Combining BERT with Numerical Features to Classify Injury Leave Based on Accident Description

Plínio M. S. Ramos[a,b], July B. Macedo[a,b], Caio B. S. Maior[a,c], Márcio C. Moura[a,b] and Isis D. Lins[a,b]

[a]CEERMA - Center for Risk Analysis, Reliability and Environmental Modeling, Universidade Federal de Pernambuco, Brazil

[b] Universidade Federal de Pernambuco, Recife, Brazil, plinio.ramos@outlook.com, julybias@gmail.com, marcio.cmoura@ufpe.br, isis.lins@ufpe.br

[c] Technology Center, Universidade Federal de Pernambuco, Caruaru, Brazil, caio.maior@ufpe.br

Abstract: The occurrence of work accidents can threaten the health of workers and have managerial and financial consequences for organizations. In this context, accident investigation reports contain information that can support companies to propose preventive and mitigative measures and identify causes and consequences of injuring events. However, this information is frequently complex, redundant and/or incomplete. Additionally, a complete human review of the entire database is arduous, considering numerous reports produced by a company. Thus, Natural Language Processing (NLP)-based techniques are suitable for analyzing a large amount of textual information. In this paper, we adopted NLP techniques to determine whether an injury leave would be expected from a given accident report. The methodology was applied on accident reports collected from an actual hydroelectric power company using Bidirectional Encoder Representations from Transformers (BERT), a state-of-art NLP method. The text representations provided by BERT model were combined with numerical and binary features extracted from the accident reports. These combined features are fed into a Multilayer Perceptron (MLP) that predicts the occurrence of accidental leave for a given accident. After cross-validation, the results showed a median accuracy of 64.5%. Indeed, accident investigation reports provide useful knowledge to support decisions in the context of safety at managerial levels of the organization.

1. INTRODUCTION

Statistics provided by the International Labour Organization (ILO) demonstrate that working conditions in many countries have not changed to the point of significantly reduce the problem of occupational injuries [1], [2]. Occupational accidents pose negative consequences to employees, environment, and people surrounding the location where the event takes place. Characteristics related to occupational accidents are usually documented as accident reports [3]. Thus, accident reports contain information that can be exploited to prevent the occurrence of similar accidents and promote workplace safety, as well as predict any consequences of the events that have already occurred [4], [5].

According to the Brazilian National Institute of Social Security [6], the organization must pay the employee's salary within 15 days of leave, after which the government will pay the employee's social security benefit. However, for the company, it is still mandatory to collect the worker's guarantee fund, in addition to the 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 accident and recovery rates [7]. Machine Learning (ML) algorithms have been extensively applied for text classification [8]–[10]. In the context of accident reports, automatic and precise categorization of accident events would remove a cumbersome manual task from the current strategy for handling these reports [11].

Natural Language Processing (NLP) describes a wide range of mathematical and linguistic approaches to interpret, evaluate and generate human text and speech [12], [13]. As the NLP model's performance strongly depends on the quantity and quality of the annotated input data [14], representation-learning

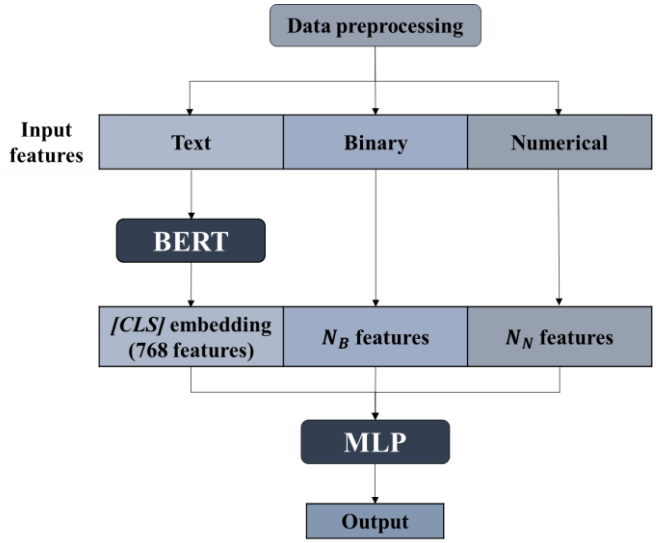
models, such as Bidirectional Encoder Representations from Transformers (BERT), have been shown to improve many NLP tasks, by employing unsupervised pre-training techniques to learn language representations from large-scale raw texts [15]. BERT was pre-trained by Google on an extremely large corpus (Wikipedia). During pre-training, the model learns the relation between words within a sentence and between sentences. Several pre-trained versions of the model are currently available. Thus, we can further train BERT to perform a supervised learning task by adding a database [16].

