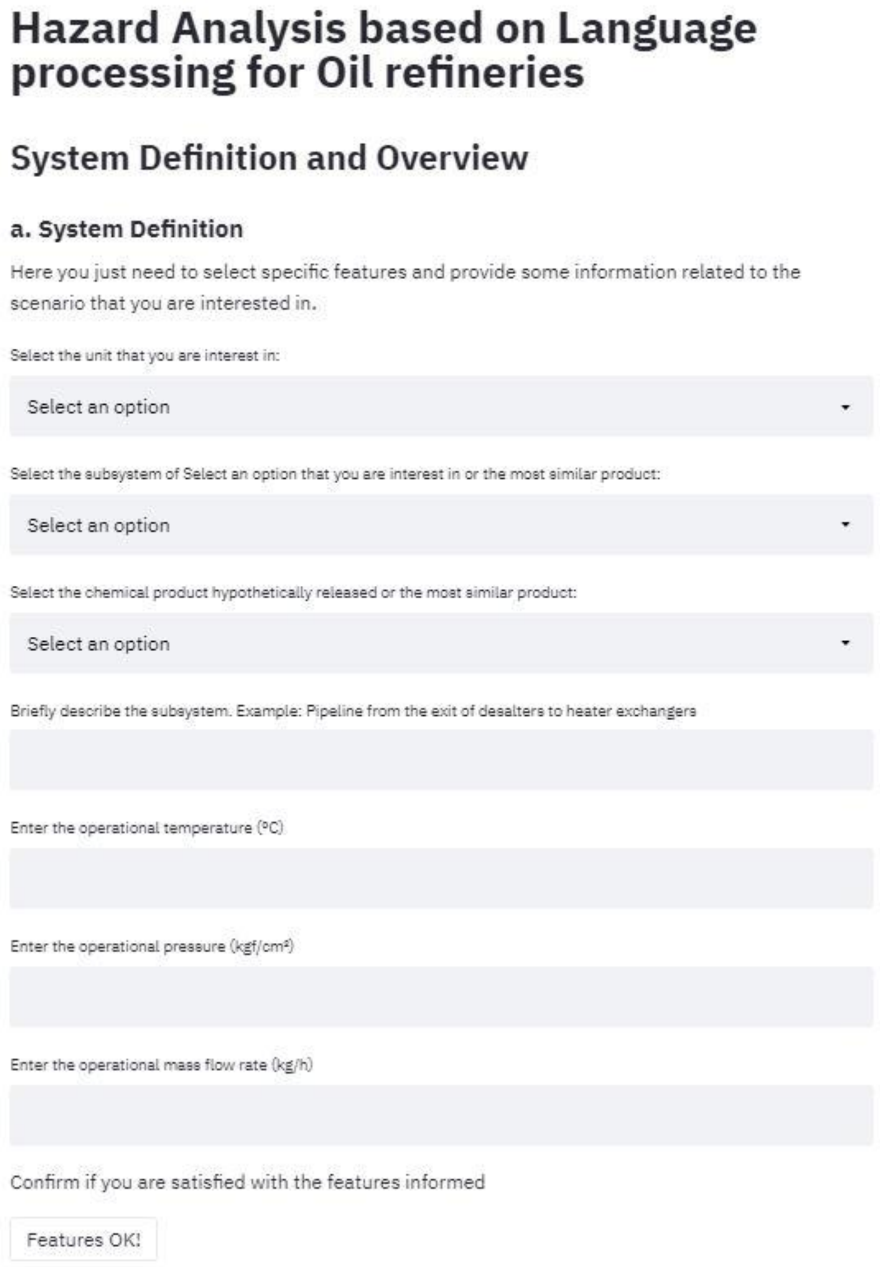
5.4.3 System Definition Overview

The third part of the app is composed of two subparts: a) System Definition (Figure 32) and b) Similar Systems Overview (Figure 33). In ‘a’, the user defines nine variables to characterize the system under analysis: unit (v_1), unit subsystem (v_2), chemical product involved in the hypothetical spill (v_3), instrument present in the subsystem analyzed (v_4), equipment material (v_5), description of the subsystem (v_6), operational temperature (v_7), pressure (v_8), and mass flow rate (v_9).

To define the features $v_1, v_2\ , v_3\ , v_4$ and v_5 the user chooses one option from a drop-down list. These are categorical variables and the ‘categories’ available are limited by the data used to build the models embedded in the app. For instance, the models are only able to predict risk features related to chemical releases in 22 units of an oil refinery (e.g., atmospheric distillation unit, delayed coking unit, hydrotreater unit, and others). If the unit that the user is interested in is not in the drop-down list, it means that there was no data related to this unit in our database. Moreover, v_4 and v_5 are optional features; if the user does not know the instruments present in the subsystem (or there are no instruments, e.g., sensors, valves) or the material of the pipeline/equipment, the user can choose ‘none’ and, thus, the model ignores these features.

The other variables to characterize the system are provided as a short text by the user. To define v_6 the user must provide a short description of the analyzed ‘section’, for example, ‘Pipeline from the exit of desalters to heater exchangers. This description must be provided in Portuguese because the models were trained in this language. Moreover, $v_7\ , v_8$ and v_9 must be provided in $°C, kgf.cm^{-2}$, and $kg.h^{-1}$ respectively. Finally, the user confirms the features by clicking on ‘Features OK!’ at the bottom of the screen.

Figure 32 – System Definition.



Hazard Analysis based on Language processing for Oil refineries

System Definition and Overview

a. System Definition

Here you just need to select specific features and provide some information related to the scenario that you are interested in.

Select the unit that you are interest in:

Select an option

Select the subsystem of Select an option that you are interest in or the most similar product:

Select an option

Select the chemical product hypothetically released or the most similar product:

Select an option

Briefly describe the subsystem. Example: Pipeline from the exit of desalters to heater exchangers

Enter the operational temperature (ºC)

Enter the operational pressure (kgf/cm²)

Enter the operational mass flow rate (kg/h)

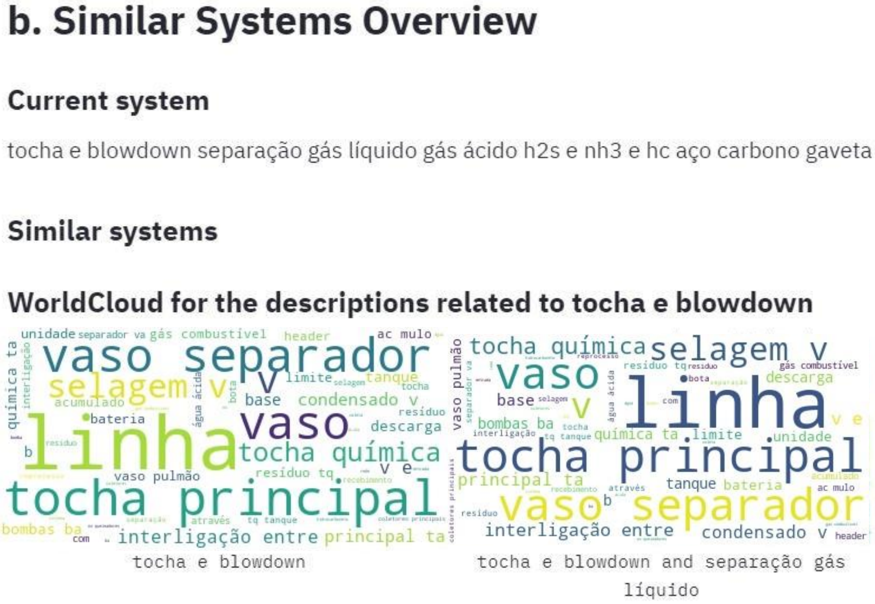
Confirm if you are satisfied with the features informed

Source: The author (2022).

5.4.4 Risk Features Predicted

When the user clicks the button, the page will load ‘b’ (Figure 33). The fourth part summarizes the system being analyzed by the user at the top of the screen and, then, displays the visual representation, through word clouds, of the descriptions of the similar systems in the dataset used to train the model. The world clouds are generated for the descriptions related to the unit selected by the user (v_1). The first word cloud represents all descriptions related to v1 and the second one is created using the descriptions of the systems that contains v_1 and v_2. Word clouds are widely adopted in text analysis in a static way to visually summarize text documents. This technique provides an overview of the texts that might help the user identify the number and kind of topics in the descriptions of similar systems. This statistical overview is accomplished by positively correlating the font size of the depicted word with the word frequency (HEIMERL, LOHMANN, *et al.*, 2014).

