DATA-LED LEARNING: USING NATURAL LANGUAGE PROCESSING (NLP) AND MACHINE LEARNING TO LEARN FROM CONSTRUCTION SITE SAFETY FAILURES

Henrietta Baker[1], Simon Smith[2], Gordon Masterton[3] and Bill Hewlett[4]

[1, 2&3] *Institute of Infrastructure and Environment, University of Edinburgh, Sanderson Building, Robert Stevenson Road, The King's Buildings, Edinburgh, EH9 3FB, UK*

[4] *Bill Hewlett Associates Ltd, Sevenoaks, England, TN13 3RY, UK*

Failures happen. Innumerable sources stress the importance of learning from these mistakes. However, within the construction industry, there is heavy reliance on learning from case-studies of catastrophic events and a lack of attention to the more frequent, lower consequence and yet repetitive failures. These smaller failures can have huge cumulative impact, not to mention their effects on the individual(s) involved. The Health and Safety Executive in their 2018 Annual Report estimated that safety injuries on site cost £490M to the UK economy. Part of this historic inattention is due to difficulties in analysis and sense-making of these failures. While information is collected about the failure event, the data tends to be in the form of free text, which is notoriously difficult to analyse. To begin addressing this, we present an attribute-based method which uses Natural Language Processing (NLP) and Machine Learning to structure text data collected after a safety failure on-site, including near misses and incidents. This structured data allows systematic analysis of these data to improve construction site practices and facilitate data driven decision-making that will reduce safety incidents. Using descriptions from 2345 safety reports, provided by a UK based construction company, we manually refine a set of attribute-based event descriptors from the text descriptions of the incidents and train an NLP model to automatically predict these in new descriptions. As well as presenting a working example of this method, factors affecting the prediction accuracy were also explored. This critique found four aspects which need deliberate consideration in application of NLP to construction safety text. These are (1) the number of attributes; (2) data class imbalance; (3) inclusion of near-miss data as well as incident reports; and (4) algorithm selection and optimisation. This method also anonymises the reports, allowing potential industry-wide data sharing and learning.

Keywords: AI, Machine Learning, Natural Language Processing (NLP), safety

INTRODUCTION

To improve its safety performance, the industry must learn from mistakes. At present, this experiential industrial learning is mostly limited to case-studies and alerts (Baker *et al.,* 2018). However, there is a wealth of information contained within accounts of

[1] s1679725@sms.ed.ac.uk

Baker, H, Smith, S, Masterton, G and Hewlett, B (2020) Data-Led Learning: Using Natural Language Processing (NLP) and Machine Learning to Learn from Construction Site Safety Failures *In:* Scott, L and Neilson, C J (Eds) *Proceedings of the 36th Annual ARCOM Conference,* 7-8 September 2020, UK, Association of Researchers in Construction Management, 356-365

more frequent, lower consequence incidents and safety observation reports of near misses or unsafe conditions. Historically, these data have been largely inaccessible for analysis due to issues of physical accessibility and the format of unstructured text data, requiring time consuming manual intervention.

In the last decade, these safety reports have been increasingly captured digitally, removing a barrier to physical accessibility. Meanwhile, the rise of digitally available data worldwide, including natural language and text data, has sparked an emergence of many new analysis methods. Natural Language Processing (NLP), also known as computational linguistics, is now being applied to text data in many industries for tasks such as information retrieval, knowledge discovery and analysis.

Previous research, such as that of Tixier *et al.,* (2016), Zhang *et al.,* (2019a), Zhong *et al.,* (2020), has explored application of NLP to construction safety documents in an American context. However, application in the UK has been limited to retrieval cases and keyword expansion, such as Zou *et al. (*2017). The research laid out here explores how NLP can extract valuable knowledge and facilitate learning from unstructured data. It also explores the challenges to doing so and uncovers some of the barriers and difficulties that need to be overcome. This is part of a larger investigation exploring learning from failure on construction sites that has previously been reported at ARCOM (see Baker *et al.,* 2018, and Velikova *et al.,* 2018).

