# Action Sequencing in Construction Accident Reports using Probabilistic Language Model

**Q. Do[a], T. Le[a] and C. Le[b]**

[a]The Glenn Civil Engineering Department, Clemson University, USA
[b]Department of Civil, Construction and Environmental Engineering, North Dakota State University, USA
E-mail: qdo@clemson.edu, tuyenl@clemson.edu, chau.le@ndsu.edu

**Abstract –**

**Construction remains among the most hazardous workplaces, thus a significant amount of time and effort in reporting and investigating the accident occurrences has been done in the past decades by government agencies. In light of construction safety, analyzing textual information in construction accident records may assist in our comprehension of past data and be used to minimize future risks. Many attempts have been made in previous studies to identify causes and related entities but yet consider worker activities and behaviors. This study presents a framework that adopts a Probabilistic Language Model to sequence actions taken by workers that depict construction scenarios from unstructured accident narrative reports. The proposed approach achieved outstanding performances with the highest sequence accuracy and pairwise sequence accuracy of 84.81 % and 89.12%, respectively. Moreover, an action sequence database that can explain the relationship between workers' actions was created. This research is anticipated to contribute to enhancing understanding and establishing safety management systems to actively forecast and prevent accidents.**

**Keywords –**

**Construction safety; Accident reports; Probabilistic Language Model; Natural Language Processing**

## 1 Introduction

Construction sites remain among the most hazardous work environments for laborers. Despite several attempts driven by this relatively low-performance level, accident statistics have never really improved appreciably over the last decade [1]. These accidents raised major health and safety issues and substantial financial loss [2]. Hence, it is critical to gain a deeper insight into construction accidents to enhance safety performance. Over time, a vast amount of detailed information in the form of data would be gathered. Accident reports are essential but underutilized due to the difficulty of extracting data from unstructured text. Due to the critical significance of accident reports, more focus has lately been put on ensuring the reliability of the data collected and report organization. Construction accident reports are valuable sources of information, and the process of assessing them may provide critical insight into previous events to avoid future recurrences.

Numerous scholars have worked to build automated models accompanied with Natural Language Processing (NLP) that might be used to analyze textual data contained in construction accident reports with the minimum human intervention. Serval studies [3]–[8] adopted a supervised approach to classifying the causes, types of injury, and injured body parts. Besides, another study [9] examined the efficacy of an unsupervised approach for clustering accident reports. Afterward, the authors monitored the results and retrieved pertinent information about the objects and factors contributing to the incidents. The studies, as mentioned above, have achieved outstanding success, and their findings have made significant contributions to advancing safety knowledge and improving safety plans. However, a crucial factor that has not been taken into consideration is the worker's sequence of actions that reflects the construction scene at the time of an accident.

The key motivation for this paper is to address that research gap. This study adopted the Probabilistic Language Model to develop an Action Sequencing Model that can sequence the actions taken by workers from unstructured accident reports obtained from the Occupational Safety and Health Administration (OSHA). Accident reports were split into separate sentences, and each sentence was annotated to a particular action label before feeding as input for the model. The major contribution of this work is an action sequence database that can explain the relationship between actions showing the scene of the construction accident. Exploring this database can help widen the horizons and develop a safety management system to predict and prevent catastrophes actively.

# 2 Background

## 2.1 Probabilistic Language Model

Models that can be utilized to assign a probability to a sentence or a sequence of words are called Probabilistic Language Models [10]. Probabilistic Language Models have been employed in a variety of research fields to date in numerous NLP applications, such as Handwriting Recognition [11], Machine Translation [12], Speech Recognition [13], Spelling Correction [14], and Information Retrieval [15]–[17]. N-gram models, commonly referred to as Markov models, are detailed in the following section.

### 2.1.1 N-Gram Language Models

Given a sequence of words $W(w_1,w_2,...w_n)$, a model that calculates the probability of either $P(W)$ or $P(w_n|w_1,w_2,...w_{n-1})$ is called a Probabilistic Language Model.