Figure 33 – Similar System Overview.



b. Similar Systems Overview

Current system

tocha e blowdown separação gás líquido gás ácido h2s e nh3 e hc aço carbono gaveta

Similar systems

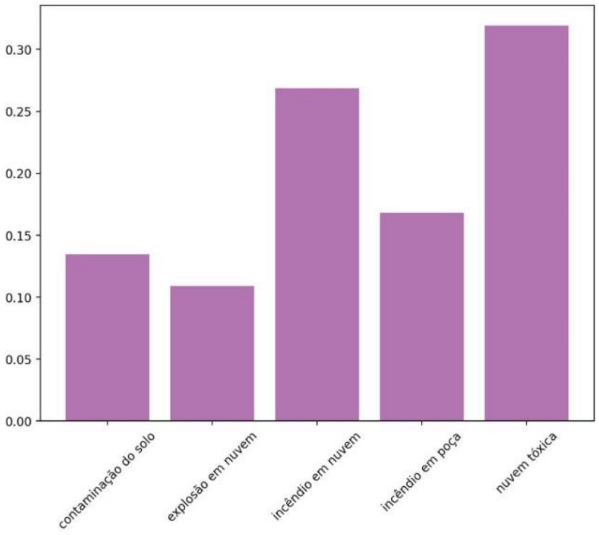
WorldCloud for the descriptions related to tocha e blowdown

Source: The author (2022).

In the same section, the app displays a bar chart (Figure 34) that shows the proportion of each potential consequences of chemical spills in the systems of v_1. For instance, in Figure 34 the consequence ‘toxic vapor cloud’ (‘nuvem tóxica’) represents 30% of the consequences of the accidental scenarios related to ‘tocha e blowdown’, in English ‘flare and blowdown’ unit (v_1), considering all accidental scenarios identified for this unit in our database. The bar chart can help the user identify consequences that are not predicted by the classifiers.

For instance, in this example we know that ‘toxic vapor cloud’ is the most frequent potential consequence for similar systems; however, the app’s classifiers might not predict this specific consequence. Thus, we could critically analyze whether this consequence is possible or not for the analyzed system. Next, the user can go to the fourth part of the app.

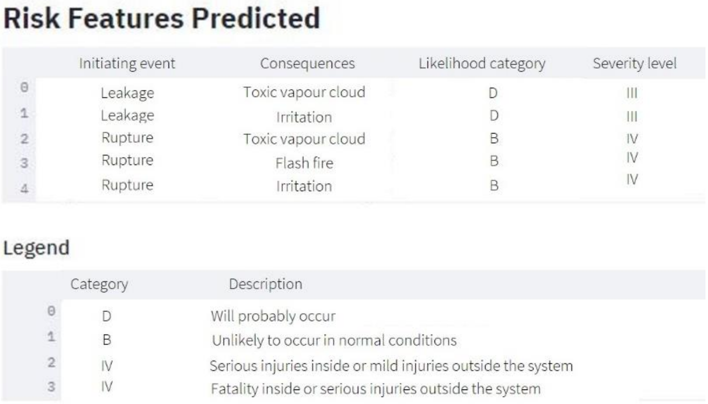
Figure 34 – Potential consequences related to similar systems.



Source: The author (2022).

When the user confirms the variables, these variables are fed into the app's classifiers (described in Sections 5.2.1, 5.2.2, and 5.2.3). Thus, the NLP-based classifiers process the textual data and predicts the risk features. The results of the classifiers predictions are displayed in a table (the first table presented in Figure 35), the column 'vazamento' corresponds to the failure modes and each row in the table corresponds to an accidental scenario (composed of failure mode, consequence, frequency, and severity).

Figure 35 – Predictions provided by HALO's classifiers.



Risk Features Predicted

	Initiating event	Consequences	Likelihood category	Severity level
0	Leakage	Toxic vapour cloud	D	III
1	Leakage	Irritation	D	III
2	Rupture	Toxic vapour cloud	B	IV
3	Rupture	Flash fire	B	IV
4	Rupture	Irritation	B	IV

Legend

	Category	Description
0	D	Will probably occur
1	B	Unlikely to occur in normal conditions
2	IV	Serious injuries inside or mild injuries outside the system
3	IV	Fatality inside or serious injuries outside the system

Source: The author (2022).

5.4.5 Conclusion

HALO can help the analyst to identify and evaluate the hazards related to chemical spills in an oil refinery by providing a list of potential consequences, their severity and likelihood categories, and visual outputs. These results can be used to prioritize the most critical scenarios and, thus, reduce the number of scenarios posteriorly quantified. Additionally, the word clouds and bar chart provided by the app summarize the knowledge from previous risk studies conducted for an actual oil refinery. These results can support the analyst to postulate additional accidental scenarios and assess the consistency of the results predicted by the classifiers.

5.5 GENERAL CONCLUSION

A method based on TM techniques and pre-trained BERT model was developed to identify risk features in an oil refinery. First, a corpus was built using PrHA documents to train the models. The texts were automatically extracted from the documents and, then, preprocessed into a convenient format for the learning algorithms. Next, the pre-trained model was tailored for performing three tasks: i) to identify possible consequences, given the occurrence of a leakage; ii) to classify the severity of the consequences; iii) to classify the likelihood of occurrence of the accident scenario. As a result, we developed three models that could extract sufficient knowledge from the textual data and yielded satisfactory training and test outcomes.

These outcomes underscore that TM and NLP can be adopted to support identification and analysis of hazards related to chemical spills in an oil refinery. The trained classifiers can provide to risk analysts a starting point to postulate different scenarios potential consequences of different scenarios and qualitative description of risks in terms of expected likelihood and severity of consequences. The proposed methodology presented satisfactory results; thus, the trained classifiers were embedded into a web-based app called HALO.

Moreover, developing techniques to identify features about catastrophic events with extreme consequences is quite useful for QRA. However, it may be quite challenging to develop good models when dealing with rare accident events, since limited knowledge exists for them. Moreover, in NLP datasets, DA is difficult due to language complexity. Thus, we explored different DA configurations to balance our dataset in order to improve the performance of the NLP model in classifying remote events.

One possible alternative to DA is to use Few-Shot Learning (BROWN, MANN, *et al.*, 2020); thus, it will be object of future research. In addition, in future works we intend to evaluate whether, through the features extracted through fine-tuning using configuration (v), it is possible to identify other characteristics, such as the severity of the consequences of these rare events.

Further studies, which would take into account other engineering documents, such as flowcharts, equipment and material lists, should be undertaken to investigate the possibility of conflating more information about the system and the data stored in the PrHA documents. This could improve the model's learning process and reduce biases usually found in early stages of QRA.

Although the scope of this study was restricted to the oil refinery context, the methodology can be applicable to different industrial systems, being necessary to provide data from the context of interest to train the models. It is important to emphasize that the model’s predictions are limited to what is provided through the training data. Therefore, caution must be taken when generalizing the predictions, and the results must be carefully evaluated by the risk analysts.

6 CONCLUDING REMARKS

This thesis aimed at developing two methodologies that can be applied at different time epochs over the lifecycle of an industrial system to aid RA, in both occupational safety (Chapter 4) and process safety (Chapter 5). Thus, this thesis may contribute to the prevention and mitigation of occupational accidents and/or major accidents that can threaten human life and lead to environmental degradation and to property damage, consequently avoiding economic losses, fatalities, and negative effects on the company's image and society.

Indeed, the outcome of this thesis strengthens the idea that information contained as text data can be automatically extracted and processed by TM and NLP techniques to support risk studies and consequently improve risk management and safety in the work/industrial environment.

First, we analyzed a dataset of accident investigation reports of a hydroelectric power company through different NLP approaches. We were able to identify the usefulness of several categories already adopted, but there were also existing ones that we found out to be ineffective in terms of their descriptions. The results obtained in the exploratory analysis suggested that a lower number of categories would be more suitable for this specific database. This is probably due to a lack of standardization and understanding of the pre-defined categories.

Moreover, the improvement on the performance of the classifiers due to the curation may indicate the presence of inconsistencies in the original classification among the ‘accident agents’. In addition, we performed a sensitivity analysis removing classes with few instances (i.e., ‘scaffolding’, ‘chemical products’, and ‘others’), and the performance of most of the models improved even further. These outcomes indicated that the classes are not well- understood by the classifiers (i.e., it consists of multiple and heterogeneous event descriptions) and are under-represented in our database.

Therefore, we showed the importance of the company’s safety culture to keep safety technician engaged in carefully constructing an accident database. In fact, the safety technician must have the correct understanding of what and how to fill out each required field in the report, which is only achieved by continuous training. This would improve the quality of the reports and allow keeping the original categorization. In addition, a well-designed database provides useful information for risk management and decision-making.

The downside of the proposed methodology to assess accident investigation reports is that it is not yet completely automatic, still relying on manual curation. Therefore, in future work we intend to improve manual curation process, such as adopting clusters to group the accidents and label them in a less manual way. Moreover, since the analyzed database has only 626 reports, in an ongoing study we are applying this methodology to a larger accident base in a different context, more specifically to aviation accident data, allowing us to build and train DL architectures.