Using a 6-year historical database, this study aims to analyze a real accident investigation reports database from a Brazilian hydroelectrical company. These reports are structured based on the Brazilian Standard ABNT NBR 14280 - Workplace accident record [17] and contain different fields to characterize the accidents. Thus, the goal is to determine based on several accident characteristics, whether there will be an injury leave. The proposed model would be used in practice to process new accident reports, in which the safety technician would fill out an incident report and the model would provide as an output the injury leave or not of the employees. Managers can use the model to reorganize their workforce due to the loss of human resources in jobs, or even assist in the management of financial resources such as the need to pay leave or hire temporary employees. Here, we consider not only the accident description, which is written by the company's safety technician and by the injured employee, but also other numerical and categorical information presented in the database such as the use or not of PPE (Personal Protective Equipment) and the years of employee's work. The proposal is to use BERT model to process the text input, which, when concatenated with the non-textual inputs, feeds a classifier model. It is known that their pretraining techniques can then be used to fine-tune the models in NLP tasks with small data available [18]. The remainder of this article is organized as follows: Section 2 describes the proposed methodology, conceptualizing the techniques used throughout the paper. Section 3 presents the characteristics of the dataset as well as the applied methodology. Section 4 details the results of several analyses performed for the classification task. Section 5 concludes remarks and discusses the limitations and implications for further application.

2. METHODOLOGY

Accident reports are not usually restricted to the event description, they usually contains fields that need to be filled in with information regarding the injured subject, and other numerical or binary information to detail the accident. This information can be used to identify common causes, consequences and sources of accidents, as well as to propose preventive and mitigative measures. One common approach, when applying models to process natural language, such as BERT, is to combine numerical and categorical features of interest, with textual features by converting all features to text [16], [19]. However, 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 amount of discrete numbers [20]. Thus, an alternative approach is to concatenate the non-textual features for BERT embedding and classify the resulting vector through a neural network. A schematic overview of our proposed methodology is shown in Figure 1.

Figure 1. Overview of the proposed methodology.



The pre-trained BERT model is fine-tuned with text features extracted from the accident reports. However, when we deal with these reports, other features may be of interest to the problem. The objective is to select textual features (e.g., accident descriptions by safety technician and injured employee), numerical features (e.g., years of experience of the injured person, age of the worker) and binary features (e.g., the use of PPE, training in occupational safety) that can be used as predictor variables in a classification model. There are no feature limits, however, it is important to be aware of BERT's sequence length limit of 512 tokens.

Then, the texts are preprocessed to remove noise and then fed into BERT. Next, numeric, and binary features will be concatenated to the 768 standard BERT features that will be used to represent the textual variables. These combined features are input to a Multilayer Perceptron (MLP) for final classification. Each step was developed in the Python computational language and is detailed below.

2.1. Text Preprocessing

Textual data frequently present noise, such as different variations of capitalization for the same word, punctuation, and special characters. Once the dataset has been collected in a usable format, pre-processing of textual data can be initiated [21]. Hence, three preprocessing operations are applied to transform the input sentences into a cleaner format that can improve the learning process: (i) lowercasing, (ii) stop words filtering, and (iii) tokenization. Additionally, data augmentation (DA) was applied to obtain a larger and balanced training set of accident descriptions, aiming to improve the performance of the classifier as more data are available [22].

The lowercasing and stop word filtering are implemented using regular expression operations and *NLTK* library [23], the tokenization was performed using the tokenizer provided by *transformers* library [24], and the DA was implemented using *nlpgaug* [25], which is a library dedicated to textual augmentation in ML experiments. Simply put, replacing 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 which performs word replacement from the large lexical database WordNet. To train and test the classifiers, we split each category into an 80/20 training/testing ratio.

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 [26]. Next, stop words filtering

was used to identify the content information, in which non-informative terms are removed (e.g., 'the', 'it', and 'is'). 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]*.

Furthermore, 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 strings to a maximum length by adding zeros, since the model requires inputs that have the same shape and size.

