# Industrial Accident Report Analysis Using Natural Language Processing

**Praveen Sankarasubramanian, Dr. EN. Ganesh**

**Abstract :** Industrial safety stays basic worry in many nations. Industrial accidents cause human suffering as well as result in immense money related misfortune and ecological effects. To counteract these accidents in the future, the examination of the risk control plan is basic. In every industry, casualty and accident reports could be accessible for past accidents. This “design research paper” proposes the Accident reports mining using NLP.

**Keywords :** KNN, NLP, SQP, SVM

## 1 INTRODUCTION

International Labor Organization (ILO) indicates that nearly 2.78 million workers frequently affected by work-related accidents [1]. Industrial accidents cause heavy loss to the employees, industry as well as the environment. To predict the accident, it is significant to investigate the past accidental reports. Based on the acquired knowledge safety experts can make the right move to evacuate or decrease the cause of an accident. Abused protective equipment, unmaintained safety articles, and catalog increase the occurrence of an accident [1]. Performing obligatory safety checks before operating a machine, bringing issues to light, frequent auditing and inspection of the machines would reduce the cause of accidents. In industries, after an accident, the disaster management expert or industrial safety expert maintains an accident/disaster investigation report. This report gives the total depiction of the accident and provides points for further investigation. This proposed idea paper; proposes Natural Language Processing (NLP) to do text mining and gather information from the accident investigation report. To categorize and find the reason for an accident a collection of Machine Learning (ML) algorithms are used.

## 2 RELATED WORK

Praveen et al. [1] have listed the list of the hazardous environment in an industrial environment. Praveen et al. [2] proposed an approach to identify the similarity between two sentences. Bertke et al. [3] used a Naïve Bayesian model to order the cause of the accident. This model provides an accuracy of 90%. To achieve a sensitivity of 67.8 % and 74 % Taylor et al. [4] combined the Fuzzy model with Naïve Bayesian. Wellman et al. [12] achieved an accuracy of 87 % using the Fuzzy Bayesian classification model external Injuries and poisoning categorized from the UN National Health Interview Survey reports. Using manual error correction technique and linear regression model accuracy increased by 1.8 %. Tixier et al. [5, 6] combined Random Forest (RF) and Stochastic Gradient Tree Boosting (SGTB) to anticipate the vitality associated with the accident, type of damage, affected body proposed from the accident investigation reports. Nearly 95% of accuracy obtained by adding NLP to the earlier proposed approach. Goh et al. [7] combined Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbor (KNN), Linear Regression (LR), Decision Tree (DT), and Naïve Bayes (NB) to group the accident reports. Chokor et al. [10] used KNN to order the accident report. Fan et al. [8] did a comparison between the content mining approach and the case-based thinking approach. They demonstrated that the content mining approach provides better accuracy. Zou et al. [9] combined NLP and Vector Space Model to group the accident report based on semantic. In the proposed approach, the recall ranged between 0.5 and 1.


- *Praveen Sankarasubramanian Research Scholar, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu, India. Email Id: praveengrb@gmail.com*
- *Dr E.N. Ganesh, Dean, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Chennai, Tamilnadu, India.Email Id: enganesh50@gmail.com*


