knowledge

MDPI

Article

The Importance of Specific Phrases in Automatically Classifying Mine Accident Narratives Using Natural Language Processing

Rambabu Pothina * and Rajive Ganguli

Department of Mining Engineering, University of Utah, Salt Lake City, UT 84112-0102, USA; rajive.ganguli@utah.edu
* Correspondence: rambabu.pothina@utah.edu

Abstract: The mining industry is diligent about reporting on safety incidents. However, these reports are not necessarily analyzed holistically to gain deep insights. Previously, it was demonstrated that mine accident narratives at a partner mine site could be automatically classified using natural language processing (NLP)-based random forest (RF) models developed, using narratives from the United States Mine Safety and Health Administration (MSHA) database. Classification of narratives is important from a holistic perspective as it affects safety intervention strategies. This paper continued the work to improve the RF classification performance in the category "caught in". In this context, three approaches were presented in the paper. At first, two new methods were developed, named, the similarity score (SS) method and the accident-specific expert choice vocabulary (ASECV) method. The SS method focused on words or phrases that occurred most frequently, while the ASECV, a heuristic approach, focused on a narrow set of phrases. The two methods were tested with a series of experiments (iterations) on the MSHA narratives of accident category "caught in". The SS method was not very successful due to its high false positive rates. The ASECV method, on the other hand, had low false positive rates. As a third approach (the "stacking" method), when a highly successful incidence (iteration) from ASECV method was applied in combination with the previously developed RF model (by stacking), the overall predictability of the combined model improved from 71% to 73.28%. Thus, the research showed that some phrases are key to describing particular ("caught in" in this case) types of accidents.

Keywords: mine safety and health; accidents; narratives; machine learning; natural language processing; random forest classification; heuristic approach; expert analysis; vocabulary lists

check for updates

Citation: Pothina, R.; Ganguli, R. The Importance of Specific Phrases in Automatically Classifying Mine Accident Narratives Using Natural Language Processing. *Knowledge* **2022**, *2*, 365–387. https://doi.org/10.3390/knowledge2030021

Academic Editors: Jerry Chun-Wei Lin and Haridimos Kondylakis

Received: 30 April 2022
Accepted: 21 July 2022
Published: 29 July 2022

1. Introduction

Natural language processing (NLP) is a powerful tool in processing text and has been an area of intense focus in terms of research and application since the 1990s [1–3]. However, its application is relatively new to the mining industry and mine safety. The NLP tools are capable of processing huge amounts of text in relatively quicker times when compared to human subjects. This is a huge advantage, as insights can be gained quickly and without using much human resources. The reason accident narratives are not analyzed at mine sites is because most lack the human resources necessary for the task. The insights can then be used to deploy intervention strategies. At the start of the 21st century, an application of NLP to safety data was demonstrated by the Pacific Northwest National Laboratory (PNNL). The laboratory's team has successfully analyzed huge amounts of safety reports from the National Aeronautics and Space Administration's (NASA) aviation safety program to gain valuable insights [4].

Mine sites in the US are required to report details of certain types of accidents to Mine Safety and Health Administration (MSHA). In turn, MSHA maintains an accident database with concise descriptions of such reported accidents (narratives) along with other metadata [5]. The database is a valuable resource for mine safety professionals in creating the text “corpus”, which can help in “training” the machine learning (ML) models. Classification of narratives into their respective accident types is an important step in accident analysis. In their past research, authors have demonstrated how NLP-based ML models (random forests) developed using the MSHA corpus can be used effectively on non-MSHA narratives [6]. This was an important accomplishment, as developing NLP models on site-specific narratives are expected to be difficult for two reasons. First, a site may not have the variety in narratives that would be necessary to develop good NLP models. Second (and more importantly), however, unless a mine categorizes accidents, NLP modeling would require a human tagging of historical narratives. Tagging is when a narrative is concisely summarized in a few words. The tags essentially serve as the “meaning” of the narrative. Tagging is an expensive and limiting part of NLP. All narratives in the MSHA database are tagged on entry, thereby making the database a convenient corpus for NLP. However, until the previous work was published [6], it was unclear if models developed on the MSHA corpus would be applicable on a non-MSHA corpus. Thus, the previous work will encourage researchers and mine sites to develop NLP models exploiting the tagged MSHA corpus, knowing that the resultant models would apply to their sites. This paper continues the work to improve the performance of the NLP models. To start with, a “caught in” accident category was chosen.