To decompose these probabilities, the chain rule of probability is applied. The chain rule of probability is a theory that allows calculating any member of a joint distribution of random variables using conditional probabilities. Given $n$ event (i.e., $x_1, x_2,...x_n$), the probability $P(x_1, x_2,...x_n)$ is

$$P(x_1, x_2, ...x_n) = P(x_1)P(x_2 \mid x_1)...P(x_n \mid x_1, x_2...x_{n-1}) \quad (1)$$

The sequence event $x_1, x_2,...x_n$ can be represented as $x_{1:n}$. The equation (1) is rewritten:

$$\begin{aligned} P(x_{1:n}) &= P(x_1)P(x_2 \mid x_1)...P(x_n \mid x_{1:n-1}) \\ &= \prod_{k=1}^{n} P(x_k \mid x_{1:k-1}) \end{aligned} \quad (2)$$

Applying the chain rule to the sequence of words $W$:

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_{1:2})...P(w_n \mid w_{1:n-1}) \\ &= \prod_{k=1}^{n} P(w_k \mid w_{1:k-1}) \end{aligned} \quad (3)$$

The chain rule emphasizes the link between calculating the joint probability of a sequence and computing the conditional probability of a word given previous words. Equation (3) suggests estimating the joint probability of an entire sequence of words by multiplying the number of conditional probabilities together. However, it is challenging to calculate the exact probability of a word given a long sequence of preceding words $P(w_n|w_{1:n-1})$.

The assumption that the probability of a word depends solely on the preceding word(s) is known as the Markov assumption. Markov models, also known as N-gram models, are the class of probabilistic models that presume that we can estimate the probability of some future items without referring too far into the past [18]. Then we approximate the probability of a word given its entire context as follows:

$$P(w_n \mid w_{1:n-1}) \approx P(w_n \mid w_{n-N+1:n-1}) \quad (4)$$

What method do we use to calculate N-gram probabilities? An intuitive method to estimate probabilities is called Maximum Likelihood Estimation (MLE). We obtain the MLE estimation for the parameters of an N-gram model by getting counts from a corpus and normalizing the counts so that they lie between 0 and 1 [10].

$$P(w_n \mid w_{n-N+1:n-1}) = \frac{C(w_{n-N+1:n-1}w_n)}{C(w_{n-N+1:n-1})} \quad (5)$$

Equation (5) estimates the N-gram probability by dividing the observed frequency of a particular sequence by the observed frequency of a prefix. This is known as a relative frequency ratio.

Because probabilities are less than or equal to one, multiplying probabilities together results in a smaller product. In practice, using log probabilities rather than raw probabilities can assist obtain figures that are not as small.

### 2.1.2 Smoothing Techniques

Smoothing is a technique for creating an approximation function that tries to capture essential patterns in data while eliminating noise and other fine-scale structures/rapid events [19]. In Probabilistic Language Model, the MLE of probabilities generally results in overfitting training data and poor performance on unseen data. It is preferable to utilize smoothed estimates of these values instead [20]. In some cases, an N-gram is never observed in the training data, resulting in the zero probability of a sequence of words. To avoid the model from assigning 0 probability to these unseen items, we must take a bit of probability from some more frequent items and give it to the items that have never been observed. This modification is called smoothing. A large number of other smoothing techniques for N-gram models have been proposed, such as Laplace Smoothing [21], Add-k smoothing [22], Stupid backoff [23], and Kneser-Ney smoothing [24].

## 2.2 Action Sequencing

Sequencing actions from natural language text intended for human consumption is difficult since it does not contain a time series attribute and needs agents to comprehend complicated contexts of actions. Husari et al. [25] proposed a framework called *ActionMiner* that combined Entropy and Mutual Information with some basic NLP techniques to extract threat actions from Cyber Threat Intelligence reports and achieved good performance. However, this study only extracted all actions to the list and cannot analyze their relationship or sequence. Manshadi et al. [26] developed a probabilistic language model and used the predicate-argument pair

(verb-object; E.g., got-tire) to represent an action. This model can capture the expected sequences in simple narrative texts which have very few verbs in the corpus of Weblog Stories. In other studies, Feng et al. [27] proposed a novel approach EASDRL to automatically extract action sequences from texts based on deep reinforcement learning, and Mei et al. [28] adopted an encoder-decoder model with long short-term memory recurrent neural networks (LSTM-RNN) translates natural language instructions to action sequences. These works can extract meaningful action sequences from complicated sentences in free natural language; however, input data require that the order of sentences corresponds to the sequence of actions. Due to this limitation, it is hard to apply unstructured textual data such as accident reports. To deal with these restrictions, various significant efforts [29]–[33] employing the state-of-the-art machine learning algorithms for the task Sentence Ordering and Coherence can be taken into account before extracting action sequences. Nevertheless, these models only performed well for judging the order of sentence pairs and achieved relatively poor performance on the whole paragraph; hence, the application of models [27] and [28] would not really be feasible.

