Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_driver

Natural Language Processing Tool for Identifying Influencing Factors in Human Reliability Analysis and Summarizing Accident Reports

Karl Johnson

Civil and Environmental Engineering, University of Strathclyde, United Kingdom, karl.johnson@strath.ac.uk

Caroline Morais

Agency for Petroleum, Natural Gas and Biofuels (ANP), Brazil, cmorais@anp.gov.br

Edoardo Patelli

Civil and Environmental Engineering, University of Strathclyde, United Kingdom, edoardo.patelli@strath.ac.uk

The development of a tool based on Natural Language Processing (NLP) models is presented. The presented tool is an improvement on the original virtual human factors classificator developed to assist experts with extracting the organizational, technological, and individual factors that may trigger human errors. To identify the performance shaping factors, the approach proposed is based on classifying text according to previously labelled accident reports by human experts, making use of BERT (Bidirectional Encoder Representations from Transformers), a popular transformer-based machine learning model for NLP.
In addition, a method to provide a summarization of each accident report is presented. This provides further detailed context alongside with the identified performance shaping factors, without the need of reading the entire report which is generally a significant task. The tool performs abstractive summarization as it aims to understand the entire report and generate paraphrased text to summarize the main points. In this work, BART (Bidirectional and Auto-Regressive Transformers), which is a denoising autoencoder for pre-training sequence-to-sequence models, has been used as the basis for the text summarization model.

Keywords: Accident Report Data, Natural Language Processing, Human Factors, Human Reliability Analysis, .

1. Introduction

Various human reliability analysis (HRA) approaches have been developed to aid in the incorporation of the human contribution to risk into overall system safety analysis. Performance shaping factors (PSFs) are factors which may have positive or negative influence on human performance these include organizational, technological, and personal factors (C. Morais, et al. 2022). Understanding the contribution and interactions of these factors is a key step in the process, aiding in the design processes, helping prevent accidents, and in turn improve overall safety. (Griffith and Mahadevan 2011) (Groth and Mosleh 2012).

There are significant learning opportunities with regards to the contributions and interactions between such factors and human errors from past major accident events across different industrial sectors. With this opportunity in mind Multi-Attribute Technological Accidents Dataset (MATA-D) was created (Moura, et al. 2016).

The process of expanding this dataset should be constant, not only to decrease epistemic uncertainty in human reliability data but also to reflect changes in human behaviour due to evolving technology. However, reading and analyzing such reports is a time-consuming process, taking multiple days to read and assess an entire report. This means the rate at which data can be extracted is limited, delaying learning opportunities, whilst also taking resources away from more critical and analytical processes (C. Morais, et al. 2022). Morais et al (2022) have developed a tool based on Natural Language Processing (NLP) models, to provide users a more efficient way to identify the organizational, technological, and human factors from accident reports, and in turn aid in the expansion of the MATA-D. The approach was based on classifying

This is a peer-reviewed, version of record of the following paper: Johnson, K., Patelli, E., & Morais, C. (2023). Natural language processing tool for identifying influencing factors in human reliability analysis and summarizing accident reports. In M. P. Brito, et al (Eds.), 33rd European Safety and Reliability Conference (ESREL 2023) Article P294 Research Publishing. https://doi.org/10.3850/978-981-18-8071-1_P294-cd

text according to the previously labelled accident reports by human experts. However, the work was based on simple NLP models such as Support Vector Machine (SVM) and bag-of-words, which impose limitations to the user such as the size of text and lack of aids to check the results without reading the full report. Here a popular transformer-based machine learning model for NLP known as BERT (Bidirectional Encoder Representations from Transformers) (Horev 2018) is applied which improves on the performance of the original classificator.

In addition, a tool to provide a summarization of the human role in each accident report is presented. This provides interested parties with further context and evidence, beyond the list of identified performance shaping factors, without the significant task that is reading the entire report. The tool performs abstractive summarization as it aims to understand the entire section and generate paraphrased text to summarize the main incidents. In this work, BART, a denoising autoencoder for pre-training sequence-to-sequence models, has been used as the basis for the text summarization model (Lewis, et al. 2020).

2. Background

This section discusses previous similar efforts, as well as the background ideas and models that support the work presented here.