## 3 METHODOLOGY

**This involves**

1. Data preprocessing
2. Grouping sentences based on similarity
3. Building a lexicon dictionary

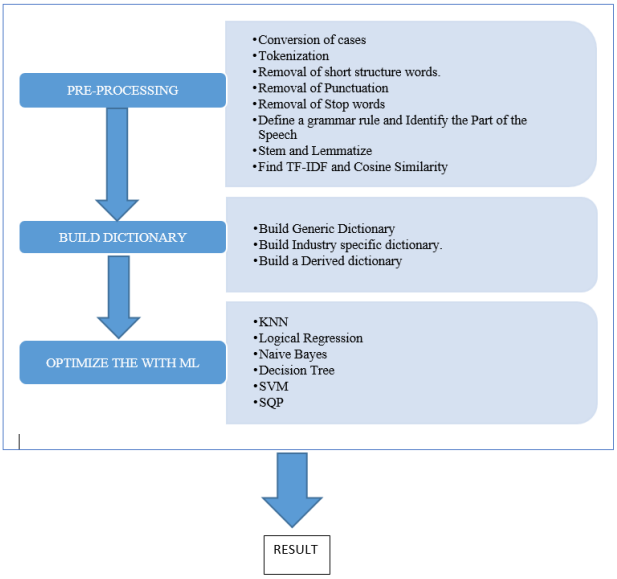



***Fig. 1.*** *Text mining process*

## 4 PRE-PROCESSING

### 4.1 Conversion of cases

Some programing language and machine learning frameworks

are case sensitive. The case of the text converted as either upper or lower. In general, lower case is preferred.

### 4.2 Removal of short structure words

1. When an address is written, the street is represented as St.
2. Months represented as Jan., Feb., and so on...
3. Measurements represented as cm, mm, km, in.

Most of the framework splits the sentences based on the dot or period symbol. Words like {i.e., pg., e.g. M.Tech} creates hindrance for the sentence splitting and tokenizing. Refer Marker No 5 in Fig.2.

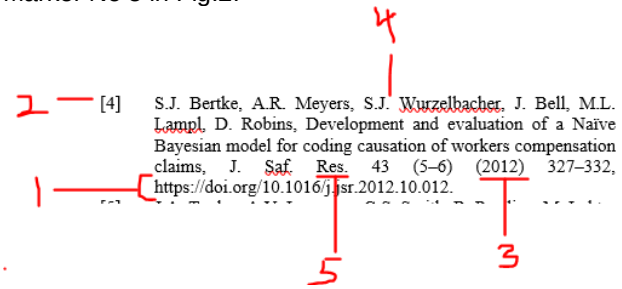



***Fig. 2.*** *Corrupted Text*

### 4.3 Tokenization

The text document parsed and chunked as smaller units like paragraphs, sentences, etc. The tokenization [17] process chops the sentences or the text stream into pieces of words called as tokens. This tokenization process followed by abbreviations or acronyms cleansing, punctuation mark cleansing, special character removal, stop word removal. Consider the example sentence: Employee rolled his ankle on moving floor plate.
tokenized as:
{‘employee’,”rolled”,”his”,”ankle”,”on”,”moving”,”floor”,”plate”}

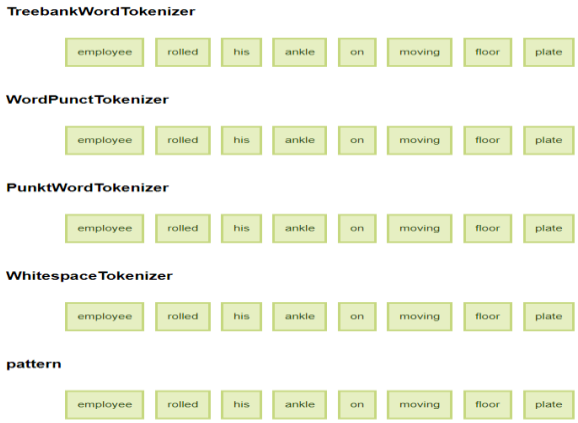



***Fig. 3.*** *Word Tokens*

### 4.4 Removal of punctuations and special characters

Punctuation marks, Special characters, domain name behaves like a noise in the text. They do not play a vital role in the information. Refer Marker No 1,2,3,4 in Fig.2. Removal of these words improves the quality of the text.

### 4.5 Removal of Stop words

Stop words [17] has extremely little value in helping a document. Removal of these words will not create a huge impact on the application.

***Fig. 4.*** *Stop words*

### 4.6 Define a grammar rule and Identify the Part of the Speech

In this process, words assigned with proper tags (noun, verb, and preposition). It helps to build the grammatical relationship between words.