### 2.3 Related Studies

In the construction domain, numerous studies were conducted by researchers to explore the accident reports. Tixier [3] developed an automated model based on keyword dictionaries and R functions. This model is capable of scanning textual injury reports and extracting precursors, injury types, energy sources, and body parts with an accuracy of 95%. Goh et al. [4] adopted six machine learning algorithms, including support vector machine (SVM), linear regression (LR), random forest (RF), k-nearest neighbor (KNN), decision tree (DT), and Naive Bayes (NB), to classifying accident reports into 11 predefined labels of causes. This research indicated the good performance of SVM compared to others; however, the performance metrics were not good. In other research, Cheng [5] proposed a hybrid supervised machine learning named Symbiotic Gated Recurrent Unit (SGRU) for the task of categorizing 1000 construction reports into 11 unique label causes; the result exhibited significant improvements to the previous study. Recently, Zhong et al. [6] employed Convolutional Neural Network to classify accident narratives automatically. The authors later used The Latent Dirichlet Allocation (LDA) model to analyze and visualize the relationship of causes and related objects. The results provide valuable insights from text data. Chokor et al. [9] conducted a K-means clustering unsupervised approach to classify construction injury reports. Four types of accident causes were identified, including fall, struck by objects, electrocutions, and trench collapse. The aforementioned research is solely concerned with determining the causes and frequent objects causing accidents, not extracting the sequence of actions taken by workers associated with accidents which might be crucial to enhance safety management.

This study addresses the research gap in previous studies; we developed an Action Sequencing Model that can sequence actions from accident reports. Our model can deal with the problem of complicated sentences and unstructured text without any effort of reordering actions and sentences. The result is able to identify potential relationships concerning the occurrences and describe the associated behaviors of workers that reflect the construction scene at the time of an accident.

## 3 Methodology

This study adopted the Probabilistic Language Model for developing an Action Sequencing Model (as depicted in Figure 1). To begin with, data preparation is to develop the datasets for training and evaluating the model. Several steps were then utilized for training the Action Sequencing Model before model evaluation was implemented.

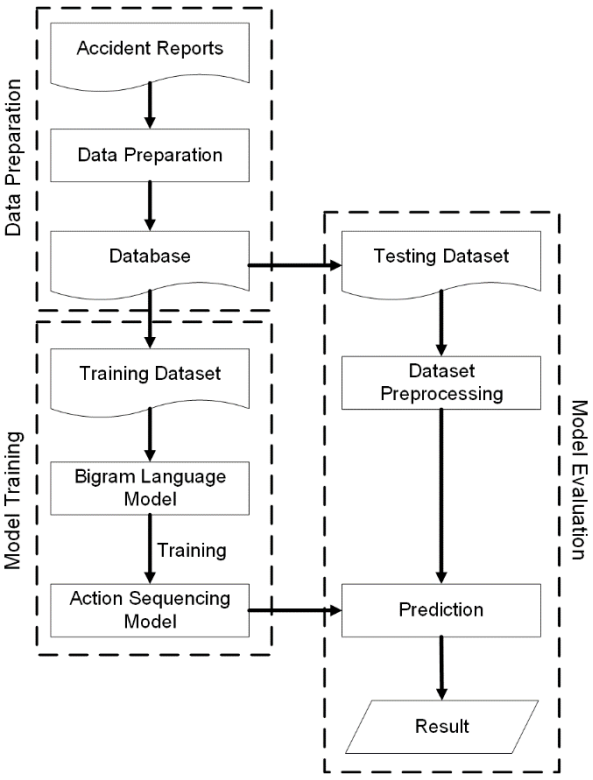


Figure 1. Methodology of Action Sequencing Model

### 3.1 Dataset Preparation

The accident reports are freely available online from the OSHA website [34]. In this study, accident reports, including a thorough account of construction site incidents, were picked and saved in a Microsoft Excel file. A sample size of 328 accident reports was chosen, and each report was split into separate sentences. As a result, the raw dataset of 1,689 sentences was developed. An accident report (also known as Accident Investigation Summary) is written by inspectors in free natural language to briefly describe some of the main points of an accident, so it mostly contains simple and short sentences.