2.1. Virtual Raphael – SVM approach

Previous works have demonstrated the possibility to automate the identification of human errors and influencing factors based on the MATA-D (C. Morais, et al. 2022) The approach has resulted in the development of an open source computational tool named ”Virtual Raphael” (C. Morais, et al. 2022), (Morais, Yung and Patelli 2019). In this approach the texts are converted into a bag-of-words objects, these together with the MATA-D entries are used as the inputs to construct a SVM model. SVM is a supervised machine learning algorithm, the main objective of an SVM is to find an optimal hyperplane that separates different classes or groups of data points in a feature space. In a binary classification problem, the hyperplane acts as a decision boundary, maximizing the margin between the two classes. The data points closest to the decision boundary are known as support vectors, which are crucial for defining the hyperplane (Osuna, Freund and Girosi 1997). This was then tested on 20% of the available accident reports excluded from the training set and compared with the expert classified entries in the MATA-D, giving the performance metrics shown in (C. Morais, et al. 2022).

Table 1 Performance Metrics Human Factors Virtual Classifier SVM Approach, adapted from *(C. Morais, et al. 2022)*

Metric	
Accuracy	86%
Precision	60%
Recall	46%
F1 score	52%

2.2. Natural language processing

Natural Language Processing (NLP) is a field of artificial intelligence that is concerned with giving computers the ability to understand, interpret and generate language, either in text or spoken forms, in much the same way a human being does (Ghazizadeh and Zhu 2020). NLP can therefore be used help to automate tasks that would otherwise require human intervention.

In NLP, human language is separated into fragments so that the grammatical structure of sentences and the meaning of words can be analyzed and understood in context. This helps computers read and understand spoken or written text in the same way as humans (Wolff 2020). The NLP pre-processing tasks include tokenization, breaking down text into smaller semantic units or single clauses. A token is an instance of a sequence of characters that are grouped together as a useful semantic unit for processing, whether these are individual words or short phrases. Stemming and lemmatization, this is standardizing words by reducing them to their root forms and sometimes the removal of Stop words, filtering out common words that add little or no unique information (Wolff 2020). There are two main algorithms you can use to solve NLP problems, rule based and machine learning algorithms. Rule-based systems rely on sets of grammatical rules that need to be created by experts in linguistics, or knowledge engineers. These were the original approaches used to craft NLP algorithms, and they are still popular**.** Machine learning models, on the other hand, are based on statistical methods and learn to perform tasks after being given examples. Machine learning

algorithms are fed training data and expected outputs to train machines to make associations between a particular input and its corresponding output. Using statistical analysis models build their own understanding and discern which features best represent the texts, before making predictions for unseen data (Ghazizadeh and Zhu 2020).

In the development of the tools presented in this work, machine learning approaches are preferred, more specifically the BERT and BART models which are used for pre-training before fine-tuning the model on the specific tasks.

2.2.1. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a neural network-based approach to pretraining language models that has achieved impressive results in a number NLP tasks, including text classification (Devlin, et al. 2018). BERT makes use of Transformer, an attention mechanism that learns contextual relations between words in a text (Horev 2018). Transformer includes two separate mechanisms, an encoder that reads the text input and a decoder that produces a prediction for the task. However, as BERT was developed as a language model, only the encoder mechanism is necessary (Vaswani, et al. 2017).

As opposed to directional models, which read the text input sequentially (left-to-right or right-to-left), the Transformer encoder reads the entire sequence of words at once. Therefore, it is considered bidirectional, though it could also be said to be non-directional (Horev 2018). This means the model can consider both the left and right context of a word when making predictions. Allowing it to better understand the meaning and context of words and phrases within a sentence or document.

BERT is pre-trained on large amounts of unlabelled text data using an unsupervised learning approach, including the BooksCorpus and English Wikipedia, which contains over 3 billion words (Horev 2018). BERT therefore has an understanding of complex patterns and relationships between words and phrases in general language. This transfer learning approach can therefore reduce the amount of pre-labelled data required for fine-tuning a model. This is one of the major advantages for this work due to the limited availability of pre-labelled accident reports. The main drawback of BERT in this application is that it was trained on sequences limited in tokens, therefore the inputs are constrained by this. However, by identifying key sections and pre-processing to remove common words, this restrictions impact is reduced.

2.2.2. BART

Similar to BERT, BART (Bidirectional and Auto-Regressive Transformers) is based on the Transformer architecture. BART is a sequence-to-sequence (seq2seq) model that can both encode and decode text. (Lewis, et al. 2020). This means that BART can be used for text generation, and therefore text summarization tasks. BART is also pre-trained on large amounts of text. However, BART was trained on a combination of two unsupervised tasks, denoising autoencoding and sequence-to-sequence pretraining (Lewis, et al. 2020). Denoising autoencoding involves corrupting the input text by randomly masking some of the tokens and then training the model to reconstruct the original text from the corrupted input. Sequence-to-sequence pretraining involves training the model to predict the output sequence given the input sequence. This task involves training the model to understand the relationships between the input and output sequences.