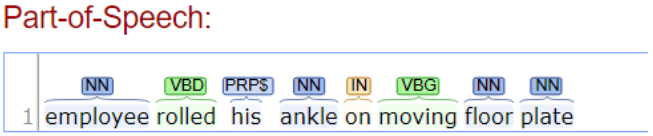



***Fig. 5.*** *Parts of Speech*

Few words are difficult to relate to the industrial context. For example, words like BREAK, CRUSH, SWING, SITE, and PLATFORM tagged with ‘NN’. N-Grams [11] of text is a set of co-occurring words within a given word window size. The word window size could change as a unigram, bigram, and trigram and so on. Consider the sentence: Employee rolled his ankle on moving floor plate

Unigram or 1-gram:
{"employee", "rolled", "his", "ankle", "on", "moving", "floor", "plate"}
Bi gram or 2-gram:
{“employee-rolled","rolled-his","his-ankle","ankle-on","on-moving","moving-floor","floor-plate”}

Tri gram or 3-gram:

{“employee-rolled-his","rolled-his-ankle","his-ankle-on","ankle-on-moving","on-moving-floor","moving-floor-plate”}

4gram:

{“employee-rolled-his-ankle","rolled-his-ankle-on","his-ankle-on-moving","ankle-on-moving-floor","on-moving-floor-plate”}

### 4.7 Stem and Lemmatize

Inflected words are words derived from other words. Inflected Language uses Inflected words in speech-language [17]. The amount of deviation of the derived word from the root word is the degree of inflection. The degree of inflection may be lower to higher. In Natural Language Processing, Stemming and Lemmatization are part of Text Normalization or Word normalization. A word in a selected document is present in different ways.

Consider the example sentence: Employee rolled his ankle on moving floor plate.
Is stemmed and lemmatized as

Employee → employe
Rolled → roll
Ankle → ankl
Moving → move

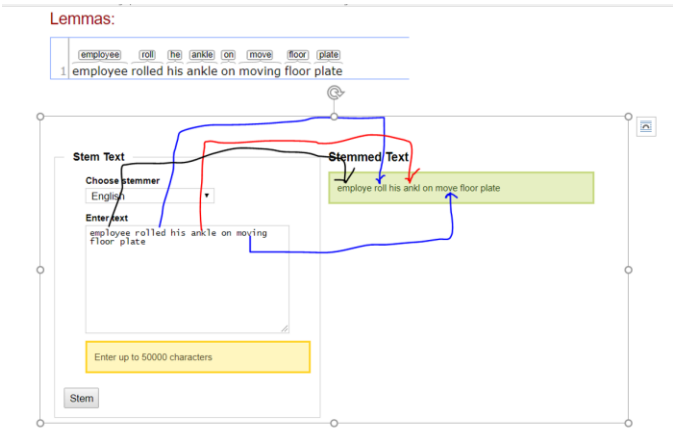



***Fig. 6***. *Lemma and Stems*

am, are, is ⇒ be
car, cars, car's, cars' ⇒ car

The result of this mapping of text will be something like:

the boy's cars are different colors ⇒
the boy car be differ color

(F)

| Rule | | | Example | | |
|---|---|---|---|---|---|
| SSES | → | SS | caresses | → | caress |
| IES | → | I | ponies | → | poni |
| SS | → | SS | caress | → | caress |
| S | → | | cats | → | cat |

***Fig. 7***. *Rules of lemmatization and stemming*

### 4.8 Find TF-IDF and Cosine Similarity

Term Frequency and Inverse Document Frequency (TF - IDF). TF-IDF [17] is a factual result of Term Frequency and Inverse Document Frequency. Term Frequency is a raw count of a word in a Document and Inverse Document Frequency is a proportion of how much data completes a word give in a document (obtained by isolating the all outnumber of archives by the number of reports containing the term, and after that taking the logarithm of that remainder). TF-IDF is equivalent to {ratio of the event of a term in a document by the number of terms in a document} * log {ratio of the complete number of documents by the quantity of document that contains the term}

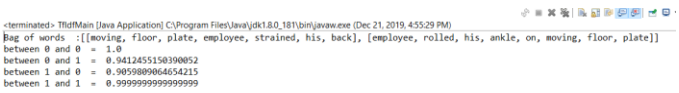



***Fig. 8***. *TF-IDF and Cosine Similarity*

### 4.9 Grouping Sentences Based on Similarity

A pair of selected sentences could be identical or non-identical [2]. Nearly 19 categories of sentence relationships.