There are certain limitations that prevent the NLP-based models from achieving their full potential in terms of high prediction success. The diversity in industry-specific safety language and narrators’ individual writing styles are some examples. Moreover, the source and circumstances of an injury differ from industry to industry [7,8], resulting in differences in vocabulary. In this context, it can be anticipated that accident classification models developed for one industry may have varied success rates in other industries. In the same context, expert or domain-specific knowledge can be helpful in gaining a deep understanding of accident circumstances and underlying mechanisms, which in turn can help improve classification success rates. For instance, according to the US Occupational Health and Safety Administration (OSHA), “caught in between” type accidents are those that involve, typically, a person or a body part being squeezed, caught, crushed, pinched, or compressed between two or more objects [9]. From the NLP standpoint, the actions that describe the “crushing” effect are what define a “caught in between” accident. Hence, the list of words or verbs that describe such actions (vocabulary lists) are necessary in training the NLP-based classification models. Key to success in NLP is thus finding the words that are essential to describing a particular class of narratives.

Domain-specific knowledge is highly valued and regarded as an essential tool for safety professionals working in the mining industry [10] as well as in any other industry [11]. This is the reason exploitation of domain-specific words and phrases, or in short form, domain knowledge elements (DKE), is common [12]. For instance, using a semantic rule-based NLP approach, Xu et al., 2021 [12] found DKEs that can provide valuable insights in analyzing the text in a Chinese construction safety management system. The process involves finding specific compound parts, parts-of-speech (pos) tagging, and dependency of words (DOW) that are important to understand the topic (construction safety)-specific language. The models developed in this paper leveraged such domain-specific knowledge elements.

Data preparation for ML models pose certain challenges when text handling is involved. Since the models can only operate on numbers, vectorization of words is necessary. At first, the text (or narratives) are split into unique words (unigram ‘tokens’) and then into representative “word vectors” of real numbers. The numbers signify an occurrence (1), no occurrence (0), or a number of occurrences (n) of tokens. Bag of words (BOW) models are popular in this context. They can reduce each accident narrative into a simple vector of words and their corresponding frequency of occurrences [13]. Due to their simplicity, these vectors take much less computer memory compared to other models. However, BOW models fail to account for the similarity or rarity of words in a narrative, which is an important aspect of classification problems. The mere list or bag of words from processing a narrative cannot convey the true meaning of a narrative. Semantic relationship of words, such as their occurrences in close proximity, context, and order, matters. For instance, the phrases "caught between rollers and fell" and "fell and caught between rollers" have the same vocabulary from a BOW model perspective, but the connotations are different. This poses a problem for classification algorithms.

The "word embedding" concept compensates for the BOW model's shortcomings by vectorizing similar words ("features") with similar scores [14]. The underlying concept is that linguistic items with similar distributions or words that occur in similar contexts have similar meanings [15]. For instance, the terms "injury", "accident", and "pain" are represented as being closer in vector space when compared to words such as "surface", "path", and "pavement". The models can also vectorize the word frequencies in a narrative in comparison to other narratives. For instance, based on how frequently a particular word occurs in a narrative, it can be set to carry more "weight" in terms of score representation in the vector. There are several types of word embedding methods and models that are available with slightly different vectorization strategies. For instance, in order to find the relevance of a word to a particular document or block of text, a term frequency–inverse document frequency (tf-idf) method is used. Depending on how frequent (term frequency) a word occurs and how commonly (score: 0) or rarely (score: 1) the word is found in a document, tf-idf method scores the words. This is a very important tool in "text mining" and is used in search engines. Several researchers used tf-idf-based models in analyzing OSHA accident data [16,17]. Some models developed were generalized and even deployed to analyze mining and metal industry accident data [17].