| Accident report | Statements | Level 1 label | Level 2 label |
|---|---|---|---|
| Employee #1 was engaged in the demolition of a structural steel amusement ride at a theme park. Employee #1 fell approximately 50 ft through a deck hole measuring approximately 2-ft by 8-ft that was created from a gear motor that had been cut and removed from the structure by the crew. Employee #1 was killed. | Employee #1 was engaged in the demolition of a structural steel amusement ride at a theme park. | Action | Demolition |
| | Employee #1 fell approximately 50 ft through a deck hole measuring approximately 2-ft by 8-ft that was created from a gear motor that had been cut and removed from the structure by the crew. | Event | Fall |
| | Employee #1 was killed. | Consequence | Fatality |

Figure 2. Sample accident reports and labeled statements

Sentence labeling is the second step after splitting accident reports. Since accident reports are written in free natural language, it is difficult to directly identify and access vast amounts of information. The authors analyzed thoroughly and determined that, despite being unstructured text, each accident report contains three key pieces of information: sentences mention actions before the accident, sentences describe accident event, and sentences provide subsequent results. Therefore, the extracted sentences were annotated into the predefined level 1 labels, namely Action, Event, and Consequence, for grouping information. Following that, the sentences in each level 1 label were annotated into level 2 labels for the task sequences extraction (as shown in Figure 2). Aside from the summary of the incident, each accident report obtained from OSHA provides additional information such as diagnosis, cause, degree (bruise, fatality, etc.), occupation etc., which the authors referred to and reviewed for the unique labels of the statements. The authors also reused and calibrated many labels from OSHA definitions to establish the labels in the dataset. In some small number of situations, if a statement contained information that might be considered as multi-label categories and could not be aligned with one unique label based on OSHA additional information, a unique label was assigned according to the principle of identifying the most significant contribution to the accident compared to the others. Figure 3 depicts the label diagram for the final labeling result in this study. As a result, the Action group has 30 unique labels, while Event and Consequence have the same number of unique labels of 12. Since the findings of dataset preparation occupy a large space and the paper length is limited, the authors could only show some typical labels. Among level 2 labels, "None" is a specially designed label that presents noise information in sentences without specific action, ambiguities or provides general information such as the sentence "There were no witnesses to the accident" or "The pit measured approximately 5 feet to 8 feet deep".

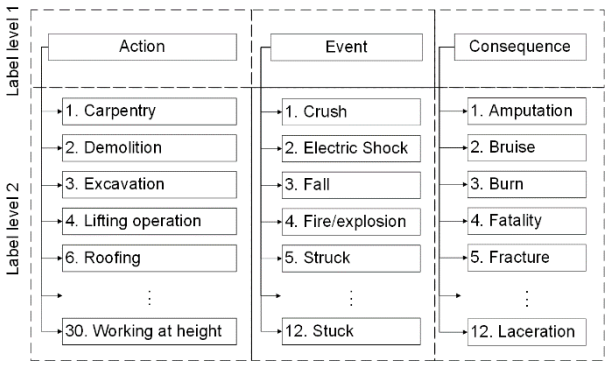


Figure 3. Labels of sentences extracted from accident reports

The next two important steps in data preparation are Dropping None label and Changing label format. As mentioned above, sentences labeled as None provided irrelevant information to extract sequences. This dropping helped filter and keep crucial extracted information. Some labels have more than one word, such as "Lifting operation" and "Working at height;" changing format step facilitated long word labels treated as one word (token) by adding underlines to link words. As a result, "Lifting operation" and "Working at height" were converted to "Lifting_operation" and "Working_at _height," respectively. For this study, the authors read carefully and annotated the actual sequence of each accident report using level 2 labels and the order of action as a reference to develop and evaluate the model. A sequence is exhibited by a sequence of words. For example, an actual sequence with three elements is presented as "Carpentry Fall Bruise"; it can be interpreted as a sequence of actions Carpentry → Fall → Bruise. Table 1 shows the distribution of the length of the actual action sequence.