The structure of the sentence identified by finding -

1. First, the length of the sentences is calculated,
2. Next, the longest common substring and cosine similarity is calculated
3. Finally, the similarity between the sentences based on a context is identified and grouped

Pre-processing steps improve the quality of the sentence and reduce the payload sent to the Machine Learning Algorithms.

## 5 BUILD LEXICAL DICTIONARY

Every industry will have its own set of terms or keywords. For example, a company can have a unit name called EYE and this is different from the general term eye. Same way, Jaguar is a car name as well as an animal name.

Consider sentences:

1. Employee Alex slipped from SDB 4 staircase.
2. While cleaning chimney Albert exposed to toxic fumes.

To extract the cause of the accident and find its remedy, simple cleansing and preprocessing will not be enough. The text related to the context of the industry. Three steps are required to relate the sentences with the context.

### 5.1 Build a Generic Dictionary

Globally available lexical databases (like Wordnet and Brown) used to build a generic dictionary. [5][6]

### 5.2 Build an Industry-specific dictionary.

Every industry will have some specific keywords. It is necessary to collect, build and train those keywords. [5][6]

1. In factories, words like chamber, boiler, unit, stage, and conveyor are common.
2. In the IT sector words like offshore development center (ODC), software development block (SDB) are common.

### 5.3 Build a Derived dictionary

Second could be building a generic derived keyword based on the Industrial keyword. Example: building concrete, pouring acid, adding caustic soda. In addition to that, complex patterns like “eye irritation due to sodium fire”. Pre-processing steps and building a lexical dictionary improves the quality of content and reduces the error rate. Cleansed text passed through the optimizer.

## 6 OPTIMIZE THE RESULT

Optimizer combines multiple algorithms to achieve better results. Optimization processes are – classification, error-handling using linear regression and manual process. The optimizer 6 modules.

***TABLE I.*** *OPTIMIZER MODULES*

| MODULE | DESCRIPTION |
|---|---|
| Probabilistic Classifier | Classification of text and document did in this module [18]. Example: Naive Bayes classifier |
| Regression | This module identifies Connection between a selected word or variable and other illustrative variables. It foresees the likelihood of given sample information [19]. Example: Logistic regression algorithm |
| Feature Similarity Classifier | This module identifies the similarity between the sentences [16]. Example: KNN: K-Nearest Neighbor |
| Rule applier | Rule applier module contains a series of rulesets. Every ruleset is cascaded. The first ruleset represents the root and the last represents the leaf. Ruleset from root to |

| | |
|---|---|
| | leaf applied in every sentence. This helps to identify the category of text [20]. Example: decision tree |
| Structural Risk Minimizer Module | It minimizes error and improves the confidence level. It solves the optimization problem. Example: Support Vector Machine [15] |
| Non-Linear Optimizer | This module uses Lagrangian multipliers, Newton's technique and Sequential Quadratic Programing (SQP) [14].<br>Sequential quadratic programing (SQP) takes care of non-linear optimization problems. This algorithm deals with any level of non-linearity. SQP joins two basic calculation for taking care of non-linearity optimization problem.<br>1. Functioning set technique<br>2. Newton's technique |

## 7 EXPERIMENTATION

To achieve higher accuracy supervised learning approach applied on the preprocessing steps.

1. Tokenization performed to break the longer sentence to shorter sentence.
2. Stopwords provide poor accuracy and consumes more processing time. Hence it is removed
3. POS tags mapped with the TOKENS.
4. Lemmatization performed to remove Inflected words.
5. Internally N-Gram is calculated.
6. Because of pre-processing, the documents represented as matrix. TF-IDF matrix is generated (Sample result section pre-processing)
7. To classify and evaluate, split the dataset as 80-20. 80 % for training and 20 % for testing.
8. Calculate the accuracy and fine tune the parameters to improve the accuracy

To make the understanding easier, this experimentation explained with single sentence.