Second, we showed that information contained in past PrHA of an oil refinery can be reused to train models to extract identify different risk features and support the initial stage of QRA. The proposed methodology based on BERT, an advanced pre-trained language model, does not require an absurd amount of data like most NLP approaches, which can reach up to trillions of training tokens (GAO, Leo, BIDERMAN, *et al.*, 2020, HOFFMANN, BORGEAUD, *et al.*, 2022), to achieve satisfactory results. In addition, the proposed methodology can be easily adapted to different industrial systems, being necessary to provide data from the context of interest to train the models. In addition. We also have developed a web-app called HALO, where we embedded the trained classifiers that were trained to identify the potential consequences of different scenarios and to describe qualitatively risks in terms of expected likelihood and severity of consequences were embedded into a web-based app called HALO.

Indeed, the proposed method could be a useful tool to support hazard identification and analysis; instead of starting the QRA from scratch, analysts could either reuse knowledge from previous studies or process studies for similar plants. This may be rather useful especially for plants, which are brand new and depend on the approval of the environmental regulators to start the development of the facility design and construction. Then, experts may use that entire source of knowledge to reduce the uncertainty for performing risk analysis based on a model trained with all the available information collected and processed from past risk studies. Furthermore, we believe that the predictions provided by the models could indicate errors made by experts during the PrHAs. Experts can critically evaluate scenarios misclassified by the trained models and decide whether the model's prediction makes sense. Thus, the proposed methodology would also enable experts to correct the PrHAs..

Regarding the methodology to support QRA in O&G industries, we explored different DA configurations to balance our dataset in order to improve the performance of the NLP model in classifying remote events. One possible alternative to DA is to use Few-Shot Learning (BROWN, MANN, *et al.*, 2020); thus, it will be object of future research. In addition, in future works we intend to evaluate whether, through the features extracted through fine-tuning using configuration (v), it is possible to identify other characteristics, such as the severity of the consequences of these rare events.

Further studies, which would take into account other engineering documents, such as flowcharts, equipment and material lists, should be undertaken to investigate the possibility of conflating more information about the system and the data stored in the PrHA documents. Furthermore, different NLP models should be explored, since a number of new models have been derived from the transformers architecture, such as XLNet (YANG, DAI, *et al.*, 2019), RoBERTa (LIU, Yinhan, OTT, *et al.*, 2019), ELECTRA (CLARK, LUONG, *et al.*, 2020), ALBERT (LAN, CHEN, *et al.*, 2019), Gato (REED, ZOLNA, *et al.*, 2022) and Chinchilla (HOFFMANN, BORGEAUD, *et al.*, 2022). This could improve the model's learning process and reduce biases usually found in early stages of QRA

REFERENCES

ABDAT, F., LECLERCQ, S., CUNY, X., *et al.* "Extracting recurrent scenarios from narrative texts using a Bayesian network: Application to serious occupational accidents with movement disturbance", **Accident Analysis & Prevention**, v. 70, p. 155–166, set. 2014. DOI: 10.1016/j.aap.2014.04.004. .

AHMAD, S. I., HASHIM, H., HASSIM, M. H., *et al.* "Development of hazard prevention strategies for inherent safety assessment during early stage of process design", **Process Safety and Environmental Protection**, v. 121, p. 271–280, 2019. DOI: 10.1016/j.psep.2018.10.006. .

AHMADPOUR-GESHLAGI, R., GILLANI, N., AZAMI–AGHDASH, S., *et al.* "Investigating the status of accident precursor management in East Azarbaijan Province Gas Company", **International Journal of Occupational Safety and Ergonomics**, p. 1–12, jul. 2020. DOI: 10.1080/10803548.2020.1770451. .

ALI, M. X. M., ARIFIN, K., ABAS, A., *et al.* "Systematic Literature Review on Indicators Use in Safety Management Practices among Utility Industries", **International Journal of Environmental Research and Public Health**, v. 19, n. 10, 2022. DOI: 10.3390/ijerph19106198. .

ANDRZEJCZAK, C., KARWOWSKI, W., THOMPSON, W. "The Identification of Factors Contributing to Self-Reported Anomalies in Civil Aviation", **International Journal of Occupational Safety and Ergonomics**, v. 20, n. 1, p. 3–18, jan. 2014. DOI: 10.1080/10803548.2014.11077029. .

ANSALDI, S. M., SIMEONI, C., FRANCESCO, A. Di, *et al.* "Extracting Knowledge from Near Miss Reports using Machine-Learning Techniques". 2020. **Anais** [...] [S.l: s.n.], 2020. DOI: 10.3850/981-973-0000-00-0 esrel2020psam15-paper.

APOSTOLAKIS, G. E. "How useful is quantitative risk assessment?", **Risk Analysis**, v. 24, n. 3, p. 515–520, 2004. DOI: 10.1111/j.0272-4332.2004.00455.x. .

ARUNRAJ, N. S., MAITI, J. "Risk-based maintenance — Techniques and applications", **Journal of hazardous materials**, v. 142, n. June 2006, p. 653–661, 2007. DOI: 10.1016/j.jhazmat.2006.06.069. .

ASNANI, K., PAWAR, J. D. "Improving Coherence of Topic Based Aspect Clusters using Domain Knowledge", **Computacion y Sistemas**, v. 22, n. 4, p. 1403–1414, 2018. DOI: 10.13053/CyS-22-4-2401. .

AVEN, T., KESSENICH, A. M. Van. "Teaching children and youths about risk and risk analysis : what are the goals and the risk analytical foundation ?", **Journal of Risk Research**, v. 23, n. 5, p. 557–570, 2020. DOI: 10.1080/13669877.2018.1547785. Disponível em: https://doi.org/10.1080/13669877.2018.1547785.

AZIZ, A., AHMED, S., KHAN, F. I. "An ontology-based methodology for hazard identification and causation analysis", **Process Safety and Environmental Protection**, v. 123, p. 87–98, 2019. DOI: 10.1016/j.psep.2018.12.008. .

BADRI, N., NOURAI, F., RASHTCHIAN, D. "A multivariable approach for estimation of vapor cloud explosion frequencies for independent congested spaces to be used in occupied building risk assessment", **Process Safety and Environmental Protection**, v. 91, n. 1–2, p. 19–30, 2013. DOI: 10.1016/j.psep.2011.12.002. .

BAKER, H., HALLOWELL, M. R., TIXIER, A. J. P. "Automatically learning construction injury precursors from text", **Automation in Construction**, v. 118, n. June, p. 103145, 2020. DOI: 10.1016/j.autcon.2020.103145. Disponível em: https://doi.org/10.1016/j.autcon.2020.103145.

BALLESTEROS, M. F., SUMNER, S. A., LAW, R., *et al.* "Advancing injury and violence prevention through data science", **Journal of Safety Research**, v. 73, p. 189–193, jun. 2020. DOI: 10.1016/j.jsr.2020.02.018. .

BATTIATO, S., FARINELLA, G. M., GALLO, G., *et al.* "On-board monitoring system for road traffic safety analysis", **Computers in Industry**, v. 98, p. 208–217, jun. 2018. DOI: 10.1016/j.compind.2018.02.014. .

BAVARESCO, R., ARRUDA, H., ROCHA, E., *et al.* "Internet of Things and occupational well-being in industry 4.0: A systematic mapping study and taxonomy", **Computers & Industrial Engineering**, v. 161, p. 107670, nov. 2021. DOI: 10.1016/j.cie.2021.107670. .

BAYBUTT, P. "On the completeness of scenario identification in process hazard analysis (PHA) Paul Baybutt", **Journal of Loss Prevention in the Process Industries**, v. 55, p. 492–499, 2018. DOI: 10.1016/j.jlp.2018.05.010. Disponível em: https://doi.org/10.1016/j.jlp.2018.05.010.

BELLEGARDA, J. R. "Statistical language model adaptation: Review and perspectives", **Speech Communication**, v. 42, n. 1, p. 93–108, 2004. DOI: 10.1016/j.specom.2003.08.002. .

BENEKOS, I., DIAMANTIDIS, D. "On risk assessment and risk acceptance of dangerous goods transportation through road tunnels in Greece", **Safety Science**, v. 91, p. 1–10, 2017. DOI: 10.1016/j.ssci.2016.07.013. Disponível em: http://dx.doi.org/10.1016/j.ssci.2016.07.013.

BENGFORT, B., BILBRO, R., OJEDA, T. **Applied Text Analysis with Python: Enabling Language-aware Data Products with Machine Learning**. [S.l.], O’Reilly Media, Inc, 2018.

BENGIO, Y., DUCHARME, R., PASCAL, V., *et al.* "A Neural Probabilistic Language Model", **Journal of Machine Learning Research**, v. 3, 2003. .

BERNECHEA, E. J., VÍLCHEZ, J. A., ARNALDOS, J. "A model for estimating the impact of the domino effect on accident frequencies in quantitative risk assessments of storage facilities", **Process Safety and Environmental Protection**, v. 91, n. 6, p. 423–437, 2013. DOI: 10.1016/j.psep.2012.09.004. .