BACKGROUND

Safety in Construction

Safety research in construction is a rich field. Safety measures, like Accident Frequency Rate (AFR), in developed nations have dropped significantly but are now, by and large, plateaued. Meanwhile, safety research has moved from retrospective (lagging) measures to include more proactive (leading) measures and safety culture. To this end, there has been recent interest in data-led safety initiatives like predictive analytics and risk libraries.

Previous research has shown the advantage of using safety event attributes to predict safety event outcomes, where attributes are descriptors of the situation before any safety event has occurred (Esmaeili *et al.,* 2015). However, methods of extracting these attributes are specific and time consuming. Tixier *et al. (*2016) developed a rule-based AI method to predict 30 precursor attributes. While this achieved extremely high accuracy values, the method was time consuming and may be specific to American English and American construction terminology.

However, in application of an attribute-based method, there are several key points of information which need addressing: what are the 'attributes' we need to identify? How can 30 attributes truly describe the complexities of a construction site area? And, do these methods still work in a UK context?

Natural Language Processing (NLP) in Construction

Natural Language Processing (NLP) is a rapidly developing multi-disciplinary field, using concepts from linguistics and data science including machine learning. As well as applications in audio recognition and machine translation, NLP is used for tasks such as text retrieval, sentiment analysis and semantic analysis.

Converting the unstructured free-text into a structured representation is the first step in most NLP analyses of free-text data. In the last 20 years, empirical 'Bag of Words' (BoW) representations (also known as vector space models) have dominated the

research space due to their notable results when trained on large datasets (Hirschberg and Manning, 2015). These representations are based on the numerical frequency of unique 'tokens' contained within the training vocabulary. 'Tokens' are generally words but also may also include punctuation or numbers. The resultant representation is a very long, sparse vector. An example of vector space transformation for a limited 7 token vocabulary is shown in Table 1.

Table 1: Example of text transformation to BoW vectors

	the	excavator	parked	in	walkway	was	blocked	path	digger
The excavator parked in the walkway	2	1	1	1	1	0	0	0	0
The path was blocked	1	0	0	0	0	1	1	1	0
The digger blocked the walkway	2	0	0	0	1	0	1	0	1
Total	**5**	**1**	**1**	**1**	**2**	**1**	**2**	**1**	**1**

The main limitations of vector space representations are that (a) word order is not preserved, and (b) semantic similarity is ignored as semantically similar words (e.g. ‘excavator’ and ‘digger’) occupy orthogonal dimensions in the vector space. Both limitations restrict the semantic meaning which can be gained from analysis of the text; however, these can be mitigated in the pre-processing stage. Specific mitigation methods are discussed in the method section.

Baker *et al. (*2020) found 12 examples of Natural Language Processing applied to construction safety incident reports. Nine of these used the vector space representations and three experimented with word embeddings, a method of representing text using deep learning. To date, no significant advantage to accuracy has been achieved using deep learning methods, despite the increase in complexity. This research employs vector space representations.

Once a structured representation of the text has been achieved, further analysis and machine learning tasks can be performed, such as text retrieval or classification. Classification tasks using machine learning classifiers can be performed for both binary and multi-class classification. Meanwhile, vector similarity can be mathematically evaluated to rate the similarity of a vector against another in the corpora allowing similar documents to be retrieved.

In recent years, a number of academic studies have showcased NLP use in construction for both text retrieval and classification tasks. Chokor *et al. (*2016), Goh and Ubeynarayana (2017), Tixier *et al. (*2016) and Chang *et al. (*2019) all use NLP methods to classify safety documents, while Kim and Chi (2019), Yu and Hsu (2013) and Zou el al. (2017) retrieve accident report cases. The most recent relevant example is Zhong *et al. (*2020) who used safety text descriptions to predict the incident causal category, e.g., 'falls', and applied topic mapping to find key labels associated with different categories. Key identifying words and phrases were extracted for each class.

Outside the domain of health and safety, there are fewer examples. However, Soibelman *et al. (*2008) used NLP to classify construction management documents and Marzouk and Enaba (2019) did the same for contractual documents.