These tasks lead to a model that can capture the meaning and context of the text, even in the presence of missing tokens, and generate high-quality output sequences that are consistent with the input. This combination gives BART, its ability to generate high-quality summaries. In a similar way to BERT, BART was trained on sequences a maximum of 1024 tokens, which restricts how larger section can be summarized at once.

3. Implementation

The section will discuss the background implementation of tools and their performance.

3.1. Dataset

The Multi-Attribute Technological Accidents Dataset (MATA-D) is a collection of 238 major accidents from a range of different industries considered to be of similar complexity, including aviation, chemical, oil & gas, nuclear, waste treatment etc., allowing the conceptual advantage of cross-learning from different industrial sectors! "# $%&(!)*!'+!-./01¦The accident reports for these incidents were then analysed by an expert

focused on the contributing human factors, classified using the CREAM (Cognitive Reliability and Error Analysis Method) framework! "2$+8' 4) +! /5561,! The CREAM taxonomy is comprised of human errors and performance shaping factors including organizational, technological, and individual factors. Of the 238 major accidents, 110 of the accident reports are currently publicly available and included in the training and testing of tool "# $& 78!) *!' +,(!- . - - 1,!

3.2. *Classificator tool*

The developed classificator tool, will be referred as "Virtual Raphael – BERT", to show it is a further development of the original virtual classificator from Section 2.1.

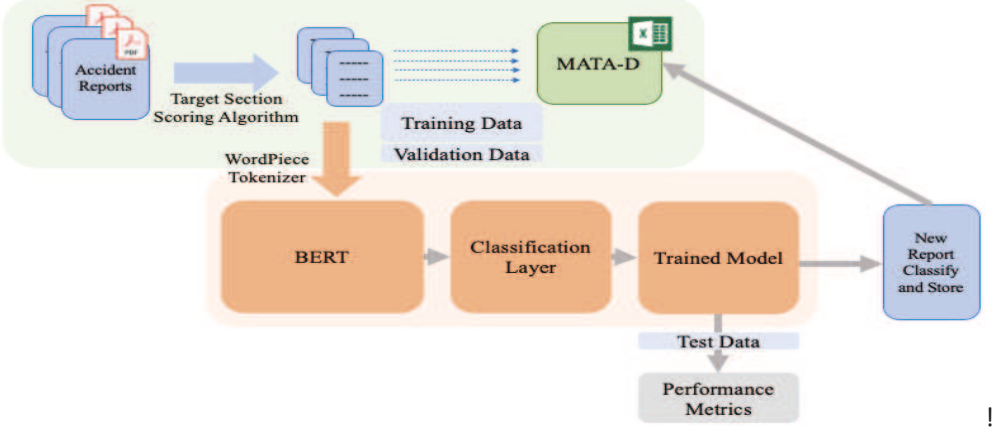
The process is outlined in Johnson et al. 2023 (Johnson, Morais and Patelli 2023) and summarised in Figure 1. Target sections are extracted and then tokenized using the same scheme used to pretrain the BERT model, which is called WordPiece tokenization (Khanna 2021). The text data is then combined with the labels from MATA-D. The model splits the training data into its training and test sets, that are used to train the classification layer added on top of the pre-trained BERT model. Which is then optimized using stochastic gradient descent, with a binary cross-entropy loss function, where the hyperparameters are fine-tuned based on performance on the test set. The model is then saved, so that its performance metrics on the validation set can be tested, and so that it can then be used in the future for new accident reports, which must first be processed through the same tokenizer, to be classified. If the tokenized document is too long, this is separated into different sections which then go through the model, before the outputs being aggregated together. The final output is the identified factors, and a binary array corresponding to all 53 factors that can be used to add the incident to the MATA-D.

3.2.1. *Classificator performance*

The benchmark for this tool's performance were the metrics obtained by the approach discussed in section 2.1. The performance metrics obtained by the BERT approach are reported in Table 2. This shows the average of the performance metrics based on the validation set of the available reports. This tool will allow the expansion of the MATA-D at a faster rate (approximately one minute per report) whilst maintaining a performance and accuracy, that can be considered more in line with the performance (in terms of classification) from a human expert.