BERTKE, S. J., MEYERS, A. R., WURZELBACHER, S. J., *et al.* "Development and evaluation of a Naïve Bayesian model for coding causation of workers’ compensation claims", **Journal of Safety Research**, v. 43, n. 5–6, p. 327–332, dez. 2012. DOI: 10.1016/j.jsr.2012.10.012. .

BHATTACHARJEE, P., DEY, V., MANDAL, U. K. "Risk assessment by failure mode and effects analysis (FMEA) using an interval number based logistic regression model", **Safety Science**, v. 132, p. 104967, 2020. DOI: 10.1016/j.ssci.2020.104967. .

BIANCHI, B., BENGOLEA MONZÓN, G., FERRER, L., *et al.* "Human and computer estimations of Predictability of words in written language", **Scientific Reports**, v. 10, n. 1, p. 1–11, 2020. DOI: 10.1038/s41598-020-61353-z. .

BIRD, S., KLEIN, E., LOPER, E. **Natural Language Processing with Python**. [S.l.], O’Reilly Media, Inc., 2009.

BOGGS, A. M., WALI, B., KHATTAK, A. J. "Exploratory analysis of automated vehicle crashes in California : A text analytics & hierarchical Bayesian heterogeneity-based approach", **Accident Analysis and Prevention**, v. 135, n. June 2019, p. 105354, 2020. DOI: 10.1016/j.aap.2019.105354. Disponível em: https://doi.org/10.1016/j.aap.2019.105354.

BREIMAN, L. "Random Forests", **Machine Learning**, v. 45, n. 1, p. 5–32, 2001. DOI: 10.1023/A:1010933404324. .

BROWN, T. B., MANN, B., RYDER, N., *et al.* "Language models are few-shot learners", **Advances in Neural Information Processing Systems**, v. 33, p. 1877–1901, 2020. .

CAI, M. "Natural language processing for urban research: A systematic review", **Heliyon**, v. 7, n. 3, p. e06322, 2021. DOI: 10.1016/j.heliyon.2021.e06322. Disponível em: https://doi.org/10.1016/j.heliyon.2021.e06322.

CARRASQUILLA, J., MELKO, R. G. "Machine learning phases of matter", **Nature Physics**, v. 13, n. 5, p. 431–434, 2017. DOI: 10.1038/nphys4035. .

CASAL, J. **Evaluation of the Effects and Consequences of Major Accidents in Industrial Plants: Second Edition**. [S.l: s.n.], 2017.

CCPS. **Guidelines for Hazard Evaluation Procedures**. 3. ed. New York, American Institute of Chemical Engineers, 2008.

CHANG, W., XU, Z., ZHOU, S., *et al.* "Research on detection methods based on Doc2vec abnormal comments", **Future Generation Computer Systems**, v. 86, p. 656–662, 2018. DOI: 10.1016/j.future.2018.04.059. Disponível em: https://doi.org/10.1016/j.future.2018.04.059.

CHENG, C.-W., LEU, S.-S., CHENG, Y.-M., *et al.* "Applying data mining techniques to explore factors contributing to occupational injuries in Taiwan’s construction industry", **Accident Analysis & Prevention**, v. 48, p. 214–222, set. 2012. DOI: 10.1016/j.aap.2011.04.014. .

CHOWDHARY, K. R. **Fundamentals of Artificial Intelligence**. Jodhpur, Springer India, 2020.

CLARK, K., LUONG, M.-T., LE, Q. V., *et al.* "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators", **arXiv preprint arXiv:2003.10555**, p. 1–18, 2020. Disponível em: http://arxiv.org/abs/2003.10555.

DE MULDER, W., BETHARD, S., MOENS, M. F. "A survey on the application of recurrent neural networks to statistical language modeling", **Computer Speech and Language**, v. 30, n. 1, p. 61–98, 2015. DOI: 10.1016/j.csl.2014.09.005. Disponível em: http://dx.doi.org/10.1016/j.csl.2014.09.005.

DEMIRBAS, A., BAMUFLEH, H. S. "Optimization of crude oil refining products to valuable fuel blends", **Petroleum Science and Technology**, v. 35, n. 4, p. 406–412, 2017. DOI: 10.1080/10916466.2016.1261162. Disponível em: https://doi.org/10.1080/10916466.2016.1261162.

DEVLIN, J., CHANG, M., KENTON, L., *et al.* "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", **arXiv preprint arXiv:1810.04805**, 2018. .

DRURY, B., ROCHE, M. "A survey of the applications of text mining for agriculture", **Computers and Electronics in Agriculture**, v. 163, n. February, p. 104864, 2019. DOI: 10.1016/j.compag.2019.104864. .

EL AKROUCHI, M., BENBRAHIM, H., KASSOU, I. "End-to-end LDA-based automatic weak signal detection in web news", **Knowledge-Based Systems**, v. 212, p. 106650, jan. 2021. DOI: 10.1016/j.knosys.2020.106650. .

FARHADI, F., NIA, V. P., LODI, A. "Activation Adaptation in Neural Networks", **arXiv preprint arXiv:1901.09849**, 2019. .

FELDMAN, R., SANGER, J. **The text mining handbook: advanced approaches in analyzing unstructured data**. [S.l.], Cambridge University Press, 2007.

GAGNE, J. C. De, HALL, K., CONKLIN, J. L., *et al.* "Uncovering cyberincivility among nurses and nursing students on twitter: A data mining study", **International Journal of Nursing Studies**, v. 89, p. 24–31, 2019. DOI: 10.1016/j.ijnurstu.2018.09.009. Disponível em: https://doi.org/10.1016/j.ijnurstu.2018.09.009.

GAO, B., PAVEL, L. "On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning", **arXiv preprint arXiv:1704.00805**, p. 1–10, 2017. .

GAO, L., BIDERMAN, S., BLACK, S., *et al.* "The Pile: An 800GB Dataset of Diverse Text for Language Modeling", **arXiv preprint**, v. arXiv:2101, 2020. Disponível em: http://arxiv.org/abs/2101.00027.

GEORGE K, S., JOSEPH, S. "Text Classification by Augmenting Bag of Words (BOW) Representation with Co-occurrence Feature", **IOSR Journal of Computer Engineering**, v. 16, n. 1, p. 34–38, 2014. DOI: 10.9790/0661-16153438. .

GOODFELLOW, I., BENGIO, Y., COURVILLE, A. **Deep Learning**. [S.l.], MIT Press, 2016.

GREENBERG, M., COX, A., BIER, V., *et al.* "Risk Analysis: Celebrating the Accomplishments and Embracing Ongoing Challenges", **Risk Analysis**, v. 40, n. 1, p. 2113–2127, 2020. DOI: 10.1111/risa.13487. .

GUARAV, N., KIRSTEN, V., LEHTO, M. "Intelligent human-machine approaches for assigning groups of injury codes to accident narratives", **Safety Science**, v. 125, p. 104585, 2020. .

GUIMARÃES, M. S., ARAÚJO, H. H. G., LUCAS, T. C., *et al.* "An NLP and Text Mining–based approach to categorize occupational accidents". 2020. **Anais** [...] [S.l: s.n.], 2020.

GUIOCHET, J. "Hazard analysis of human-robot interactions with HAZOP-UML", **Safety Science**, v. 84, p. 225–237, 2016. DOI: 10.1016/j.ssci.2015.12.017. .

GULIJK, C. van, HOLMES, V. "Verification of Safety Rules using NLP". 2020. **Anais** [...] [S.l: s.n.], 2020. DOI: 10.3850/981-973-0000-00-0 esrel2020psam15-paper.

GUO, X., JI, J., KHAN, F., *et al.* "Fuzzy bayesian network based on an improved similarity aggregation method for risk assessment of storage tank accident", **Process Safety and Environmental Protection**, v. 149, p. 817–830, 2021. DOI: 10.1016/j.psep.2020.07.030. .

HÄMÄLÄINEN, P., TAKALA, J., KIAT, T. B. **Global Estimates of Occupational Accidents and Work-Related Illnesses 2017**. **Workplace Safety and Health**

Institute. [S.l: s.n.], 2017.

HAO, M., NIE, Y. "Hazard identification, risk assessment and management of industrial system: Process safety in mining industry", **Safety Science**, v. 154, n. October 2020, p. 105863, 2022. DOI: 10.1016/j.ssci.2022.105863. Disponível em: https://doi.org/10.1016/j.ssci.2022.105863.

HAVRLANT, L., KREINOVICH, V. "A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation)", **International Journal of General Systems**, v. 46, n. 1, p. 27–36, 2017. DOI: 10.1080/03081079.2017.1291635. .

HE, R., LI, X., CHEN, G., *et al.* "A quantitative risk analysis model considering uncertain information", **Process Safety and Environmental Protection**, v. 118, p. 361–370, ago. 2018. DOI: 10.1016/j.psep.2018.06.029. Disponível em: https://doi.org/10.1016/j.psep.2018.06.029.