Consider the sentence:
"Employee is splashed with hot water and is burned"
Figure 9 – 13 shows the how the relationship between the words are present in normal approach and the enhanced approach. It shows that removal of stop words improves the constituency parser and sentiment analysis. However, there is a greater impact on extracting the relationship between the words. When stop words are present, Entities mixes with stop words. {In this case, Employee, Splashed, Hot water, Burnt are the entities}. Stopwords and other impurity changes the mood of the sentence. In normal approach, the sentence behaves like a POSITIVE mood. In enhanced mode, it improved to identify as a NEUTRAL mode

### *A.* Normal approach

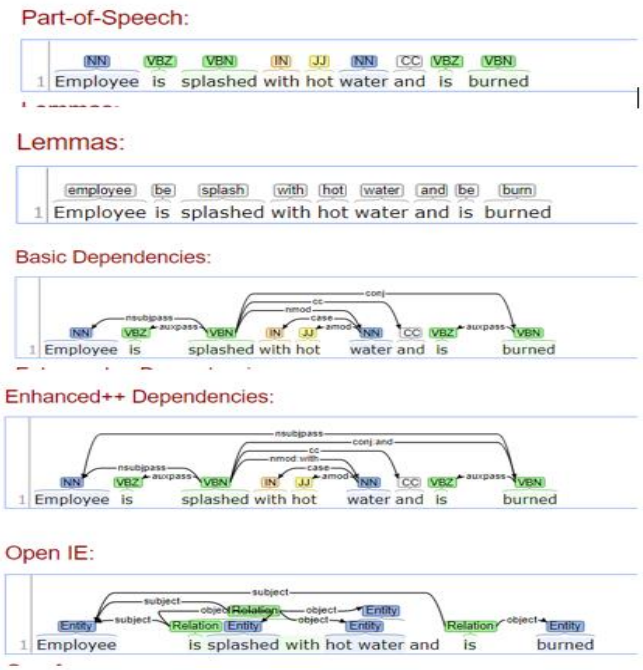
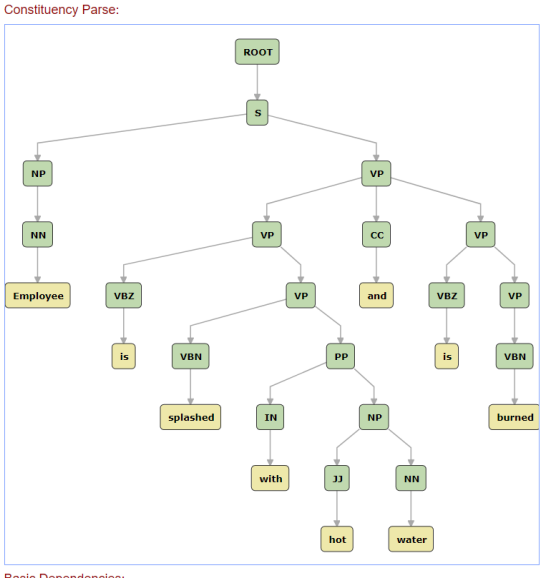