2.2. BERT and MLP

In its raw format, the corpus is useless for algorithms that work on numeric feature spaces. For this reason, it must be preprocessed and converted into a standard format for distilling knowledge (e.g., as input to an ML model), which consists of discovering patterns and identifying information through the application of different mechanisms on the vectorized corpus [27]. Moreover, processed texts need to be converted into a numerical representation, a feature vector that can be used as inputs for supervised and/or unsupervised algorithms. However, obtaining high-quality word representations is quite challenging because they should represent the syntax, semantics, and context of a word [28].

BERT consists of a multi-layer transformer encoder; transformer is an architecture developed by Vaswani [29] entirely based on attention mechanisms that learn contextual relations between words in a text. 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 to improve 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 untrained layers of neurons on top of the pre-trained model. In general, during fine-tuning, the pre-trained parameters are adopted to initialize the model and then they are fine-tuned using specific labeled data to solve the supervised task.

2.3. Modeling Process

We used the *Pytorch* implementation of pre-trained BERT available at transformers library [24], which provides Application Programming Interfaces (APIs) to quickly download and use several pre-trained models on a given text, and thus fine-tune them on our own datasets. We here built our model on top of the '*bert-base-multilingual-cased*' version of the '*BertForSequenceClassification*' model that is a version pre-trained in 104 languages (including Portuguese, in which our text data are originally written).

The MLP implemented has 770 nodes in the input layer (dimension of our combined input vector) and the dimension of the MLP layers was 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).

We apply a cross-validation (CV) algorithm based on the leave-*P*-out cross-validator (LPO-CV) [30], 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. Unlike K-Fold, the test sets will overlap for $p > 1$.

From this, a confusion matrix with the prediction of the test data for each run was formed, in order to assist in the performance evaluation of the models. The classifier's performance has been evaluated considering the classification accuracy, A, as seen in Equation (*1*:

$$A = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP, TN, FP and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. Thus, the next section will present the results.

3. APPLIED METHODOLOGY

The database is presented in the form of a spreadsheet, where rows correspond to an accident investigation report and columns are characteristics about the event itself (e.g., location, causes, accident agent, consequences, financial impact) and employee involved in the event (e.g., job position, experience in the activity, training, injury leave). The dataset contains 626 reports that describe, in Portuguese, accidental events that occurred in a 6-year period.

In this database, the safety technician grouped the accidents based on NBR 14280 [17]. The high number of characteristics may lead to believe that the descriptions of the events are detailed. However, there are some gaps in filling throughout the reports. In addition, the lack of standardization in the description of characteristics can hinder the efficient use of the accident investigation report database to support decision-making for risk management.

After evaluating all the features present in the reports, we consider that four of them represent a relevant source of information for the classifier: (i) accident description (provided as free text by safety technician), (ii) accident description (provided as free text by injured employee), (iii) use of PPE during the accident (binary variable), and (iv) the employee's length of experience (continuous variable). Thus, we removed the accident samples that did not present these features filled in the report. The binary feature assumes 0 value if the employee was not using PPE and 1 otherwise. The continuous variable was scaled between 0 and 1. Finally, the text data were preprocessed as described in the following section.

3.1. Preprocessing Dataset

After the feature selections, it is necessary to preprocess the textual features. First, lowercasing and, stop words filtering procedures were performed. Here, after the 80/20 split, the DA procedure was applied to the test set to: (i) balance the injury leave category; (ii) to increase the dataset. Therefore, the training set consists of 1000 samples among augmented and original descriptions (495 original), whereas the test set contains 131 non-augmented descriptions (78 reports of injury leave and 53 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. 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 1 shows the post tokenization sentence.

Table 1. Example of tokenization procedures in an adapted description.

Procedures	Description
Tokenization	[CLS] [the] [employee] [was] [working] [in] [the] [warehouse] [when] [he] [tripped] [and] [fell] [fracturing] [his] [right] [arm] [SEP]

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 features will go to the BERT which will be concatenated with the other features, and finally feed the MLP classifier.

3.2. BERT and MLP Applied

Finally, the processed texts are converted into a numerical representation using BERT. Thus, we can feed the ML model with the concatenated vector, as shown in Figure 2. The first tensor represents the textual feature after tokenization and encoding, the second tensor refers to the binary feature, and finally, the third tensor represents the numeric feature after normalization. These combined features 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 features as text and feeding it through BERT, the numerical/binary features are treated separately to improve classifier performance.