HEIDARYSAFA, M., KOWSARI, K., BARNES, L., *et al.* "Analysis of Railway Accidents ' Narratives Using Deep Learning". 2018. **Anais** [...] [S.l.], IEEE, 2018. p. 1446–1453. DOI: 10.1109/ICMLA.2018.00235.

HEIDINGER, D., GATZERT, N. "Awareness, Determinants and Value of Reputation Risk Management: Empirical Evidence from the Banking and Insurance Industry", **Journal of Banking and Finance**, v. 91, p. 106–118, 2018. DOI: 10.1016/j.jbankfin.2018.04.004. Disponível em: https://doi.org/10.1016/j.jbankfin.2018.04.004.

HEIMERL, F., LOHMANN, S., LANGE, S., *et al.* "Word cloud explorer: Text analytics based on word clouds". 2014. **Anais** [...] [S.l: s.n.], 2014. DOI: 10.1109/HICSS.2014.231.

HOFFMANN, J., BORGEAUD, S., MENSCH, A., *et al.* "Training Compute-Optimal Large Language Models", **arXiv preprint**, v. arXiv:2203, p. 1–36, 2022. Disponível em: http://arxiv.org/abs/2203.15556.

HOWARD, J., RUDER, S. "Universal Language Model Fine-tuning for Text Classification", **arXiv preprint arXiv:1801.06146**, 2018. .

HUGHES, P., SHIPP, D., FIGUERES-ESTEBAN, M., *et al.* "From free-text to structured safety management: Introduction of a semi-automated classification method of railway hazard reports to elements on a bow-tie diagram", **Safety Science**, v. 110, n. March, p. 11–19, 2018. DOI: 10.1016/j.ssci.2018.03.011. .

INSS. **Auxílio-doença**. 2022. Disponível em: https://www.gov.br/inss/pt-br/saiba-mais/auxilios/auxilio-doenca. Acesso em: 8 fev. 2022.

ISO. "ISO 31000: Risk management — Guidelines", 2018. .

JIN, R., WANG, F., LIU, D. "Dynamic probabilistic analysis of accidents in construction projects by combining precursor data and expert judgments", **Advanced Engineering Informatics**, v. 44, n. January, p. 101062, 2020. DOI: 10.1016/j.aei.2020.101062. Disponível em: https://doi.org/10.1016/j.aei.2020.101062.

JONES, S., KIRCHSTEIGER, C., BJERKE, W. "The importance of near miss reporting to further improve safety performance", **Journal of Loss Prevention in the Process Industries**, v. 12, n. 1, p. 59–67, 1999. DOI: 10.1016/S0950-4230(98)00038-2. .

KALYAN, K. S., RAJASEKHARAN, A., SANGEETHA, S. "AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing", **arXiv preprint arXiv: arXiv:2108.05542v2**, v. 2108.05542, p. 1–42, 2021. Disponível em: http://arxiv.org/abs/2108.05542.

KAMIL, M. Z., TALEB-BERROUANE, M., KHAN, F., *et al.* "Dynamic domino effect risk assessment using Petri-nets", **Process Safety and Environmental Protection**, v. 124, p. 308–316, 2019. DOI: 10.1016/j.psep.2019.02.019. .

KHODADADYAN, A., RESEARCHER, P. D., MYTHEN, G., *et al.* "Grasping the nettle? Considering the contemporary challenges of risk assessment", **Journal of Risk Research**, v. 24, n. 12, p. 1605–1618, 2021. DOI: 10.1080/13669877.2021.1894472. Disponível em: https://doi.org/10.1080/13669877.2021.1894472.

KHURANA, D., KOLI, A., KHATTER, K., *et al.* "Natural Language Processing: State of The Art, Current Trends and Challenges", **arXiv preprint arXiv:1708.05148**, 2017. .

KIM, J., YOON, J., PARK, E., *et al.* "Patent document clustering with deep embeddings", **Scientometrics**, v. 123, n. 2, p. 563–577, 2020. DOI: 10.1007/s11192-020-03396-7. Disponível em: https://doi.org/10.1007/s11192-020-03396-7.

KOC, K., GURGUN, A. P. "Scenario-based automated data preprocessing to predict severity of construction accidents", **Automation in Construction**, v. 140, n. May, p. 104351, 2022. DOI: 10.1016/j.autcon.2022.104351. Disponível em: https://doi.org/10.1016/j.autcon.2022.104351.

KUHN, K. D. "Using structural topic modeling to identify latent topics and trends in aviation incident reports", **Transportation Research Part C**, v. 87, n. December 2017, p. 105–122, 2019. DOI: 10.1016/j.trc.2017.12.018. Disponível em: https://doi.org/10.1016/j.trc.2017.12.018.

KURIAN, D., SATTARI, F., LEFSRUD, L., *et al.* "Using machine learning and keyword analysis to analyze incidents and reduce risk in oil sands operations", **Safety Science**, v. 130, n. February, p. 104873, 2020. DOI: 10.1016/j.ssci.2020.104873. Disponível em: https://doi.org/10.1016/j.ssci.2020.104873.

LAN, Z., CHEN, M., GOODMAN, S., *et al.* "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", **arXiv preprint arXiv:1909.11942**, p. 1–17, 2019. Disponível em: http://arxiv.org/abs/1909.11942.

LANDUCCI, G., PALTRINIERI, N. "A methodology for frequency tailorization dedicated to the Oil & Gas sector", **Process Safety and Environmental Protection**, v. 104, n. Part A, p. 123–141, 2016. DOI: 10.1016/j.psep.2016.08.012. .

LAU, J. H., BALDWIN, T. "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation", p. 78–86, 2016. DOI: 10.18653/v1/w16-1609. .

LEE, H., YOON, Y. "Engineering doc2vec for automatic classification of product descriptions on O2O applications", **Electronic Commerce Research**, v. 18, n. 3, p. 433–456, 2018. DOI: 10.1007/s10660-017-9268-5. .

LEU, S., CHANG, C. "Bayesian-network-based safety risk assessment for steel construction projects", **Accident Analysis and Prevention**, v. 54, p. 122–133, 2013.

.

LI, T., MEI, T., KWEON, I. S., *et al.* "Contextual bag-of-words for visual categorization", **IEEE Transactions on Circuits and Systems for Video Technology**, v. 21, n. 4, p. 381–392, 2011. DOI: 10.1109/TCSVT.2010.2041828. .

LI, X., CHEN, G., JIANG, S., *et al.* "Developing a dynamic model for risk analysis under uncertainty : Case of third-party damage on subsea pipelines", **Journal of Loss Prevention in the Process Industries**, v. 54, n. April, p. 289–302, 2018. .

LI, Y., WANG, H., BAI, K., *et al.* "Dynamic intelligent risk assessment of hazardous chemical warehouse fire based on electrostatic discharge method and improved support vector machine", **Process Safety and Environmental Protection**, v. 145, p. 425–434, 2021. DOI: 10.1016/j.psep.2020.11.012. Disponível em: https://doi.org/10.1016/j.psep.2020.11.012.

LISI, R., CONSOLO, G., MASCHIO, G., *et al.* "Estimation of the impact probability in domino effects due to the projection of fragments", **Process Safety and Environmental Protection**, v. 93, p. 99–110, 2015. DOI: 10.1016/j.psep.2014.05.003. .

LIU, G., BOYD, M., YU, M., *et al.* "Identifying causality and contributory factors of pipeline incidents by employing natural language processing and text mining techniques", **Process Safety and Environmental Protection**, v. 152, p. 37–46, 2021. DOI: 10.1016/j.psep.2021.05.036. .

LIU, S., LEE, K., LEE, I. "Document-level multi-topic sentiment classification of Email data with BiLSTM and data augmentation", **Knowledge-Based Systems**, v. 197, n. 7, p. 105918, 2020. DOI: 10.1016/j.knosys.2020.105918. .

LIU, Y., OTT, M., GOYAL, N., *et al.* "RoBERTa: A Robustly Optimized BERT Pretraining Approach", **arXiv preprint arXiv:1907.11692**, n. 1, 2019. Disponível em: http://arxiv.org/abs/1907.11692.

LOMBARDI, M., FARGNOLI, M., PARISE, G. "Risk profiling from the European statistics on accidents at work (ESAW) accidents′ databases: A case study in construction sites", **International Journal of Environmental Research and Public Health**, v. 16, n. 23, 2019. DOI: 10.3390/ijerph16234748. .

LONGO, F., PADOVANO, A., GAZZANEO, L., *et al.* "Human factors, ergonomics and Industry 4.0 in the Oil&Gas industry: A bibliometric analysis", **Procedia Computer Science**, v. 180, n. 2019, p. 1049–1058, 2021. DOI: 10.1016/j.procs.2021.01.350. Disponível em: https://doi.org/10.1016/j.procs.2021.01.350.