***Fig. 8***. *Result and weightage*

**Named Entity Recognition**
This text do not contain any global named entities [13].
Constituency Parse




***F***
***ig. 9***. *Consistency Parser*

**Sentiment Analysis**

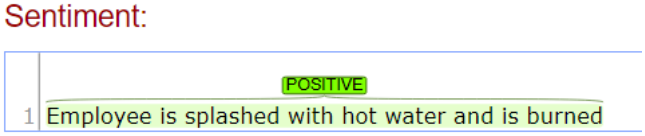



***Fig. 10***. *Sentiment analysis of the sentence*

### *B.* Using proposed framework

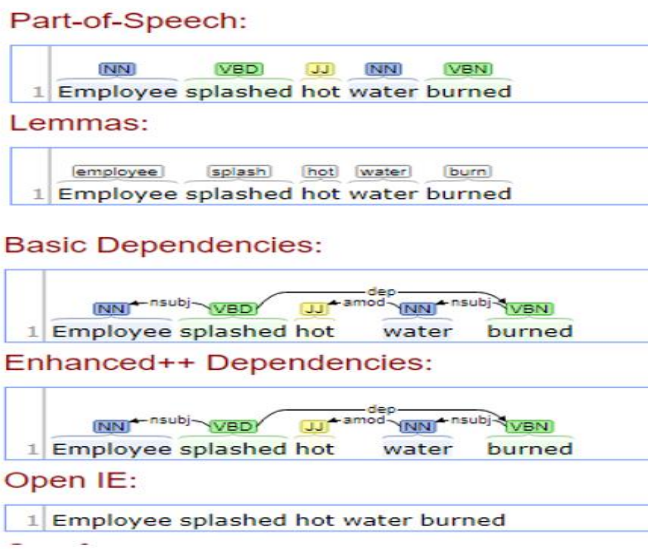



***Fig. 11**. Result and weightage*

**Constituency Parse**

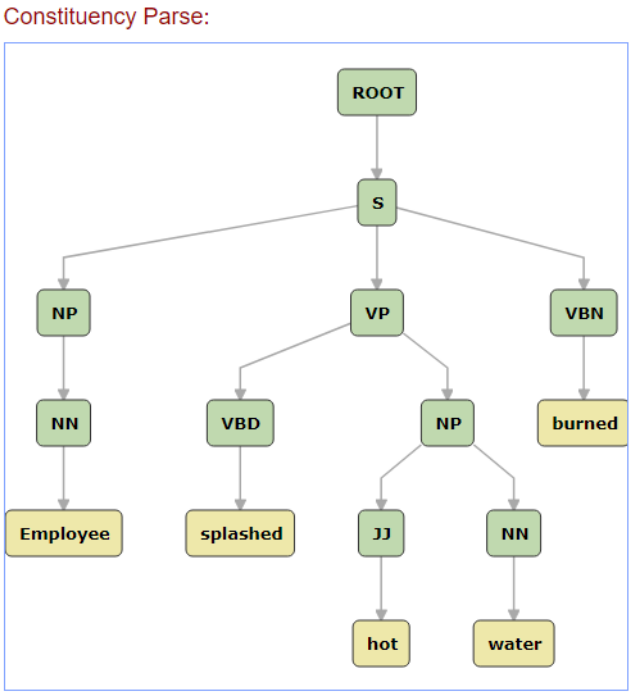



***Fig. 12**. Constituency Parser*

**Sentiment Analysis**

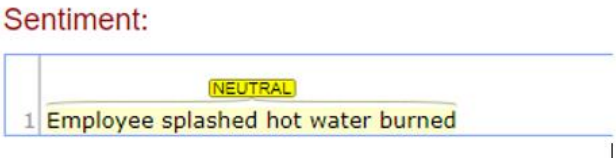



***Fig. 13.** Sentiment Analysis*

## 8 CHALLENGES AND LIMITATION

The current application fails to identify the sentiment of the text. Consider the sentence specified in the earlier section, it denotes that an employee met with a minor accident. The expected result should be NEGATIVE. However, the result was POSITIVE or NEUTRAL. Thus, algorithm should be fine-tuned to adapt the sentiment analysis. This approach requires a huge dictionary of industry-specific keywords and derived keywords. The dictionary should be auto-indexing and should be self-tuning. The performance of the application is the next challenge. Proper infrastructure, algorithms should be used to process the data in seconds.

## 9 CONCLUSION AND FUTURE OF WORK

Analyzing the historical accident reports identifies what went wrong in the past and this valuable knowledge prevents future accidents. This is an essential prevention strategy. This can mitigate the potential risks of failure or accidents. However, manual classification of accident investigation reports is time dragging, labor-intensive and error prone. The proposed model is self-learning and robust. In the end, this provides a safer analytical approach. This proposed model outperforms a single model in terms of quality of recall and accuracy. Future work will concentrate on improvisation of data quality, Performance metrics of the optimizer module needs to be calculated. This idea needs to integrate with the other two types of research [1, 2] and its usage in other sectors. For example, this approach designed for analyzing the accident investigation report. The future of work is to identify the usage of this model in an alternative sector like health (hospitals), travel, education, and others.