Word embedding models are to some extent limited by a meaning conflation deficiency. The problem stems from the representation of words that have multiple meanings in a single vector [18]. For instance, the word "pin" as a noun represents a pin. As a verb in the phrase "pinned between two objects", it conveys a different meaning. A "pinner" in the underground mining context, however, implies "a roof bolter". A word represented in its true sense ("sense representation") may be the solution for the problem. Collados et al., 2018 [18] presented a comprehensive overview on the two major types of sense representation models in the area, that are, unsupervised and knowledge-based. The former model depends on automatic word processing by algorithms looking for different senses of a word in the given context of words. The latter depends on expert-made resources such as WordNet, Wikipedia, and so on. In this context, expert-knowledge-based word clusters are used for training the algorithms in this paper in order to classify accident types.

"Pretraining" compensates for a shortage of training data, which is one of the biggest challenges for the NLP community currently [19]. In pretraining, models are trained on very large datasets, such as millions and billions of annotated text examples. For instance, Google's Word2Vec is a "pretrained word embedding" technique that is trained on the Google News dataset (which consists of about 100 billion words) [20]. Once pretrained, the models can be fine-tuned on smaller datasets [19] to provide enhanced performance. Popular embedding techniques such as Word2Vec and GloVe perform "context-free" vector representation of words; for instance, the word "bank" in "bank account" and in "river bank" would have the same representation [19]. To compensate for the problem, bidirectional encoder representations from transformers (BERT) was invented [21] and has been widely popular among NLP researchers in the recent past [22]. BERT contextually represents words and can perform the sentence processing bidirectionally (left to right, and vice versa) [19]. Unfortunately, many advances in NLP do not apply to mine safety narratives, as the language of safety is relatively unique. Mine safety is a niche topic with nuances in the language, and therefore blanket application of generic language models can be quite misleading. Therefore, this research focused on nuances of the language of mine safety.

Ensemble learning methods use multiple algorithms to improve on their individual predictive performances. The random forest (RF) method is an example where results from multiple decision trees are used to obtain the final result [23]. The "stacking" style approach in ensemble methods takes advantage of diverse sets of algorithms to achieve the best result for a given problem. An individual model can have better predictive strength in one area of the dataset but fare poorly on certain other areas. A separate model (stacked model) can be developed for such areas to improve overall predictive performance. Several researchers have applied the stacking methods to traffic-related problems, such as analysis of incidents, accidents, congestion, and so on [24–26].

Random forest (RF) methods are popular among the other classification methods due to their robustness and accuracy [23]. Several researchers have used the methods in a wide variety of areas and industrial contexts, such as construction injury prediction and narrative classification [27,28], flood hazard risk assessment [29], building energy optimization and predictive climate control [30], predicting one-day mortality rate in hospitals [31], and crime count forecasting using Twitter and taxi data [32]. The accuracy and performance of RF methods has been found to be superior when compared to other ML methods by several researchers [28,33]. For instance, in classifying unstructured construction data, Goh and Ubeynarayana, 2017 [28], used and compared the performance of various ML methods such as logistic regression (LR), k-nearest neighbors (k-NN), support vector machines (SVM), naïve Bayes (NB), decision tree (DT), and RF. In finding the factors affecting unsafe behaviors, Goh et al., 2018 [33] used and compared the performance of various ML methods such as k-NN, DT, NB, SVM, LR, and RF. In both the cases, RF showed superior performance. Due to this reason, it was the method of choice for the accident narrative classification done in previous research by the authors in Ganguli et al., 2021 [6]. In the current work, novel models were developed to be "stacked" with RF models (to improve the performance) previously developed in Ganguli et al., 2021 [6].