MA, E. "nlpaug Documentation", 2021. .

MACÊDO, J. B., AICHELE, D., MOURA, M. das C., *et al.* "A text mining and NLP approach for identifying potential consequences of accidents in an oil refinery", **30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM 2020**, p. 1269–1275, 2020a. DOI: 10.3850/978-981-14-8593-0. .

MACÊDO, J. B., AICHELE, D., MOURA, M. das C., *et al.* "A Text Mining and NLP Approach for Identifying Potential Consequences of Accidents in an Oil Refinery". 2020b. **Anais** [...] Singapore, Research Publishing, 2020. DOI: 10.3850/978-981-14-8593-0_4527-cd.

MACÊDO, J. B., AICHELE, D., MOURA, M. das C., *et al.* "A web app to support hazard identification of oil refineries". 2021. **Anais** [...] Angers, France, [s.n.], 2021.

MACÊDO, J. B., MOURA, M. das C., AICHELE, D., *et al.* "Identification of risk features using text mining and BERT-based models: Application to an oil refinery". 2021. **Anais** [...] Rio de Janeiro, [s.n.], 2021.

MACÊDO, J. B., MOURA, M. das C., AICHELE, D., *et al.* "Identification of risk features using text mining and BERT-based models: Application to an oil refinery", **Process Safety and Environmental Protection**, v. 158, p. 382–399, fev. 2022. DOI: 10.1016/j.psep.2021.12.025. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0957582021006996.

MACÊDO, J. B., MOURA, M. das C., LINS, I. D., *et al.* **Hazard Analysis based on Language processing for Oil refineries (HALO)**. . Recife, Universidade Federal de Pernambuco. Processo n°: BR512022000211-6. , 2021

MACÊDO, J. B., MOURA, M. das C., LINS, I. D., *et al.* "Identification of features of rare risk events in oil refineries using Natural Language Processing (NLP)", **Proceedings of the 32nd European Safety and Reliability Conference (ESREL 2022)**, n. Esrel, p. 689–697, 2022. DOI: 10.3850/978-981-18-5183-4. .

MACÊDO, J. B., MOURA, M. das C., RAMOS, M., *et al.* "Machine learning-based models to prioritize scenarios in a Quantitative Risk Analysis : An application to an actual atmospheric distillation unit", **Journal of Loss Prevention in the Process Industries**, v. 77, p. 104797, 2022. DOI: 10.1016/j.jlp.2022.104797. .

MACÊDO, J. B., RAMOS, P., MAIOR, C. B. S., *et al.* "Identifying low-quality patterns in accidents reports from textual data", **International Journal of Occupational Safety and Ergonomics**, p. 1–27, 2022. DOI: 10.1080/10803548.2022.2111847. Disponível em: https://doi.org/10.1080/10803548.2022.2111847.

MADEIRA, T., MELÍCIO, R., VALÉRIO, D., *et al.* "Machine Learning and Natural Language Processing for Prediction of Human Factors in Aviation Incident Reports", **Aerospace**, v. 8, n. 2, p. 47, 2021. DOI: 10.3390/aerospace8020047. .

MAIOR, C. B. S., MOURA, M. D. C., LINS, I. D. "Particle swarm-optimized support vector machines and pre-processing techniques for remaining useful life estimation of bearings", **Eksploatacja i Niezawodnosc**, 2019. DOI: 10.17531/ein.2019.4.10. .

MAIOR, C. B. S., MOURA, M. das C., SANTANA, J. M. M., *et al.* "Real-time SVM classification for drowsiness detection using eye aspect ratio". 2018. **Anais** [...] [S.l: s.n.], 2018.

MAIOR, C. B. S., SANTANA, J. M. M., DO NASCIMENTO, L. M., *et al.* "Personal protective equipment detection in industrial facilities using camera video streaming". 2018. **Anais** [...] [S.l: s.n.], 2018.

MAIOR, C. B. S., SANTANA, J. M. M., MOURA, M. das C., *et al.* "Automated Classification of Injury Leave based on Accident Description and Natural Language Processing". 2020. **Anais** [...] [S.l: s.n.], 2020. p. 1276–1281. DOI: 10.3850/978-981-14-8593-0_4559-cd.

MARCHIORI, D., GUIDA, S. Di. "Supplemental Material for Noisy Retrieval Models of Over- and Undersensitivity to Rare Events", **Decision**, v. 2, n. 2, p. 82–106, 2015. DOI: 10.1037/dec0000023.supp. .

MARHAVILAS, P. K., FILIPPIDIS, M., KOULINAS, G. K., *et al.* "The integration of HAZOP study with risk-matrix and the analytical-hierarchy process for identifying critical control-points and prioritizing risks in industry – A case study", **Journal of Loss Prevention in the Process Industries**, p. 103981, 2019. DOI: 10.1016/j.jlp.2019.103981. Disponível em: https://doi.org/10.1016/j.jlp.2019.103981.

MCKINNEY, W. "Data Structures for Statistical Computing in Python". 2010. **Anais** [...] [S.l: s.n.], 2010. DOI: 10.25080/majora-92bf1922-00a.

MENG, X., ZHU, J., FU, J., *et al.* "An accident causation network for quantitative risk assessment of deepwater drilling", **Process Safety and Environmental Protection**, v. 148, p. 1179–1190, 2021. DOI: 10.1016/j.psep.2021.02.035. .

MIN, K. B., SONG, S. H., MIN, J. Y. "Topic modeling of social networking service data on occupational accidents in Korea: Latent dirichlet allocation analysis", **Journal of Medical Internet Research**, v. 22, n. 8, p. 1–12, 2020. DOI: 10.2196/19222. .

MINAEE, S., KALCHBRENNER, N., CAMBRIA, E., *et al.* "Deep Learning Based Text Classification: A Comprehensive Review", **arXiv**, v. 54, n. 3, 2021. .

MOON, S., CHI, S., IM, S.-B. "Automated detection of contractual risk clauses from construction specifications using bidirectional encoder representations from transformers (BERT)", **Automation in Construction**, v. 142, p. 104465, out. 2022. DOI: 10.1016/j.autcon.2022.104465. .

MORENO, A., REDONDO, T. "Text Analytics : the convergence of Big Data and Artificial Intelligence", **International Journal of Interactive Multimedia and Artificial Intelligence**, v. 3, n. 6, p. 57–64, 2016. DOI: 10.9781/ijimai.2016.369. .

MOURA, M. D. C., AZEVEDO, R. V., DROGUETT, E. L., *et al.* "Estimation of expected number of accidents and workforce unavailability through Bayesian population variability analysis and Markov-based model", **Reliability Engineering and System Safety**, v. 150, p. 136–146, 2016. DOI: 10.1016/j.ress.2016.01.017. .

MUGURO, J. K., SASAKI, M., MATSUSHITA, K., *et al.* "Trend analysis and fatality causes in Kenyan roads: A review of road traffic accident data between 2015 and 2020", **Cogent Engineering**, v. 7, n. 1, 2020. DOI: 10.1080/23311916.2020.1797981. .

NATEGHI, R., AVEN, T. "Risk Analysis in the Age of Big Data: The Promises and Pitfalls", **Risk Analysis**, v. 41, n. 10, p. 1751–1758, 2021. DOI: 10.1111/risa.13682. .

NAYAK, R., PIYATRAPOOMI, N., WELIGAMAGE, J., *et al.* "Application of text mining in analysing road crashes for road asset". 2009. **Anais** [...] [S.l: s.n.], 2009. p. 49–50.

NBR, 14280. "NBR 14280:2000. Cadastro de acidente do trabalho - Procedimento e classificação.", **Nbr**, p. 94, 2001. .

OK, C., LEE, G., LEE, K. "Informative Language Encoding by Variational Autoencoders Using Transformer", **Applied Sciences (Switzerland)**, v. 12, n. 16, 2022. DOI: 10.3390/app12167968. .

OSMANI, A., MOHASEFI, J. B., GHAREHCHOPOGH, F. S. "Enriched Latent Dirichlet Allocation for Sentiment Analysis", **Expert Systems**, v. 37, n. 4, p. 1–31, 2020. DOI: 10.1111/exsy.12527. .

OTTER, D. W., MEDINA, J. R., KALITA, J. K. "A Survey of the Usages of Deep Learning for Natural Language Processing", **IEEE Transactions on Neural Networks and Learning Systems**, v. 32, n. 2, p. 604–623, 2021. DOI:

10.1109/TNNLS.2020.2979670. .

PAREJO-MOSCOSO, J. M., RUBIO-ROMERO, J. C., PÉREZ-CANTO, S. "Occupational accident rate in olive oil mills", **Safety Science**, 2012. DOI: 10.1016/j.ssci.2011.08.064. .