Table 1. The distribution of the length of the actual action sequence

| The length of action sequence | The number of action sequence | Percentage |
|---|---|---|
| 2 elements | 15 | 4.57% |
| 3 elements | 147 | 44.82% |
| 4 elements | 118 | 35.98% |
| 5 elements | 35 | 10.67% |
| 6 elements | 12 | 3.66% |
| 7 elements | 0 | 0% |
| 8 elements | 1 | 0.3% |
| Total | 328 | 100% |

Finally, the final database was randomly split into 75% and 25% for the training and testing datasets, respectively.

## 3.2 Model Training

### 3.2.1 Bigram Language Model

In this study, the authors adopted bigram (N=2) to train the N-gram Language Model. The Bigram Language Model is the underlying Probabilistic Language Model, which has a wide application. As mentioned, the authors annotated the actual sequence of each accident report to train the model. Each element (level 2 label) of the actual sequence plays a role as a unigram, while a bigram is a sequence of two adjacent elements. Hence, the probability of an individual unigram given the bigram assumption:

$$P(w_n \mid w_{1:n-1}) \approx P(w_n \mid w_{n-1}) \tag{6}$$

The chain rule to the sequence of unigrams *W:*

$$P(w_{1:n}) = \prod_{k=1}^{n} P(w_k \mid w_{k-1}) \tag{7}$$

where:

$$P(w_k \mid w_{k\text{-}1}) = \frac{C(w_{k\text{-}1} w_k)}{C(w_{k\text{-}1})} \tag{8}$$

*C* is the frequency (count) of each pattern in the corpus.

Equation (8) calculates the probability of a bigram by dividing the observed frequency of this bigram by the observed frequency of the first unigram belonging to this bigram. This probability is also known as a relative frequency ratio. For example, to calculate the probability of a bigram "Struck Fall," we need to get the counts of bigram "Struck Fall" and unigram "Struck" from the corpus level 2 labels. Afterward, we calculate the division of these two values.

$$P(\text{Fall} \mid \text{Struck}) = \frac{C(\text{Struck Fall})}{C(\text{Struck})}$$

P(Fall | Struck) denotes the probability of the unigram Fall given the unigram Struck; it is also known as P(Struck Fall). It can be interpreted as the probability of "Fall" occurring after "Struck."

### 3.2.2 Smoothing Technique

Laplace Smoothing is used in this study to deal with the zero Bigram probability. This is the simplest and quickest technique to smooth data by adding one to all Bigram counts before normalizing them to probabilities. The probability of an individual unigram in equation (8) is expressed as:

$$P(w_k \mid w_{k\text{-}1}) = \frac{C(w_{k\text{-}1} w_k) + 1}{C(w_{k\text{-}1}) + V} \tag{9}$$

where V denotes the vocabulary, the set of all unigrams under consideration.

### 3.2.3 Training Action Sequencing Model

This study using Bigram Language Model developed the Bigram Sequence Probability Database as a root for training the Action Sequencing model:

- A Bag of Bigram was created based on a corpus of actual sequences retrieved from the training dataset.
- Adopt MLE as shown in equations (8) and (9) to estimate the probabilities of all bigrams in the Bag of Bigram. These probabilities are also known as the probabilities of the sequence of two actions. The obtained database is called the Bigram Sequence Probability Database.
- Apply the chain rule in equation (7) to calculate the probability of the action sequences. The probability of the bigrams retrieved from the Bigram Sequence Probability Database.

The obtained Bigram Sequence Probability Database contains all bigrams (sequence of two elements level 2 label) along with their probabilities that illustrate their likelihood. As a matter of fact, the resulting database is not only a component of the Action Sequencing Model but still has practical implications. For example, when considering what are immediately potential consequences following the event "Fall;" querying the Bigram Sequence Probability Database, we can achieve all results such as "Fracture" occupies the highest probability with P(Fracture|Fall) = 0.3, "Bruise" with P(Bruise|Fall) = 0.05, and "Fatality" with P(Fatality|Fall) = 0.1. This retrieval provides insight and enhances our understanding of all possible outcomes and what is most likely to happen for the prediction task.