Table 2 Performance Metrics Virtual Human Factors Classifier BERT Approach

Metric	
Accuracy	91%
Precision	88%
Recall	77%
F1 score	82%



!

Figure 1 Workflow of Classificator Tool

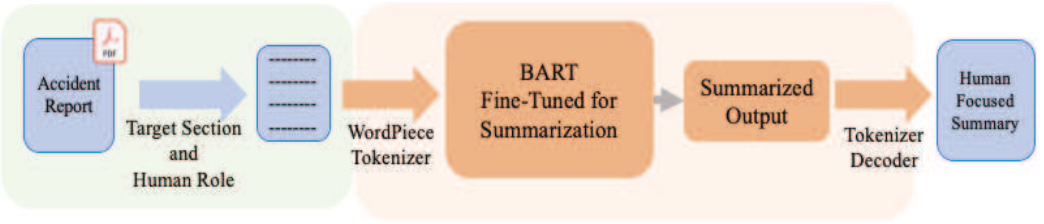


Figure 2 Workflow of Summarization Tool

3.3. Summarization tool

The background process for the summarization tool is shown in Figure 2, and outlined below. The first step in the model training for the *Summarization tool* is the same as for the *classificator tool*. The target section, 'recommendations' and 'lessons learned', within the report are identified. However, additionally from the entire report any sentences containing pronouns, or human related nouns (such as user, operator, manager etc) are extracted and stored together with the identified target section. The number of words is then checked, if this is greater than 1024, the text is split into sections, which then leads to multiple summarized sections being produced.

The text is tokenized using the pre-trained BART tokenizer again from Hugging Face Transformers library on Python (Abid, Carrigan, et al., BART 2021). This can then be fed into BART for summarization. BART works by producing multiple possible summarizations, control by the number of beams parameter, and then selects the one with the highest score based on the language model probabilities (Lewis, et al. 2020). The maximum length of the summary can also be controlled, for the accident reports this was set to 250 tokens, per section if the text required to be split.

The summary produced by the model is then converted back into human-readable text by using the tokenizer decoding method.

3.3.1. Summarization tool performance

When discussing the performance of a summarization tool there are multiple factors that may need to be considered, such as quality, relevance, efficiency, length and domain specific performance. There are several evaluation metrics that can also be used such as, ROUGE (Recall-Oriented Understudy for Gisting Evaluation), BLEU (Bilingual Evaluation Understudy), and to assess the quality of the generated summaries (Lin 2004), (Papineni, et al. 2002). These metrics compare the generated summary against one or more reference summaries and compute a score based on their similarity. In this application, there are no reference summaries to compare to, as the model has not been fine-tuned with examples. Therefore, the performance of the tool was judged in two ways. First by reading and manually assessing the generated summaries compared with the extracted sections from the original accident reports. Secondly, by comparing the earlier presented classificators performance on the summarized text versus the original report.

From this manual assessment, the tool generates a summary that accurately captures the main points and key information of the identified sections in a concise and readable format. In a quick and efficient manner. However, in an attempt to fit to the token limit, the tool can sometimes combine human actions in a way that misrepresents what is stated in the original accident report, missing key information of contextual significance. An example of this, was "*….The operator shut down the pump when a leak was detected in the pipeline. The technician repaired the leak and the pump was restarted….*" was summarized to "*The operator shut down the pump and the technician restarted the pump.*" Here any information regarding the leak and repair was excluded.

In order to quantitatively test the performance and information loss by the summarization tool, the earlier presented classificator tool is applied to the summarized version of ten reports available from the MATA-D. The classificators outputs are

then compared with the entries in the MATA-D, so that the performance metrics can be calculated for the summarized versions. The accuracy (83%), precision (77%) and recall (65%) decreased by 8% to 12% for the summarized reports when compared to the full report metrics. These values show that some information is lost as expected, however these compare well with the bench mark figures attained by the SVM based classificator and demonstrate that the summarization retains significant amounts of information regarding the human role in incidents.

Fine-tuning the summarization layer of the tool using human written summaries may improve the performance of the tool particularly with regards to domain specific language and key events. This would come at a significant time investment as a human expert would need to read and summarize each accident report in the training set. However, from other similar applications in literature it is suggested this would be a worthwhile investment (Yadav, Patel and Jani 2023).

4. Case Study

Within this section the classificator tool has been applied to two different document types to demonstrate different possible uses and applications of the tool, and the summarization tool also applied to the first document. The first of which is an accident report regarding an incident involving the unexpected activation of the firefighting system during maintenance work on the diesel generators, trapping two employees in the room, leading to one fatality.