Figure 2. Tokenization and encoding concatenated vector

```
(tensor([  101,  9765, 12462,  2139,  6019,  2213, 21877,  4135, 11320,  6843,
          2139,  3430,  4830,  7367,  3207,  7166,  2080,  8529,  2050,  7151,
          2063,  6643,  6873,  4748, 25300, 20528, 29068,  2050,  4012,  9808,
          4569, 10446, 16843,  2015,  1041, 20118,  3413,  2906, 11498,  1051,
          4592,  8695, 16021,  9080, 22684,  2229,  4787, 24501, 26029,  3736,
         15985, 21877,  4135,  2695,  2080,  1042, 10448, 24485,  2906,  1037,
          3417,  2050,  2139,  6019,  2213,  2079, 28352, 20808,  1041,  9765,
          2050,  4078, 28139, 13629,  8557,  1041, 29080,  2226, 17540,  2890,
          1051, 21877, 25325, 13094,  3527,  2079,  7861, 28139,  3654,  3527,
          1037,  9530, 14808,  7113,  4078,  7011, 14550, 10696,  2140, 10861,
          9530, 18886,  8569, 17922, 11498,  1037,  1051, 27108,  7389,  7405,
          2079,  5648, 15781,  8081, 19629,  2080,  2139,  4372, 13186,  3406,
          4830,  3417,  2050,  2061,  1042, 10448,  2566,  3401, 17062,  2050,
         24110,  3527,  1037,  2033, 26212,  9765, 12462,  7367, 14017,  5794,
          3527,  2079,  4372,  3540,  7646,  2063,  1041, 11557,  3527,  7861,
         18704, 20808, 20118,  5650, 19170,  2079, 15270, 12821, 26467,  1051,
         15270, 12821, 26467,  9765, 12462,  8823, 15482,   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,     0,     0,     0,     0,     0,     0,
             0,     0,     0,     0,     0,     0,     0,     0,     0,     0]),
 tensor([0.]),
 tensor([0.5805]),
```

4. RESULTS AND DISCUSSIONS

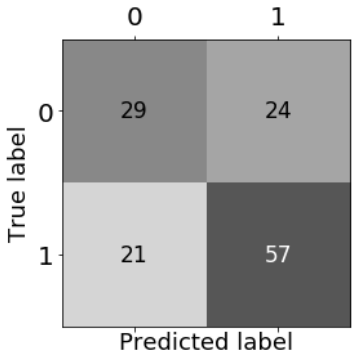
We considered 10 different training/validation sets, randomly split in a proportion of 90/10, to avoid biases and to account for the variability. It is noteworthy that, unlike k-fold CV, the validation set samples may overlap considering the different split of the complete set. Considering this, we computed the median and standard deviation for the accuracy of the test set classification, seen in Table 2.

Table 2. Median and standard deviation for the accuracy test set.

Median	Standard deviation
64.5 %	1.64

Despite the median being around 64%, one of the models (rounds) achieved the best test accuracy of 66%. In fact, seven of the ten (7/10) rounds had test results greater than or equal to 64%, with the smallest four achieving close results (62%). Figure 3 shows the confusion matrix for the results of one round of cross-validation on the test set, where 0 represents no injury leave and 1 represents that there was a leave.

Figure 3. Confusion matrix for classification of test data with the best result.



As one can see, there were 24 FP and 21 FN in the test set classification, which means that 24 accidents that did not lead to injury leave were erroneously classified as accidents that led to injury leave; meanwhile 21 accidents that led to injury were misclassified as accidents that did not lead to injury leave. For false positives, time and resources for planning could be spent unnecessarily. However, the plans would only be put into practice when there was the injury leave. The false negative, on the other hand, would take more time to reorganize tasks with the worker absent, as employees would be expected to return to their jobs, which in fact did not happen.

However, there is a need for curation of data, a restructuring of the database and clarification in the completion of reports, especially in fields with dubious meanings. Thus, a model that correctly predicts may provide useful information that can support managers to propose a plan to effectively deal with the worker leave.

5. CONCLUSION

In this paper, we inferred the occurrence of injury leaves for occupational accidents based on the analysis of a dataset of accident investigation reports from a hydroelectric power company. More specifically, we propose an approach that considers, in addition to the accident description, 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 features were combined, and the resulting vector was trained by the MLP classifier.