## 3.3 Model Evaluation

Model evaluation is to evaluate the performance of the trained model on the testing dataset. The process includes preprocessing the testing dataset, prediction and evaluating results.

Firstly, data preprocessing was performed on the developed testing dataset by following steps:

- Concatenating labels (Generating preliminary sequence): The labels of sentences of each accident report were concatenated into a sequence for which the order of labels corresponds to the order of sentences. For example, an accident report contains the order of sentences corresponding to labels "Carpentry," "Struck," "Fracture," "Fall"; the obtained sequence after concatenating is "Carpentry Struck Fracture Fall."
- Generating permutations: The preliminary sequence of each accident report obtained from the Concatenating labels step was used to produce all possible sequences. For example, the sequence "Carpentry Struck Fracture" can be generated as "Carpentry Struck Fracture," "Carpentry Fracture Struck," "Struck Carpentry Fracture," "Struck Fracture Carpentry," "Fracture Struck Carpentry" and "Fracture Carpentry Struck."

Figure 4 presents the workflow of the Action Sequence Prediction. The possible sequences got from the sequence permutation step were fed as input for the Action Sequencing Model. The output is the probability corresponding to each sequence. Eventually, Action sequence prediction was implemented by voting the permutation that had the highest probability.

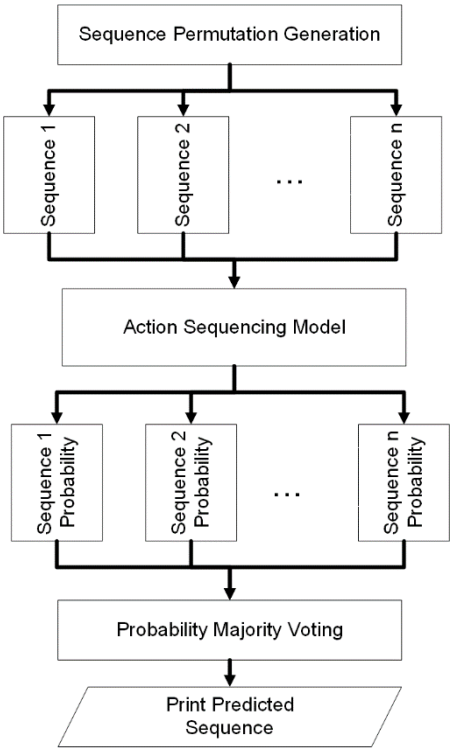


Figure 4. Workflow of the Action Sequence Prediction

Lastly, to evaluate the results (predicted orders), we used two types of metrics: Sequence Accuracy (SA) and Pairwise Sequence Accuracy (PSA):

- **Sequence Accuracy (SA):** It measures the percentage of actions sequence that is correctly predicted.
- **Pairwise Sequence Accuracy (PSA):** this metric calculates the percentage of bigrams for which the relative order is predicted correctly. In other words, PSA is the ratio of the number of correct ordered word pairs and total possible word pairs.

## 4 Results and Discussion

This section presents the performance of the Action Sequencing Model. We developed two models, the first model is the baseline model without applying a smoothing technique, and the second model adopted a smoothing technique.

Our evaluation was based on sequence prediction on the testing set, and Table 2 shows the excellent performances. To begin with, the baseline model has a sequence accuracy of 74.68%, which is not too remarkable but sufficient for a successful action sequencing task. Besides, the pairwise sequence accuracy shows the ideal value of 80.83%. The second model with smoothing technique has over 10% higher than the baseline model in sequence accuracy with 84.81%. In terms of pairwise sequence accuracy, its metric is 89.12% indicating a robust value compared to the baseline model.

Table 2. Performance results of the Probabilistic Language Model

| Performance Metrics | Modeling without Smoothing Technique | Modeling with Smoothing Technique |
|---|---|---|
| SA (%) | 74.68 | 84.81 |
| PSA (%) | 80.83 | 89.12 |

Overall, these evaluations demonstrate that the Probabilistic Language Model is robust for developing the Action Sequencing Model and the application of the smoothing technique resulting in better performance metrics. A number of incorrect predictions are primarily due to long, complicated sequences, which made the model confusing. This error source can be observed through the difference between PSA and SA, where models performed well for judging the order of sequence pairs but operated poorer on the whole sequence. For example, the actual sequence of an accident report is "Roofing Fall Fall Struck Crush Fatality"; however, the prediction is "Roofing Fall Struck Fall Crush Fatality." It is easy to see that there are three correct pairwise sequences, including "Roofing Fall," "Fall Struck," and

"Crush Fatality" out of 5 pairwise sequences. This error is able to be mitigated by expanding the dataset and applying more types of N-grams instead of bigram.