METHOD

This research is formed of two parts: development of the attributes through systematic labelling of the safety report dataset, followed by application of NLP and ML to predict attributes in new safety event descriptions. This approach is adapted from

protocols developed and observed at the University of Colorado, Boulder (for example (Tixier *et al.,* 2016b).

In this section, following introduction of the data set, the method for development of the attributes and data labelling is described. Then, the method for development of automatic prediction of the attributes from new text is introduced.

Data

The data used in this analysis consists of 2345 safety reports, near misses and safety observations from 28 infrastructure construction projects in 10 sectors. 879 of these were accident reports. These projects took place between 2011 and 2019.

Safety Event Attributes and Data Labelling Method

Precursor attributes were defined as attributes which are identifiable before an incident occurs and contribute to the incident occurring. These were separated into three categories: objects - materials, tools and machinery; actions - actions being undertaken which contributed to the incident; and worksite descriptors - defining features of the workspace or area of incident.

Precursor attributes were manually labelled by four researchers at the University of Edinburgh, within the School of Engineering, using Microsoft Forms to collate the data. Bearing this in mind, it is vital to recognise the impact of these researchers on the development of the attribute dataset. By using personnel familiar with construction and engineering, rather than linguists, they should have been able to more accurately identify the pertinent information in the text. However, previous experience of construction could also have resulted in unconscious bias where individuals have preconceived notions about what is important on construction sites. Table 2 demonstrates the labelling process from unstructured text to precursor attributes.

Table 2: Examples of safety event descriptions and labelled attributes

IP slipped down temporary steps. The steps were wooden and wet which made them slippery.	Objects: Stairs Actions: Moving around Worksite descriptors: Slippery surface
IP was cutting the old safety barrier with a cut off saw. He went to step over the barrier to make a new cut when the saw, which was still running, slipped causing an abrasion to his left thigh.	Objects: Barrier, Powered Saw, Sharp Edge Actions: Cutting Worksite descriptors: None

Attribute Prediction Method

This sub-section describes the method used to process the text data and the classification algorithms used to predict the attributes.

The data was split into three sets. A validation set of 10% was set aside to test for final accuracy results. Train and test datasets were made using K-fold split with K=5 on the remaining data. This means that the train/test data was split 80:20 and the algorithms trained five times, using a different 20% of the data to test each time. The accuracy scores are then averaged. Model parameters were optimised on this train/test data, before final accuracy testing using the validation set.

First, the raw text is pre-processed then transformed into a TF-IDF (Term Frequency - Inverse Document Frequency) vector space representation. To elaborate:

1. Tokens were created by splitting text on whitespace and punctuation.
2. To decrease vocabulary length and integrate some semantic relationships into the model, the lexical stem of each token was extracted using the Snowball algorithm (Porter, 2001). For example, 'management' and 'managing' would both map to 'manag'.
3. To mitigate against word order loss, bigrams (pairs of words) which occur more than 5 times in the training set were found and included as tokens. For example, 'circular saw'. Less frequent and larger phrases were not included as this would increase the vector length and sparsity to such an extent that it becomes difficult to fit any model.
4. At this stage, stopwords (words which are deemed not to add sematic meaning to text), punctuation and numbers were removed.
5. TF-IDF transformation scales the term frequency (i.e. number of token counts) by log (inverse document frequency) i.e., log(number of documents in total / number of documents containing the word). These logarithmically scaled word counts identify defining words for the document (Jones, 2004). This transformation can be considered standard for initial investigations using NLP vector space representations.
   For example, transforming the first entry in Table 1, 'the' would have a count of 2 x log (3/3) = 0 while 'excavator' would have a count of 1 x log(3/1) = 0.477.

These parameters, e.g., vocabulary and TF-IDF transformation coefficients, are calculated on the training dataset then applied to the test and validation set.

For prediction of attributes from the safety event description text, each attribute was considered independently. Binary classification algorithms were trained using the TF-IDF text representation with their associated classification for that attribute. Table 3 gives brief explanations of the algorithms investigated.