Both of the presented case study documentation are originally in Portuguese, therefore to be input into the tool these are translated to English using the document translate option freely available on Google Translate.

4.1. Case Study Accident Report

With regards to the firefighting system accident, the human factors classificator tool's best performance identified the following performance shaping factors, 'Communication failure ', 'Missing information ',' Maintenance failure ', 'Design failure ', 'Insufficient skills ', 'Insufficient knowledge ',' Inadequate team support ', 'Fatigue ','Cognitive bias ', 'Equipment failure ', 'Inadequate procedure', 'Inadequate quality control ', 'Inadequate task allocation ', 'Wrong Place'. This report contained a Human Factors analysis section which was removed before being uploaded to the tool, as to not influence the tool with information the tool has been developed to produce. From the factors discussed in this section, the tool failed to identify the following four factors, 'Adverse ambient conditions', 'Access problems', 'Excessive demand', 'Inadequate workplace layout', and also identified 'Wrong Place' which was not discussed.

To quantify the tools performance, it has been applied to this report ten times, and the average and standard deviation (s.d.) of the performance metrics calculated. The average accuracy of 88% (s.d. 2.5%), average precision of 89% (s.d. 4.5%) and an average recall of 73% (s.d. 4% The performance of the tool here is in line with the metrics attained on the MATA-D reports, and demonstrates how the tool can be used on translated documents and to expand the MATA-D.

When applying the summarization tool to this accident report, the tool takes the roughly 70 page document, and is able to produce a page and a half summary. The summary covers the main points stating for example;

- Firefighting system was designed to safeguard operators, introduced additional risks due to its failure
- Safety technicians were assigned to test and visually inspect the systems, rescue team did not know the industrial plant.
- Lack of communication from the issuer to the executors in the planning meetings
- Teams failure in maintenance caused leakage from generators
- Design flaws present in room prevented works escape route
- Escape route was incorrect way, experienced members took other route.
- Operators were impeded visibility.
- Failure to plan and provide necessary resources for maintenance team

As in section 3.3.1, the summarized output was input into the classificator tool attaining performance metrics within 10-12% of the metrics obtained using the entire report.

4.2. Case Study Procedure Guide

Although the tool has not been trained on procedure guides there is language concerning processes, actions, etc., present in both. Therefore, the tool will have learnt relationships between certain words/phrases related to processes/actions and performance shaping factors. Therefore, applying the tool to each step/section of the procedure, the performance shaping factors most likely to influence the performance/risk of the step can be identified. To action this the user has to breakdown the procedure, and submit each step/section into the tool, an option to automate this process is planned for development.

It was therefore of interest to test the human factors classificator tool on a procedure. The chosen procedure guide is regarding instructions for a pigging operation, is also presented. Pigging in the oil and gas industry is a form of flow assurance where pipeline pigs are used to purge, clean, and/or inspect pipelines to keep them running smoothly. The summarization tool has not been trained to work with documents such as procedure guides, and thus testing on this document provided no insight or benefit. The human factors classificator identified the following factors: 'faulty diagnosis', 'wrong reasoning', 'priority error', 'inadequate procedures', 'access limitations', 'incomplete information', 'communication failure', 'missing information', 'maintenance failure', 'inadequate quality control', 'design failure', 'inadequate task allocation', 'insufficient skills', these could now be used when assessing the procedure for risk and the potential performance shaping factors.

5. Conclusion

This work demonstrates how Natural Language Processing approaches can be used to aid data gathering and information sharing within the context of Human Reliability Analysis. The developed tool "Virtual Raphael" based on BERT approach, shows excellent performances and overcome the original implementation of Virtual Raphael based on SVM. The development of a web-based interface for this tool is important, as this will allow users to simply upload their accident report and obtain the output of identified human factors without any understanding of the background code required.

The presented tool provides the classification of a report at a considerably faster rate than an expert would be able to, (approximately one minute per accident report), whilst maintaining an accuracy and precision that may be considered more in line with the performance (in terms of classification) of a human expert, compared to the original version of the classificator. This is one of the major advantages for this work due to the limited availability of pre-labelled accident reports.The improvement in the performance metrics for this version of this tool will allow the inclusion of the new entries into the MATA-D based on its classification output. This is an improvement on the recommendation with the original tool which was mainly to aid an expert in the assessment of the accident. This will increase the rate of expansion of the dataset which will reduce epistemic uncertainty when using the data for other tasks.