## REFERENCES

[1] Praveen Sankarasubramanian and Ganesh. E.N, "IoT Based Prediction for Industrial Ecosystem," International Journal of Engineering and Advanced Technology, vol. 8, no. 5, pp. 1544-1548, June 2019.

[2] Praveen Sankarasubramanian and Ganesh. E.N, "Algorithm to Identify the Connection between Sentences," International Journal Of Information And Computing Science, vol. 6, no. 7, pp. 158-162, July 2019.

[3] Bertke, Steve & Meyers, Alysha & Wurzelbacher, Steven & Bell, Jennifer & Lampl, Michael & Robins, David., "Development and evaluation of a Naive Bayesian model for coding causation of workers' compensation claims," Journal of safety research, vol. 43, pp. 327-32, 2012.

[4] Taylor, Jennifer & Lacovara, Alicia & Smith, Gordon & Pandian, Ravi & Lehto, Mark, "Near-miss narratives from the fire service: A Bayesian analysis," Accident analysis and prevention, vol. 62C, pp. 119-129, 2013

[5] Tixier, Antoine & Hallowell, Matthew & Rajagopalan, Balaji & Bowman, Dean, "Application of machine learning to construction injury prediction," Automation in Construction, vol. 69, pp. 102-114. 2016.

[6] Tixier, Antoine & Hallowell, Matthew & Rajagopalan, Balaji & Bowman, Dean, "Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports," Automation in Construction, vol. 62, pp. 45-56, 2016.

[7] Y.M. Goh, C.U. Ubeynarayana, "Construction accident narrative classification: an evaluation of text mining techniques," Accident Analysis & Prevention, vol.108, pp. 122–130, 2017.

[8] Fan, Hongqin & Li, Heng. "Retrieving similar cases for alternative dispute resolution in construction accidents using text mining techniques," Automation in Construction, vol. 34,

pp. 85–91, 2013.
[9] Zou, Yang & Kiviniemi, Arto & Jones, Stephen, “Retrieving similar cases for construction project risk management using Natural Language Processing techniques,” Automation in Construction, vol. 80, pp. 66-77, 2017.
[10] A. Chokor, H. Naganathan, W.K. Chong, M. El, “Analyzing Arizona OSHA injury reports using unsupervised machine learning,” Procedia Engineering, vol. 145, pp. 1588–1593, 2016.
[11] C.D. Manning, Foundations of Statistical Natural Language Processing, in: H. Schütze (Ed.) MIT press, 1999.
[12] H.M. Wellman, M.R. Lehto, G.S. Sorock, G.S. Smith, “Computerized coding of injury narrative data from the National Health Interview Survey,” Accident Analysis & Prevention. vol.36, no.2, pp. 165-171, 2004.
[13] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” ACL Anthology. [Online]. Available: http://www.aclweb.org/anthology/P/P14/P14-5010.
[14] J. L. Morales, J. Nocedal and Y. Wu, "A sequential quadratic programming algorithm with an additional equality constrained phase," in IMA Journal of Numerical Analysis, vol. 32, no. 2, pp. 553-579, April 2012.
[15] V. Kecman, Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. Cambridge, MA: MIT Press, 2001.
[16] S. Ren and A. Fan, "K-means clustering algorithm based on coefficient of variation," 2011 4th International Congress on Image and Signal Processing, Shanghai, 2011, pp. 2076-2079.
[17] R. Lourdusamy and S. Abraham, “A Survey on Text Pre-processing Techniques and Tools,” International Journal of Computer Sciences and Engineering, vol. 06, no. 03, pp. 148–157, 2018.
[18] P. Liu, H. Yu, T. Xu, and C. Lan, “Research on archives text classification based on Naive bayes,” 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017.
[19] P. Liu, H. Yu, T. Xu, and C. Lan, “Research on archives text classification based on Naive bayes,” 2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2017.
[20] A. Abdelhalim, I. Traore, and Y. Nakkabi, “Creating Decision Trees from Rules using RBDT-1,” Computational Intelligence, vol. 32, no. 2, pp. 216–239, 2014.