From the literature review, it can be observed that although the NLP-based ML models are swift in auto-processing huge amounts of text, they have limitations in achieving high success rates. This is due to the diversity in industry-specific safety-related vocabulary. A certain level of human (expert) intervention is required in fine tuning or better training these algorithms. Due to this reason, in the PNNL case, linguistic rules developed by human experts were used for modeling. The linguistic rules developed considered specific phrases and sentence structures common in aviation reports. When used in modeling, these rules were able to automatically identify causes of safety incidents at a level comparable to human experts. The PNNL team, however, noted the reliance of the algorithms on human experts with domain-specific knowledge [4]. When classification challenges occur where traditional or learned approaches fail, application of heuristic methods is not uncommon [34,35]. For instance, in identifying the accident causes in aviation safety reports, Abedin et al., 2010 [34] used a simple heuristic approach in labeling the reports. The approach looks for certain words and phrases that are acquired during the semantic lexicon learning process in the reports. Using ontology as a key component, Sanchez-Pi et al., 2014 [35] developed a heuristic algorithm for automatic detection of accidents from unstructured texts (accident reports) in the oil industry. ASECV draws inspiration from such heuristic approaches.

Hence, the novelty of the research presented in this paper lies in developing models to: (i) resolve the ambiguity in the classification problem, that is, one narrative classified into multiple accident categories (addressed by the SS model); (ii) save process time and memory, and reduce the large size of the training set vocabulary by utilizing industry (mining)-specific expert knowledge and heuristics (addressed by ASECV model); and (iii) improve RF method performance from past research by "stacking" with the best of SS and ASECV models (stacking approach).

2. Previous Research and Importance of This Paper

An important practical application of this research would be in the development of automated safety dashboards at mine sites. In this context, a dashboard is a collection of visual displays of important safety metrics and key performance indicators (KPI) in real time. Currently, the safety dashboards used by mine management simply report on injuries, rather than the causes. This is because of the fact that narratives are not analyzed in a holistic manner, as most mine sites lack manpower that is skilled in analytics. Automatic classification of safety narratives would allow causation (accident type or category "tag") to be included in the dashboards. For example, in addition to listing the number of back sprains, if the dashboard also includes causation tagging, such as "overexertion due to lifting" or "overexertion due to pulling", mine management could deploy more meaningful interventions to improve overall workers' safety. Thus, in the context of mine safety research, the work presented in the paper is novel and very helpful to academic researchers in the area as well.

In the previous research, nine separate RF-based classification models were developed for nine corresponding accident categories; which are explained in detail in Section 3.2. The models were trained (50:50 train-to-test split) based on the narratives in the MSHA database [6]. The trained models were first tested on narratives in the MSHA database, then on narratives in a non-MSHA (partner mine's) database. After the testing, it was found that the MSHA-based RF models were extremely successful in classifying the non-MSHA narratives (96% accuracy across the board). While this is a welcome development, several challenges were encountered. Since nine RF models were deployed separately (standalone), it resulted in certain narratives being classified into multiple accident categories instead of one narrative into one category (which is desirable). According to MSHA criteria, a narrative is generally classified into one category that best describes it. In addition, certain narrow categories, being special cases of (or closely related to) other categories, posed multiple classification problem. For instance, a narrative can be classified as "caught in, under or between a moving and a stationary object" or CIMS, as well as "caught in", since the former is a special case of the latter. Due to their shared vocabulary, certain narratives were also classified into two very different categories, such as "caught in" and "stuck by". To resolve such classification overlaps and tag a narrative with only one accident category, a "similarity score" (SS) approach was used in past research. This method is explained in detail in Section 3.3, in the description of methodology.