Table 3: Machine Learning classification algorithms

Algorithm	Short Description
Naïve Bayes	This purely statistical classification algorithm relies on application of Bayes Theorem to calculate the probability of "***the class***" given the tokens vs the probability of not "***the class***" given the tokens.
kNN (Next Nearest Neighbour)	This classification method uses vector distance to identify the closet example in the training set, then adopts this example's classification. If k>1, an average is used. Here, k=5.
Decision Tree	Decision trees split the data using binary 'queries', repeating this until each end point only contains one data class. The 'split' or branches are optimised by maximising the entropy gain.
Gradient Tree Boosting	This ensemble method regularises the base algorithm by training several shallow examples and averaging the results. Regularising algorithms reduces their potential for overfit.
SVM (Support vector machines)	SVM relies on graphical divisions to separate classes of data. In 2-D, this could be represented as a line best dividing the classes.
SVM Bagging	Also known as bootstrap aggregating, bagging is an ensemble method where the base model is trained several times on different bootstrap samples of the training data, and the results averaged. Bootstrap sampling involves sampling without replacement.
Hard Voting - Gradient Boosting & SVM Bagging	Hard voting is a stacking ensemble whereby two or more classification algorithms are trained, and the classification result is 'voted' on. In this case, both algorithms would need to predict an attribute for the attribute to be predicted.

An initial set of algorithm parameters were tuned using the test dataset. The final accuracy of each algorithm for the validation set was recorded. By classifying each as 'attribute present' or 'not present', this binary classification was the prediction of the individual attribute. This process was repeated for each attribute, creating a list of attribute predictions for each text in the test set.

Three values for accuracy were calculated during this analysis: recall, precision and F1 a harmonic average. These values are calculated as follows:

$$Recall = \frac{TP}{TP + FN} \qquad Precision = \frac{TP}{TP + FP} \qquad F1 = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

$$where\ TP = True\ positives, FP = False\ positives\ and\ FN = False\ negatives$$

Equation 1: Recall, Precision and F1 calculations

Imbalanced classes, when one class dominates the dataset, can confuse machine learning algorithms (Haixiang *et al.,* 2017). A method employed for dealing with this is data sampling for the training dataset where the imbalance is addressed by artificially changing the ratio of the classes. Examples of methods employed include deliberately oversampling the positive counts or under sampling the negative ones. Oversampling of the positive counts is employed in this investigation.

RESULTS

Attribute Development

In development of the attributes set, 553 precursor attributes were identified in the labelling exercise. Many of these attributes were similar and, therefore, the next iteration combined attributes which had the same, or extremely similar, semantic meaning. Examples include 'animal', 'rat' and 'mouse' attributes, identified during the labelling exercise, all map to a single 'animal' attribute for analysis.

In total, 250 unique precursor attributes were identified. This is a much higher number than the 30 previously identified by Tixier *et al.,* (2016). Additionally, only 58 attributes (listed in Table 4) occurred in 1% or more of the safety descriptions.

This high proportion of infrequent attributes is indicative of the complexity of a construction site environment, which often sees specialist tools, materials and activities. Although most construction personnel could probably name the frequent activities and their main components, terminology differs across the country, increasing the complexity of the labelling task. These factors can also affect the accuracy of text classification, as discussed in the next sub-section.

Despite the complexity, there is still great potential in this method. Figure 1 is included to illustrate possible further analysis unlocked with this attribute method. The figure shows a network of co-occurring attributes. Only those co-occurring more than 30 times are included for graph clarity. Transforming the unstructured text descriptions into structured data in the form of attribute features allows network analysis methods to be performed. Other possible further analyses unlocked by this method include risk analysis, graphical analytics, learning, and finer trend analysis.

Attribute Prediction using NLP and Machine Learning

For attribute prediction, only attributes which were observed in 1% or more of the training dataset were considered beyond an exploratory run of the classification algorithms. This is partly due to the inability of the models used to deal with the

extremely imbalanced data classes, and partly because there is a high chance that they are completely absent from the test data.