## 5 Conclusion

Construction accident reports are valuable documentation data, and analyzing them may give critical knowledge of prior occurrences to prevent unanticipated recurrence catastrophes. This study presents a framework that adopts a Probabilistic Language Model to sequence actions taken by workers that depict the construction scene at the time of accident from unstructured accident narrative reports. The Action Sequencing Model can deal with the problem of complicated sentences and unstructured text without any effort of reordering actions and sentences. This study produced excellent results with the highest sequence accuracy and pairwise sequence accuracy of 84.81 % and 89.12%, respectively, which illustrate the good performance for both judging the order of sequence action pair and the whole sequence actions of each record.

This research provides threefold contributions to the body of knowledge. To begin with, a reliable automated model was developed that can exploit various action relationship information from construction accident records. Secondly, a dataset was built and potentially used for further research in construction safety interest. Lastly, a sequence action database was formed in the final result that can explain the relationship between workers' actions at the time of accidents. This database can be adopted in the Sequence Mining task to provide a probabilistic forecast of likely next actions for a given action or sequence of actions. In terms of Industry implications, construction organizations can employ this automated model to analyze the sequence of action information in accident reports that generate consistent results and save time and resources. This information is used to establish safety management systems to actively forecast and prevent accidents on construction sites.

The results of this research were encouraging; however, some aspects can be further optimized in the future. A dataset size used in training is small; thus, expanding in size is needed to generalize the result. In addition, the use of trigram or more longer grams instead of bigram can potentially achieve better performance. Finally, this study introduced a simple probabilistic language model. The state-of-the-art machine learning algorithms might be incorporated into the probabilistic language model resulting in a hybrid model. The neural probabilistic language model would be a desirable objective for the action sequencing task.

## References


[1] Bureau of Labor Statistics (BLS), "National Census of fatal occupational injuries." https://www.bls.gov/news.release/pdf/cfoi.pdf (accessed Feb. 20, 2022).

[2] C. U. Ubeynarayana and Y. M. Goh, "An ensemble approach for classification of accident narratives," in Computing in Civil Engineering 2017.

[3] A. J. P. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman, "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, Feb. 2016, doi: 10.1016/j.autcon.2015.11.001.

[4] Y. M. Goh and C. U. Ubeynarayana, "Construction accident narrative classification: An evaluation of text mining techniques," Accident Analysis and Prevention, vol. 108, pp. 122–130, Nov. 2017, doi: 10.1016/j.aap.2017.08.026.

[5] M. Y. Cheng, D. Kusoemo, and R. A. Gosno, "Text mining-based construction site accident classification using hybrid supervised machine learning," Automation in Construction, vol. 118, Oct. 2020, doi: 10.1016/j.autcon.2020.103265.

[6] B. Zhong, X. Pan, P. E. D. Love, L. Ding, and W. Fang, "Deep learning and network analysis: Classifying and visualizing accident narratives in construction," Automation in Construction, vol. 113, May 2020, doi: 10.1016/j.autcon.2020.103089.

[7] H. Baker, M. R. Hallowell, and A. J. P. Tixier, "AI-based prediction of independent construction safety outcomes from universal attributes," Automation in Construction, vol. 118, Oct. 2020, doi: 10.1016/j.autcon.2020.103146.

[8] B. Zhong, X. Pan, P. E. D. Love, J. Sun, and C. Tao, "Hazard analysis: A deep learning and text mining framework for accident prevention," Advanced Engineering Informatics, vol. 46, Oct. 2020, doi: 10.1016/j.aei.2020.101152.

[9] A. Chokor, H. Naganathan, W. K. Chong, and M. el Asmar, "Analyzing Arizona OSHA Injury Reports Using Unsupervised Machine Learning," in Procedia Engineering, 2016, doi: 10.1016/j.proeng.2016.04.200.

[10] D. Jurafsky, "Speech & language processing," Pearson Education India, 2000.