Alongside the classificator, the summarization tool will provide users with a better rounded access to information regarding the human role in the accident report. The summarised output will be, in part, able to allow users to more easily and quicky identify some of the evidence and reasoning for the factors output by the classificator. This reasoning could play an important role in the interpretability of the results by presenting support evidence to for example, justify the need of further investment in the design, maintenance and training to reduce the potential influence of performance shaping factors on overall risk. The presented methodology demonstrates useful results and justify the models implementation in practice.

NLP based models, much like HRA, are constantly evolving, with new technologies and more data becoming available. As the corpus of text data increases, NLP models will produce even better results This work demonstrates just the how these models can be put to practice in HRA, whether this be data gathering efforts or automating repetitive text driven tasks.

Acknowledgements

This work was partially supported by the EPSRC grant EP/T517938/1. All data underpinning this publication are openly available from https://datacat.liverpool.ac.uk/1018/ and the Virtual Raphael tools available on: https://github.com/cossan-working-group .

References

Abid, Abubakar, Matthew Carrigan, Lysandre Debut, and Leandro von Werra. 2021. *BART.* https://huggingface.co/docs/transformers/model_doc/bart.

BERT 2021. https://huggingface.co/docs/transformers/model_doc/bert.

Devlin, J, C Ming-Wei, K Lee, and K Toutanova. 2018. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805.*

Ghazizadeh, Eghbal, and Pengxiang Zhu. 2020. "A Systematic Literature Review of Natural Language Processing: Current State, Challenges and Risks." *Future Technologies Conference (FTC)* . 634–647. https://www.ibm.com/uk-en/topics/natural-language-processing.

Griffith, Candice, and Sankaran Mahadevan. 2011. "Inclusion of fatigue effects in human reliability analysis." *Reliability Engineering and Sytem Safety* 96 (1437-1447).

Groth, Katrina M., and Ali Mosleh. 2012. "Deriving causal Bayesian networks from human reliability analysis data: A methodology and example model." *Journal of Risk and Reliability* 226 (4): 361-379.

Hollnagel, E. 1998. *Cognitive Reliability and Error Analysis Method (CREAM).* Oxford: Elsevier Science Ltd.

Horev, Rani. 2018. *BERT Explained: State of the art language model for NLP* . https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270.

Johnson, K, C Morais, and E Patelli. 2023. "AI Tools for Human Reliability Analysis." *5th International Conference on Uncertainty Quantification in Computational Science and Engineering (UNCECOMP 2023).* Athens.

Khanna, Chetna. 2021. *WordPiece: Subword-based tokenization algorithm* . https://towardsdatascience.com/wordpiece-subword-based-tokenization-algorithm-1fbd14394ed7.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* . Association for Computational Linguistics. 7871–7880 .

Lin, Chin-Yew. 2004. "ROUGE: A Package for Automatic Evaluation of Summaries." *Workshop on Text Summarization Branches Out.* Barcelona.

Morais, C., K. Yung, and E Patelli. 2019. "Machine-learning tool for human factors evaluation-application to lion air Boeing 737-8 max accident." *UNCECOMP 2019 and 3rd ECCOMAS Thematic Conference.*

Morais, Caroline, Kai Lai Yung, Karl Johnson, Raphael Moura, Michael Beer, and Edoardo Patelli. 2022. "Identification of human errors and influencing factors: A machine learning approach." *Safety Science* 146.

Moura, Raphael, Beer Michael, Edoardo Patelli, John Lewis, and Franz Knoll. 2016. "Learning from major accidents to improve system design." *Safety Science* 84: 37-45.

Osuna, E., R. Freund, and F. Girosi. 1997. "Support Vector Machines: Training and Applications ."

Papineni, K., S. Roukos, T. Ward, and W. J. Zhu. 2002. "BLEU: a Method for Automatic Evaluation of Machine Translation." *40th Annual Meeting of the Association for Computational Linguistics (ACL).* Philadelphia. 311-318.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need ."

Wolff, Rachel. 2020. *What is natural language processing?* February 26. https://monkeylearn.com/blog/what-is-natural-language-processing/.

Yadav, Hemant, Nehal Patel, and Dishank Jani. 2023. "Fine-Tuning BART for Abstractive Reviews Summarization." In *Computational Intelligence. Lecture Notes in Electrical Engineering.*, by A Shukla, B.K. Murthy, N Hasteer and J.P. Van Belle, 968. Singapore: Springer.