Table 4: List of precursor attributes

Attribute Type	Attributes			Number
Actions	Cutting	Driving	Using a tool	9
	Cleaning	Exiting/entering	Lifting (by machinery)	
	Lifting/pulling/ manipulating (manual)	Striking/stripping (i.e. formwork/shuttering)	Walking/moving around	
Objects	Alarm	Cabin	Cable	36
	Concrete	Crane	Door	
	Electrical source	Formwork	Gate	
	Guardrail	Hand size pieces	Hazardous substance	
	Heavy vehicle	High fence	Light vehicle	
	Lumber/Timber	Machinery	Manlift	
	Mobile Phone	Mud	Object on the floor	
	Piping	Pressure systems	Rebar	
	Scaffold	Sharp edge	Small machinery	
	Airborne particles	Stairs	Steel Sections	
	Storage tank	Unpowered hand tool	Vegetation	
	Vehicle	Water source	Object at height	
Site Descriptors	Adverse weather (storm, rain)	Congested/confined work space	Excavation	13
	Exclusion zone	Insufficient edge/fall protection	Slippery surface	
	Poor Housekeeping	Railway	Vehicle Movement Zone	
	Uneven surface	Unstable support / surface	Wind	
	Working at height			

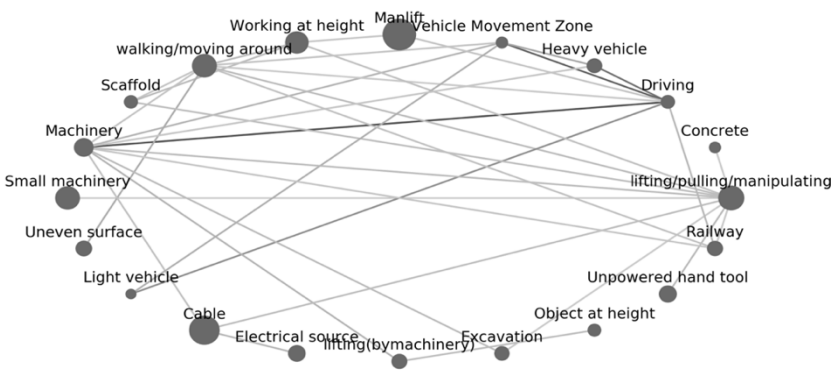

Figure 1: Network Graph of Attribute Co-occurrence (attributes with >30 co-occurrences)

To mitigate against the class imbalance, deliberate oversampling of positive examples in the training set was used for training the algorithms. In each case, the positive examples were duplicated until they accounted for a minimum of 10% of the training data. This resulted in an average increase of 0.11 to the F1 score, however, 20 / 58 attributes were adversely affected by this process. The investigation would have benefitted from a more sophisticated sampling method.

The accuracy results for the 58 attributes which occur in 1% or more of the training data is shown in Table 5, after oversampling. The best values for each are underlined and in bold. For actions and site descriptors, SVM Bagging proved to be the best option, yielding the highest F1 score. For objects, this was marginally beaten by Gradient Boosting.

In both cases, where a base and ensemble algorithm were investigated i.e., Decision Tree and Gradient Boosting and SVM and SVM Bagging, the stacked model equalled or exceeded the F1 score for the root algorithms. Also, while resulting in a lower F1, using a hard-voting algorithm increased the precision of the prediction. This is to be expected; however, it has implications for choosing methods for practice where precise results are more important than recalling all possible attributes.

Table 5: Precision, Recall and F1 scores for attribute prediction

Classification Algorithm	ACT			OBJ			SIT		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
NB	0.29	0.05	0.08	0.40	0.12	0.16	0.35	0.12	0.18
kNN	0.27	0.28	0.25	0.26	0.45	0.30	0.22	**0.41**	0.26
DT	0.41	**0.39**	0.38	0.48	0.50	0.48	0.34	0.39	0.35
GBoost	0.52	0.32	0.38	0.59	**0.54**	**0.54**	0.50	0.38	**0.41**
SVM	0.54	0.38	0.43	0.59	0.51	0.52	0.45	0.33	0.37
SVM Bagging	0.57	**0.39**	**0.45**	0.61	0.50	0.53	0.51	0.37	**0.41**
Voting (GBoost & SVM Bagging)	**0.60**	0.25	0.34	**0.68**	0.43	0.50	**0.58**	0.27	0.35