PASMAN, H., ROGERS, W. "How trustworthy are risk assessment results, and what can be done about the uncertainties they are plagued with?", **Journal of Loss Prevention in the Process Industries**, v. 55, p. 162–177, 2018. DOI: 10.1016/j.jlp.2018.06.004.This. .

PASSMORE, D., CHAE, C., KUSTIKOVA, Y., *et al.* "An exploration of text mining of narrative reports of injury incidents to assess risk", **MATEC Web of Conferences**, v. 251, 2018. .

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., *et al.* "Scikit-learn: Machine Learning in Python", v. 12, p. 2825–2830, 2011. DOI: 10.1007/s13398-014-0173-7.2. Disponível em: http://arxiv.org/abs/1201.0490.

PIMM, C., RAYNAL, C., TULECHKI, N., *et al.* "Natural Language Processing (NLP) tools for the analysis of incident and accident reports To cite this version : HAL Id : halshs-00953658 Natural Language Processing (NLP) tools for the analysis of incident and accident reports", n. April, 2014. .

PRAMOTH, R., SUDHA, S., KALAISELVAM, S. "Resilience-based Integrated Process System Hazard Analysis (RIPSHA) approach: Application to a chemical storage area in an edible oil refinery", **Process Safety and Environmental Protection**, v. 141, p. 246–258, 2020. DOI: 10.1016/j.psep.2020.05.028. Disponível em: https://doi.org/10.1016/j.psep.2020.05.028.

RACHMAN, A., RATNAYAKE, R. M. C. "Machine learning approach for risk-based inspection screening assessment", **Reliability Engineering and System Safety**, v. 185, n. February 2018, p. 518–532, 2019. DOI: 10.1016/j.ress.2019.02.008. Disponível em: https://doi.org/10.1016/j.ress.2019.02.008.

RADFORD, A., NARASIMHAN, K., SALIMANS, T., *et al.* **Improving Language Understanding by Generative Pre-Training**. 2018. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf. DOI: 10.4310/HHA.2007.v9.n1.a16.

RAMOS, M., LÓPEZ DROGUETT, E., MOSLEH, A., *et al.* "A human reliability analysis methodology for oil refineries and petrochemical plants operation: Phoenix-PRO qualitative framework", **Reliability Engineering and System Safety**, v. 193, n. September 2019, p. 106672, 2020. DOI: 10.1016/j.ress.2019.106672. Disponível em: https://doi.org/10.1016/j.ress.2019.106672.

RAMOS, M., THIEME, C., UTNE, I., *et al.* "A generic approach to analysing failures in human – System interaction in autonomy", **Safety Science**, v. 129, n. April, p. 104808, 2020. DOI: 10.1016/j.ssci.2020.104808. Disponível em: https://doi.org/10.1016/j.ssci.2020.104808.

RAMOS, P., MACÊDO, J. B., MAIOR, C. B. S., *et al.* "Combining BERT with numerical features to classify injury leave based on accident description", **Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability**, 2022. DOI: doi.org/10.1177/1748006X221140194. .

REED, S., ZOLNA, K., PARISOTTO, E., *et al.* "A Generalist Agent", **arXiv preprint**,

v. arXiv:2205, p. 1–40, 2022. Disponível em: http://arxiv.org/abs/2205.06175.

REHUREK, R., SOJKA, P. "Software Framework for Topic Modelling with Large Corpora", **Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks**, 2010. .

ROBINSON, S. D. "Temporal topic modeling applied to aviation safety reports: A subject matter expert review", **Safety Science**, v. 116, n. March, p. 275–286, 2019. DOI: 10.1016/j.ssci.2019.03.014. Disponível em: https://doi.org/10.1016/j.ssci.2019.03.014.

SALGUERO-CAPARROS, F., SUAREZ-CEBADOR, M., RUBIO-ROMERO, J. C. "Analysis of investigation reports on occupational accidents", **Safety Science**, v. 72, p. 329–336, 2015. DOI: 10.1016/j.ssci.2014.10.005. .

SARKAR, S. "Text Mining based Safety Risk Assessment and Prediction of Occupational Accidents in a Steel Plant". 2016. **Anais** [...] [S.l: s.n.], 2016. p. 439–444.

SARKAR, S., VERMA, A., MAITI, J., "Prediction of Occupational Incidents Using Proactive and Reactive Data : A Data Mining Approach". In: SPRINGER (Org.), **Industrial Safety Management**, Singapore, [s.n.], 2018. p. 65–79.

SARKAR, S., VINAY, S., RAJ, R., *et al.* "Application of optimized machine learning techniques for prediction of occupational accidents", **Computers & Operations Research**, v. 106, p. 210–224, jun. 2019. DOI: 10.1016/j.cor.2018.02.021. .

SARVESTANI, K., AHMADI, O., MORTAZAVI, S. B., *et al.* "Development of a predictive accident model for dynamic risk assessment of propane storage tanks", **Process Safety and Environmental Protection**, v. 148, p. 1217–1232, 2021. DOI: 10.1016/j.psep.2021.02.018. .

SCHWARZ, C. "ldagibbs: A command for topic modeling in stata using latent Dirichlet allocation", **Stata Journal**, v. 18, n. 1, p. 101–117, 2018. DOI: 10.1177/1536867x1801800107. .

SI, S., WANG, R., WOSIK, J., *et al.* "Students Need More Attention: BERT-based AttentionModel for Small Data with Application to AutomaticPatient Message Triage", p. 1–20, 2020. .

SILVA, J., JACINTO, C. "Finding occupational accident patterns in the extractive industry using a systematic data mining approach", **Reliability Engineering and System Safety**, v. 108, p. 108–122, 2012. DOI: 10.1016/j.ress.2012.07.001. Disponível em: http://dx.doi.org/10.1016/j.ress.2012.07.001.

SINGH, K., MAITI, J., DHALMAHAPATRA, K. "Chain of events model for safety management : Data analytics approach", **Safety Science**, v. 118, n. June, p. 568–582, 2019. DOI: 10.1016/j.ssci.2019.05.044. Disponível em: https://doi.org/10.1016/j.ssci.2019.05.044.

SINGLE, J. I., SCHMIDT, J., DENECKE, J. "Knowledge acquisition from chemical accident databases using an ontology-based method and natural language processing", **Safety Science**, v. 129, n. May, p. 104747, 2020. DOI: 10.1016/j.ssci.2020.104747. .

SJÖBLOM, O. "Data Mining in Promoting Aviation Safety Management". 2014. **Anais** [...] [S.l: s.n.], 2014. p. 186–187.

SPADA, M., BURGHERR, P., HOHL, M. "Toward the validation of a National Risk Assessment against historical observations using a Bayesian approach : application to the Swiss case Toward the validation of a National Risk Assessment against historical observations using a Bayesian approach : a", **Journal of Risk Research**, v. 22, n. 11, p. 1323–1342, 2019. DOI: 10.1080/13669877.2018.1459794. Disponível em: https://doi.org/10.1080/13669877.2018.1459794.

SRIVASTAVA, R., SINGH, P., RANA, K. P. S., *et al.* "A topic modeled unsupervised approach to single document extractive text summarization", **Knowledge-Based Systems**, v. 246, p. 108636, 2022. DOI: 10.1016/j.knosys.2022.108636. Disponível em: https://doi.org/10.1016/j.knosys.2022.108636.

STEIJN, W. M. P., VAN KAMPEN, J. N., VAN DER BEEK, D., *et al.* "An integration of human factors into quantitative risk analysis using Bayesian Belief Networks towards developing a 'QRA+'", **Safety Science**, v. 122, n. September 2019, p. 104514, 2020. DOI: 10.1016/j.ssci.2019.104514. Disponível em: https://doi.org/10.1016/j.ssci.2019.104514.

SUH, Y. "Sectoral patterns of accident process for occupational safety using narrative texts of OSHA database", **Safety Science**, v. 142, n. April 2019, p. 105363, 2021. DOI: 10.1016/j.ssci.2021.105363. Disponível em: https://doi.org/10.1016/j.ssci.2021.105363.

SUTSKEVER, I., VINYALS, O., LE, Q. V. "Sequence to sequence learning with neural networks", **Advances in Neural Information Processing Systems**, v. 4, n. January, p. 3104–3112, 2014. .

SWUSTE, P., THEUNISSEN, J., SCHMITZ, P., *et al.* "Process safety indicators, a review of literature", **Journal of Loss Prevention in the Process Industries**, v. 40, p. 162–173, mar. 2016. DOI: 10.1016/j.jlp.2015.12.020. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S095042301530098X.

TE, W., ADHITYA, A., SRINIVASAN, R. "Sustainability trends in the process industries : A text mining-based analysis", **Computers in Industry**, v. 65, p. 393–400, 2014. .

THEKDI, S. A., AVEN, T. "Risk analysis under attack: How risk science can address the legal, social, and reputational liabilities faced by risk analysts", **Risk Analysis**, p. 1–10, 2022. DOI: 10.1111/risa.13984. .