[11] R. Plamondon and S. N. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 1, pp. 63–84, 2000.

[12] D. Chiang, "A hierarchical phrase-based model for statistical machine translation," in Proceedings of the 43rd annual meeting of the association for computational linguistics, 2005.
[13] A. Stolcke, "SRILM-an extensible language modeling toolkit," 2002.
[14] E. Mays, F. J. Damerau, and R. L. Mercer, "Context based spelling correction," Information Processing & Management, vol. 27, no. 5, pp. 517–522, 1991.
[15] B. Croft and J. Lafferty, "Language modeling for information retrieval," vol. 13. Springer Science & Business Media, 2003.
[16] S. Missaoui, M. Viviani, R. Faiz, and G. Pasi, "A language modeling approach for the recommendation of tourism-related services," in Proceedings of the Symposium on Applied Computing, 2017.
[17] J. M. Ponte and W. B. Croft, "A language modeling approach to information retrieval," in ACM SIGIR Forum, 2017.
[18] K. Armeni, R. M. Willems, and S. L. Frank, "Probabilistic language models in cognitive neuroscience: Promises and pitfalls," Neuroscience and Biobehavioral Reviews, vol. 83. Elsevier Ltd, pp. 579–588, Dec. 01, 2017. doi: 10.1016/j.neubiorev.2017.09.001.
[19] V. Cherian and M. S. Bindu, "Heart disease prediction using Naive Bayes algorithm and Laplace Smoothing technique," Int. J. Comput. Sci. Trends Technol, vol. 5, no. 2, pp. 68–73, 2017.
[20] S. F. Chen and R. Rosenfeld, "A survey of smoothing techniques for ME models," IEEE Transactions on Speech and Audio Processing, vol. 8, no. 1, pp. 37–50, 2000, doi: 10.1109/89.817452.
[21] E. R. Setyaningsih and I. Listiowarni, "Categorization of Exam Questions based on Bloom Taxonomy using Naïve Bayes and Laplace Smoothing," in 3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021, Apr. 2021, doi: 10.1109/EIConCIT50028.2021.9431862.
[22] Marrara, S., Pasi, G., Viviani, M., Cesarini, M., Mercorio, F., Mezzanzanica, M., & Pappagallo, M, "A language modelling approach for discovering novel labour market occupations from the web," Apr. 2017, doi: 10.1145/3106426.3109035.
[23] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large language models in machine translation," 2007.
[24] R. Pickhardt, T. Gottron, M. Körner, P. G. Wagner, T. Speicher, and S. Staab, "A Generalized Language Model as the Combination of Skipped n-grams and Modified Kneser-Ney Smoothing," Apr. 2014.
[25] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, "Using entropy and mutual information to extract threat actions from cyber threat intelligence," in 2018 IEEE International Conference on Intelligence and Security Informatics (ISI), 2018.
[26] M. Manshadi, R. Swanson, and A. S. Gordon, "Learning a Probabilistic Model of Event Sequences from Internet Weblog Stories.," in FLAIRS Conference, 2008.
[27] W. Feng, H. H. Zhuo, and S. Kambhampati, "Extracting action sequences from texts based on deep reinforcement learning," 2018.
[28] H. Mei, M. Bansal, and M. R. Walter, "Listen, attend, and walk: Neural mapping of navigational instructions to action sequences," 2016.
[29] X. Chen, X. Qiu, and X. Huang, "Neural sentence ordering," 2016.
[30] Y. Yin, L. Song, J. Su, J. Zeng, C. Zhou, and J. Luo, "Graph-based neural sentence ordering," 2019.
[31] Y. Zhu, K. Zhou, J.-Y. Nie, S. Liu, and Z. Dou, "Neural Sentence Ordering Based on Constraint Graphs," 2021.
[32] L. Logeswaran, H. Lee, and D. Radev, "Sentence ordering and coherence modeling using recurrent neural networks," 2018.
[33] S. B. R. Chowdhury, F. Brahman, and S. Chaturvedi, "Is Everything in Order? A Simple Way to Order Sentences," 2021.
[34] Occupational Safety and Health Administration (OSHA), "Fatality and Catastrophe Investigation Summaries." https://www.osha.gov/pls/imis/accidentsearch.html (accessed Dec. 11, 2021).