What affects attribute prediction accuracy?
Set against other recent research, these accuracy values may seem comparatively low. For example, Zhong *et al. (*2020) achieved an average F1 = 0.59 using SVM classification on word embedded text representations, achieved via word2vec with skip-gram. Meanwhile, Baker *et al. (*2020) achieved F1 = 0.72 using TF-IDF representation and SVM on a set of 6 incident types. These authors also achieved marginally higher F1 average using deep learning classifiers. There are four identified reasons these prediction tasks, on similar data, may have a higher F1 accuracy.

Firstly, these predictions had significantly fewer categories. For example, Zhong *et al. (*2020) predicted only 11 categories. Also, these were not attributes but incident categories, e.g., 'electrocution', 'falls'. Fewer categories mean that outlier accuracies can more significantly affect the average. In this case, 'electrocution' predicted with F1=0.92 brings the average from 0.46 to 0.59.

Secondly, having fewer categories may indicate a lower-class imbalance. This is also indicated in the category types. 'Incident category' or 'type' tends to be a multiple-choice option on safety incident report forms and is compulsory in most cases. This means that not only are there fewer categories, but every incident must contain at least one of them. Additionally, Baker *et al. (*2020) employed a tailored oversampling factor for each class. Oversampling was shown to have a significant effect on the accuracy results and this application could benefit from further optimisation.

Furthermore, the data used in both previous papers mentioned contained only incident reports, not near-miss or observation data. These reports tend to be more carefully filled out, using more formal English. It can also be postulated that it is easier for both those capturing the data and researchers labelling the datasets to identify precursor attributes in the case of an incident as there is less subjectivity in identifying key situational descriptors before a specific incident than in the case of unsafety.

Finally, the granularity of optimisation in these papers was much finer than used during this investigation. As stated by Bottou *et al. (*2018), "optimization is one of the foundations of machine learning"; this includes not just optimisation during the training process but optimisation of the learning parameters. Optimising algorithm

parameters has a significant effect on the accuracy results. In this investigation, SVM parameters were tuned, increasing the accuracy of prediction on the test set up to 15%.

CONCLUSION

In conclusion, presented here is an attribute-based method to transform unstructured safety incident and observation data into structured data, to be used in further analysis of construction safety. Attributes identified were objects, actions or site descriptors within the text which directly contributed to the incident or unsafety observation.

Manual annotation identified 250 unique attributes, of which 58 occurred in 1% or more of the dataset. Manual annotation is time-consuming and this task, labelling 2345 safety incident/observation descriptions, took over 300 solid man hours (not including breaks or moving from one description to another). Manual annotation is therefore not suitable for deployment. Automatic detection of these attributes is required to make this method viable in industry.

To automatically predict attributes from text, Natural Language Processing (NLP) was used to process the text into vector space representations, otherwise known as 'Bag of Words'. Machine learning classifiers were then trained to predict the attributes from these inputs. Of those investigated, ensemble classifiers performed best, notably SVM Bagging. Factors identified as influencing the accuracy of results are imbalance of classes and how these are dealt with; inclusion of less formal data types e.g., safety observation data alongside incident reports; and granularity of algorithm optimisation. Future research should carefully consider and address these points.

The use of NLP for analysis of free-text safety reports has been shown to be an emergent but potentially reliable science. While this research demonstrated the promise such methods hold, it also indicated the need for application by experts who understand the nuances of the methods and data biases. Additionally, a layman could be easily seduced by claims of high accuracy which could lead to the selection overly simplistic methods. This could have the unintended consequence of poor safety outcomes. There is more work to do before results can be relied upon.

This research is part of a piece of wider doctoral research which will optimise the methods employed and investigate further the potential knowledge discovery of this data for UK construction.