The results with CV achieved 64.5% and 1.64% of median accuracy and standard deviation, respectively. The best performance reached 66% of accuracy. In fact, all models (rounds) showed accuracy above 60%. Thus, the correct prediction of the injury leaves can provide useful information that helps managers effectively deal with the worker's absence and its consequences (costs, work replanning). However, here, the lack of standardization of the descriptions and 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. Indeed, it was possible to trace the low quality of the database with the performance of the model. Another point is that difficulties in filling out the reports are a common problem already mentioned by the safety technician, in which some fields seem to be confused about their meaning. The high number of factors may lead to believe that the descriptions of the events are minutely detailed, however, filling of the long report makes the process of documenting the accidental event boresome.

In this work, even using the DA procedure to balance the injury leave category, the results of the different models reached an accuracy lower than 70%. One possible reason is that accident reports are

grouped into nine different accident agent classes. These classes, which represents the object, substance, or environment to which the unsafe condition is related, and which caused the accident, is a source of information to identify common elements associated to accidents. Thus, accidents of certain classes occurred a minimum number of times, making this event unprecedented and causing the model to classify it based on other reports, presenting a different severity than what actually happened.

In future research, we intend to analyze the same approach for some of these specific classes. In addition, we will consider a broader problem, such as predicting the number of days of injury leave and accident costs associated with injury leaves. As recommendations, there is a need to improve procedures, especially in filling out the report, making it more streamlined and elucidating the difference between the fields. In addition, training on how to fill out the accident reports and a possible review in order to identify missing points or misspelled words.

Acknowledgements

The authors thank the National Agency for Research (CNPq), the Foundation of Support for Science and Technology of Pernambuco (FACEPE), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and the 'Programa de Recursos Humanos (PRH) da Agência Nacional de Petróleo (ANP) and Finep (Brazilian Innovation Agency) - PRH-ANP 38.1: Risk Analysis and Environmental Modeling in Exploration, Development and Production of Oil and Gas for the financial support through research grants.

References

[1] N. Cavazza and A. Serpe, "Effects of safety climate on safety norm violations: exploring the mediating role of attitudinal ambivalence toward personal protective equipment," *J. Safety Res.*, vol. 40, no. 4, pp. 277–283, 2009, doi: 10.1016/j.jsr.2009.06.002.

[2] A. Haeri, "Analyzing safety level and recognizing flaws of commercial centers through data mining approach," *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, vol. 234, no. 3, pp. 512–526, 2020, doi: 10.1177/1748006X19889812.

[3] M. Bevilacqua and F. E. Ciarapica, "Human factor risk management in the process industry: A case study," *Reliab. Eng. Syst. Saf.*, vol. 169, pp. 149–159, Jan. 2018, doi: 10.1016/j.ress.2017.08.013.

[4] F. Abdat, S. Leclercq, X. Cuny, and C. Tissot, "Extracting recurrent scenarios from narrative texts using a Bayesian network: Application to serious occupational accidents with movement disturbance," *Accid. Anal. Prev.*, vol. 70, pp. 155–166, Sep. 2014, doi: 10.1016/j.aap.2014.04.004.

[5] L. Zhang, H. Wang, Q. Meng, and H. Xie, "Ship accident consequences and contributing factors analyses using ship accident investigation reports," *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, vol. 233, no. 1, pp. 35–47, Feb. 2019, doi: 10.1177/1748006X18768917.

[6] INSS, "Auxílio-doença," 2022. https://www.gov.br/inss/pt-br/saiba-mais/auxilios/auxilio-doenca.

[7] J. M. Parejo-Moscoso, J. C. Rubio-Romero, and S. Pérez-Canto, "Occupational accident rate in olive oil mills," *Saf. Sci.*, 2012, doi: 10.1016/j.ssci.2011.08.064.

[8] R. Rivas, N. Montazeri, N. X. T. Le, and V. Hristidis, "Automatic classification of online doctor reviews: Evaluation of text classifier algorithms," *J. Med. Internet Res.*, 2018, doi: 10.2196/11141.

[9] I. Ahmed, R. Ali, D. Guan, Y.-K. Lee, S. Lee, and T. Chung, "Semi-supervised learning using frequent itemset and ensemble learning for SMS classification," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1065–1073, Feb. 2015, doi: 10.1016/j.eswa.2014.08.054.

[10] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," in *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Jul. 2015, pp. 136–140, doi: 10.1109/ICCI-CC.2015.7259377.