TIXIER, A. J. P., HALLOWELL, M. R., RAJAGOPALAN, B., *et al.* "Application of machine learning to construction injury prediction", **Automation in Construction**, v. 69, p. 102–114, 2016a. DOI: 10.1016/j.autcon.2016.05.016. Disponível em: http://dx.doi.org/10.1016/j.autcon.2016.05.016.

TIXIER, A. J. P., HALLOWELL, M. R., RAJAGOPALAN, B., *et al.* "Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports", **Automation in Construction**, v. 62, n. 2016, p. 45–56, 2016b. DOI: 10.1016/j.autcon.2015.11.001. .

TNO. **Purple Book - Guidelines for quantitative risk assessment**. 3. ed. The Hague, Committee for the Prevention of Disasters, 2005.

UYSAL, A. K., GUNAL, S. "The impact of preprocessing on text classification", **Information Processing and Management**, v. 50, n. 1, p. 104–112, 2014. DOI: 10.1016/j.ipm.2013.08.006. .

VAPNIK, V., IZMAILOV, R. "Rethinking statistical learning theory: learning using statistical invariants", **Machine Learning**, v. 108, n. 3, p. 381–423, 2019. DOI: 10.1007/s10994-018-5742-0. .

VASWANI, A., SHAZEER, N., PARMAR, N., *et al.* "Attention Is All You Need". 2017. **Anais** [...] [S.l: s.n.], 2017. p. 5998–6008.

VAYANSKY, I., KUMAR, S. A. P. "A review of topic modeling methods", **Information Systems**, v. 94, p. 101582, 2020. DOI: 10.1016/j.is.2020.101582. Disponível em: https://doi.org/10.1016/j.is.2020.101582.

VIJAYARANI, S., ILAMATHI, M. ., AND NITHYA, M. "Preprocessing Techniques for Text Mining - An Overview", **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2015. .

VILLA, V., PALTRINIERI, N., KHAN, F., *et al.* "Towards dynamic risk analysis : A review of the risk assessment approach and its limitations in the chemical process industry", **Safety Science**, v. 89, p. 77–93, 2016. DOI: 10.1016/j.ssci.2016.06.002. Disponível em: http://dx.doi.org/10.1016/j.ssci.2016.06.002.

VINNEM, J., RØED, W. **Offshore Risk Assessment**. 4. ed. [S.l.], Springer, London, 2020. v. 1.

WALLACE, E., WANG, Y., LI, S., *et al.* "Do NLP models know numbers? Probing numeracy in embeddings", **EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference**, p. 5307–5315, 2020. DOI: 10.18653/v1/d19-1534. .

WANG, N., AN, S., MAI, Q. "Space Engineering Risk Analysis from Risk Assessment Matrix Using Text Mining". 2016. **Anais** [...] [S.l: s.n.], 2016. p. 917–922.

WANG, Q., ZHANG, L., HU, J. "Real-time risk assessment of casing-failure incidents in a whole fracturing process", **Process Safety and Environmental Protection**, 2018. DOI: 10.1016/j.psep.2018.06.039. Disponível em: https://doi.org/10.1016/j.psep.2018.06.039.

WANG, S.-W., YU, D.-L. "Adaptive air-fuel ratio control with MLP network", **International Journal of Automation and Computing**, v. 2, n. 2, p. 125–133, 2005. DOI: 10.1007/s11633-005-0125-y. .

WEI, J., ZOU, K. "EDA: Easy data augmentation techniques for boosting performance on text classification tasks". 2020. **Anais** [...] [S.l: s.n.], 2020. DOI: 10.18653/v1/d19-1670.

WOLF, T., DEBUT, L., SANH, V., *et al.* "Transformers : State-of-the-Art Natural Language Processing". 2020. **Anais** [...] [S.l.], Association for Computational Linguistics, 2020. p. 38–45. Disponível em: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

WU, Y., SCHUSTER, M., CHEN, Z., *et al.* "Google ' s Neural Machine Translation System : Bridging the Gap between Human and Machine Translation", **arXiv preprint arXiv:1609.08144**, p. 1–23, 2016. .

XIANG, L. "Application of an Improved TF-IDF Method in Literary Text Classification", **Advances in Multimedia**, n. 3, p. 1–10, 2022. DOI: 10.1155/2022/9285324. .

XING, W., LEE, H. S., SHIBANI, A. "Identifying patterns in students' scientific argumentation: content analysis through text mining using Latent Dirichlet Allocation", **Educational Technology Research and Development**, v. 68, n. 5, p. 2185–2214, 2020. DOI: 10.1007/s11423-020-09761-w. Disponível em: https://doi.org/10.1007/s11423-020-09761-w.

XU, W., GUO, L., LIANG, L. "Mapping the academic landscape of the renewable energy field in electrical and electronic disciplines", **Applied Sciences (Switzerland)**, v. 10, n. 8, 2020. DOI: 10.3390/APP10082879. .

YAN, F., XU, K. "Methodology and case study of quantitative preliminary hazard analysis based on cloud model", **Journal of Loss Prevention in the Process Industries**, v. 60, p. 116–124, 2019. DOI: 10.1016/j.jlp.2019.04.013. .

YANG, Z., DAI, Z., YANG, Y., *et al.* "XLNet : Generalized Autoregressive Pretraining for Language Understanding", n. NeurIPS, p. 1–18, 2019. .

YIM, S., WARSCHAUER, M. "Web-based Collaborative Writing in L2 Contexts: Methodological Insights from Text Mining", **Language Learning & Technology**, v. 21, n. 1, p. 146–165, 2017. .

YOUNG, I. J. B., LUZ, S., LONE, N. "A systematic review of natural language processing for classification tasks in the field of incident reporting and adverse event analysis", **International Journal of Medical Informatics**, v. 132, n. August, p. 103971, 2019. DOI: 10.1016/j.ijmedinf.2019.103971. .

YUN, J., GEUM, Y. "Automated classification of patents: A topic modeling approach", **Computers & Industrial Engineering**, v. 147, p. 106636, set. 2020. DOI: 10.1016/j.cie.2020.106636. .

ZARE, P. "The Investigation of Multiple Product Rating Based on Data Mining Approaches", **Computer Engineering and Intelligent Systems**, v. 10, n. 5, p. 15–25, 2019. DOI: 10.7176/CEIS. .

ZENG, Z., ZIO, E. "A classification-based framework for trustworthiness assessment of quantitative risk analysis", **Safety Science**, v. 99, p. 215–226, 2017. .

ZHANG, L., WANG, J., WANG, Y., *et al.* "Automatic construction site hazard identification integrating construction scene graphs with BERT based domain knowledge", **Automation in Construction**, v. 142, p. 104535, out. 2022. DOI: 10.1016/j.autcon.2022.104535. .

ZHANG, X., GREEN, E., CHEN, M., *et al.* "Identifying secondary crashes using text mining techniques", **Journal of Transportation Safety & Security**, p. 1–21, 2019. DOI: 10.1080/19439962.2019.1597795. Disponível em: https://doi.org/10.1080/19439962.2019.1597795.

ZHANG, X., MAHADEVAN, S. "Ensemble machine learning models for aviation incident risk prediction", **Decision Support Systems**, v. 116, n. September 2018, p. 48–63, 2019. DOI: 10.1016/j.dss.2018.10.009. Disponível em: https://doi.org/10.1016/j.dss.2018.10.009.

ZHANG, Y., JIN, R., ZHOU, Z. H. "Understanding bag-of-words model: A statistical framework", **International Journal of Machine Learning and Cybernetics**, v. 1, n. 1–4, p. 43–52, 2010. DOI: 10.1007/s13042-010-0001-0. .

ZHOU, B., ZOU, L., MOSTAFAVI, A., *et al.* "VictimFinder: Harvesting rescue requests in disaster response from social media with BERT", **Computers, Environment and Urban Systems**, v. 95, p. 101824, jul. 2022. DOI:

10.1016/j.compenvurbsys.2022.101824. .

ZHOU, J., RENIERS, G. "A matrix-based modeling and analysis approach for fire-induced domino effects", **Process Safety and Environmental Protection**, v. 116, p. 347–353, 2018. DOI: 10.1016/j.psep.2018.02.014. .

ZHU, Y., KIROS, R., ZEMEL, R., *et al.* "Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books". 2015. **Anais** [...] [S.l: s.n.], 2015. p. 19–27.

ZIO, E. "The future of risk assessment", **Reliability Engineering and System Safety**, v. 177, n. March, p. 176–190, 2018. DOI: 10.1016/j.ress.2018.04.020. .

ZIO, E., AVEN, T. "Industrial disasters: Extreme events, extremely rare. Some reflections on the treatment of uncertainties in the assessment of the associated risks", **Process Safety and Environmental Protection**, v. 91, n. 1–2, p. 31–45, jan. 2013. DOI: 10.1016/j.psep.2012.01.004. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0957582012000055.