REFERENCES

Baker, H, Hallowell, M R and Tixier, A J-P (2020) Automatically learning construction injury precursors from text, *Automation in Construction*, (Accepted for publication, **118**, [Oct 2020])

Baker, H, Smith, S D, Velikova, M, Masterton, G and Hewlett, B (2018) *Learning from Failure: Processes and Attitudes in the Construction Industry*, 34th Annual ARCOM Conference (Working Paper), Belfast: UK, ARCOM

Bottou, L, Curtis, F E and Nocedal, J (2018) Optimization methods for large-scale machine learning, *Siam Review*, **60**(2), 223-311.

Chokor, A, Naganathan, H, Chong, W K and El, M (2016) Analysing Arizona OSHA injury reports using unsupervised machine learning, *Procedia Engineering*, **145**, 1588-1593.

Esmaeili, B, Hallowell, M R and Rajagopalan, B (2015) Attribute-based safety risk assessment II: Predicting safety outcomes using generalized linear models, *Journal of Construction Engineering Management*, **141**(8), 04015022.

Goh, Y M and Ubeynarayana, C U (2017) Construction accident narrative classification: An evaluation of text mining techniques, *Accident Analysis and Prevention*, **108**(May), 122-130.

Haixiang, G, Yijing, L, Shang, J, Mingyun, G, Yuanyue, H and Bing, G (2017) Learning from class-imbalanced data: Review of methods and applications, *Expert Systems with Applications*, **73**, 220-239

Hirschberg, J and Manning, C D (2015) Advances in natural language processing, *Science* **349**(6245), 261-316.

Jones, K S (2004) A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, **60**, 11-21.

Kim, T and Chi, S (2019) Accident case retrieval and analyses: Using natural language processing in the construction industry, *Journal of Construction Engineering and Management*, **145**(3), 04019004.

Marzouk, M and Enaba, M (2019) Text analytics to analyse and monitor construction project contract and correspondence, *Automation in Construction*, **98**(November 2018), 265-274.

Porter, M F (2001) *Snowball: A Language for Stemming Algorithms*. Available from https://tartarus.org/martin/PorterStemmer/index-old.html

Soibelman, L, Wu, J, Caldas, C, Brilakis, I and Lin, K Y (2008) Management and analysis of unstructured construction data types, *Advanced Engineering Informatics*, **22**(1), 15-27.

Tixier, A J-P, Hallowell, M R, Rajagopalan, B and Bowman, D (2016a) Application of machine learning to construction injury prediction, *Automation in Construction*, **69**, 102-114.

Tixier, A J-P, Hallowell, M R, Rajagopalan, B and Bowman, D (2016b) Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports, *Automation in Construction*, **62**, 45-56.

Velikova, M, Baker, H and Smith, S D (2018) The meaning of failure: Establishing a taxonomy of failure in the construction industry to improve organisational learning. *In:* Gorse, C and Neilson, C J (Eds.), *Proceedings 34th Annual ARCOM Conference*, 3-5 September 2018, Queen's University, Belfast, UK. Association of Researchers in Construction Management, 16–25.

Yu, W and Hsu, J (2013) Automation in construction content-based text mining technique for retrieval of CAD documents, *Automation in Construction*, **31**, 65-74.

Zhang, F, Fleyeh, H, Wang, X and Lu, M (2019a) Construction site accident analysis using text mining and natural language processing techniques, *Automation in Construction*, **99**, 238-248.

Zhang, F, Fleyeh, H, Wang, X and Lu, M (2019b) Construction site accident analysis using text mining and natural language processing techniques, *Automation in Construction*, **99**(June 2018), 238-248.

Zhong, B, Pan, X, Love, P E D, Ding, L and Fang, W (2020) Deep learning and network analysis: Classifying and visualizing accident narratives in construction, *Automation in Construction*, **113**(January), 103089.

Zou, Y, Kiviniemi, A and Jones, S W (2017) Retrieving similar cases for construction project risk management using Natural Language Processing techniques, *Automation in Construction*, **80**(April), 66-76.