[11] H. P. Evans *et al.*, “Automated classification of primary care patient safety incident report content and severity using supervised machine learning (ML) approaches,” *Health Informatics J.*, vol. 26, no. 4, pp. 3123–3139, Dec. 2020, doi: 10.1177/1460458219833102.
[12] S. D. Robinson, “Temporal topic modeling applied to aviation safety reports: A subject matter expert review,” *Saf. Sci.*, vol. 116, no. March, pp. 275–286, 2019, doi: 10.1016/j.ssci.2019.03.014.
[13] C. van Gulijk, P. Hughes, M. Figueres-Esteban, R. El-Rashidy, and G. Bearfield, “The case for IT transformation and big data for safety risk management on the GB railways,” *Proc. Inst. Mech. Eng. Part O J. Risk Reliab.*, vol. 232, no. 2, pp. 151–163, 2018, doi: 10.1177/1748006X17728210.
[14] C. B. S. Maior, M. das C. Moura, and I. D. Lins, “Particle swarm-optimized support vector machines and pre-processing techniques for remaining useful life estimation of bearings,” *Eksploat. i Niezawodn. - Maint. Reliab.*, vol. 21, no. 4, pp. 610–619, Sep. 2019, doi: 10.17531/ein.2019.4.10.
[15] F. Li, Y. Jin, W. Liu, B. P. S. Rawat, P. Cai, and H. Yu, “Fine-tuning bidirectional encoder representations from transformers (BERT)–based models on large-scale electronic health record notes: An empirical study,” *J. Med. Internet Res.*, 2019, doi: 10.2196/14830.
[16] J. B. Macêdo, M. das Chagas Moura, D. Aichele, and I. D. Lins, “Identification of risk features using text mining and BERT-based models: Application to an oil refinery,” *Process Saf. Environ. Prot.*, vol. 158, pp. 382–399, Feb. 2022, doi: 10.1016/j.psep.2021.12.025.
[17] 14280 NBR, “NBR 14280:2000. Cadastro de acidente do trabalho - Procedimento e classificação.,” *Nbr*, p. 94, 2001.
[18] S. Si *et al.*, “Students Need More Attention: BERT-based AttentionModel for Small Data with Application to AutomaticPatient Message Triage,” pp. 1–20, 2020, [Online]. Available: http://arxiv.org/abs/2006.11991.
[19] C. B. S. Maior, J. M. M. de Santana, M. das C. Moura, and I. D. Lins, “Automated Classification of Injury Leave based on Accident Description and Natural Language Processing,” in *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference*, 2020, pp. 1276–1281, doi: 10.3850/978-981-14-8593-0_4559-cd.
[20] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner, “Do NLP models know numbers? Probing numeracy in embeddings,” *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 5307–5315, 2020, doi: 10.18653/v1/d19-1534.
[21] J. Pence, P. Farshadmanesh, J. Kim, C. Blake, and Z. Mohaghegh, “Data-theoretic approach for socio-technical risk analysis: Text mining licensee event reports of U.S. nuclear power plants,” *Saf. Sci.*, vol. 124, no. November 2019, p. 104574, 2020, doi: 10.1016/j.ssci.2019.104574.
[22] C. B. S. Maior, J. M. M. Santana, I. D. Lins, and M. J. C. Moura, “Convolutional neural network model based on radiological images to support COVID-19 diagnosis: Evaluating database biases,” *PLoS One*, vol. 16, no. 3, p. e0247839, Mar. 2021, doi: 10.1371/journal.pone.0247839.
[23] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. O’Reilly Media, Inc., 2009.
[24] T. Wolf *et al.*, “Transformers : State-of-the-Art Natural Language Processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45.
[25] E. Ma, “nlpaug Documentation,” 2021, [Online]. Available: https://nlpaug.readthedocs.io/en/latest/index.html.
[26] A. K. Uysal and S. Gunal, “The impact of preprocessing on text classification,” *Inf. Process. Manag.*, vol. 50, no. 1, pp. 104–112, 2014, doi: 10.1016/j.ipm.2013.08.006.
[27] B. Bengfort, R. Bilbro, and T. Ojeda, *Applied Text Analysis with Python: Enabling Language-aware Data Products with Machine Learning*. O’Reilly Media, Inc, 2018.
[28] R. Feldman and J. Sanger, *The text mining handbook: advanced approaches in analyzing unstructured data*. Cambridge University Press, 2007.

[29] A. Vaswani *et al.*, “Attention Is All You Need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[30] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011, doi: 10.1007/s13398-014-0173-7.2.