In this context, the aim of the current work presented in this paper is to resolve the challenges present in the past research. This is accomplished by turning the SS approach in the previous work into an "SS model" and devising new models to experiment to improve upon the previous success rates of RF models. For the current work, however, non-MSHA narratives were not used; only MSHA narratives were used. Due to the complexity of the problem and the presence of multiple classes in RF-based classification in previous work, the authors aimed to start the experimentation with a sizable portion (with a subcategory such as "caught in"). If it achieved success in this category, the intent was to extend the scope of strategies and methodology to all other categories in future work. Hence, the goal of the current research is to improve the classification performance in the category "caught in", while other categories are considered out of scope.

The concept behind the SS model can be described briefly as follows. Words in each narrative in the modeling or training set were weighted (scored) based on the frequency of their occurrence within an accident category. Thus, the same word is "weighted" differently by different accident categories. From the training set narratives, the SS model builds vocabulary lists for each accident category. Each word or token in a vocabulary list has a set of "weights" corresponding to an accident type. During the application of the model to the test set, each narrative is scored based on the word weights obtained from the training set (accident specific vocabulary weights). This means that a narrative has a score specific to each accident type. Ultimately, a narrative will be assigned to the accident category where it scored the highest when compared to other categories [6]. In this way, the SS method is a unique tool in predicting and assigning a narrative to its most relevant category. Hence, the concept was used in building a new classification model ("SS model") in an attempt to improve the overall success rates of RF models on the MSHA dataset. As opposed to SS approach used in past research—as an additional tool on RF results—the current work uses the SS model as a standalone classification tool.

One challenge for the SS model is that, in order to assign weight scores to words in a given narrative, thousands of narratives need to be processed to form the training set. For instance, there are 4563 narratives that belong to the "caught in" category in the MSHA training set of 40,649 narratives. After text processing and lemmatization, there are 4894 unique words derived out of the 4563 narratives in the category. This paper explores whether a smaller group of words or phrases selected using expert knowledge can be used to detect the "caught in" category or if all of 4894 words need to be used. In this context, a "word-clusters"-based model was developed. It is a heuristic model, using certain expert choice (authors being experts in the mining discipline) vocabulary clusters, and is named as "accident specific expert choice vocabulary" (ASECV). A goal for the model was to keep the false positive rate under 5% to improve its robustness. Finally, as a third (and important) approach, the models (ASECV or SS) were "stacked" on the previous RF models to improve upon the overall success rates. Stacking can be performed in variety of ways. The stacking option chosen for this paper is explained in detail in the methodology outlined in Section 3.5.

Although the RF method alone is not the focus of the current work, it is included in the methodology described in Section 3.2. in order to provide proper background on the past research. As part of the SS and ASECV methods, several experiments (iterations) were conducted to progressively minimize the false positive rates of the models, thus improving the prediction performance. The experimental (iteration) parameters with low false positive rates are ultimately selected for use in stacking method. Thus, it should be understood that the aim of the research is to improve the performance of the RF model (from past research) by the novel approaches, but not solely to compare and contrast the novel models with the RF method.

3. Research Methodology

3.1. MSHA Accident Database

In order to accomplish the research problem, 81,298 narratives from the MSHA accident database, collected for the years 2011 through early 2021, were analyzed [5]. A narrative is typically one to five sentences in length and concisely describes a "reportable" injury that occurred at any of the mines located in the USA (Table 1). The database has 57 fields to describe various attributes of an accident, such as place of accident, mine location, employee age, equipment involved, and so on. There are 45 different accident types in the database that are consistently represented in short sentences or phrases. The sentences "Caught in, under or between a moving and a stationary object" and "Over-exertion in wielding or throwing objects" are some examples.

Table 1. Typical MSHA narrative and its lemmatized form.

MSHA Narrative	Text
Original	"Employee was assisting 3 other miners move Grizzly component in place. While maintaining a vertical position on the component to rehook, the component became unstable and shifted. The employee's effort to maintain it upright failed and it leaned, pinned his elbow against the rib, bending back and breaking left wrist".
Lemmatized form	Assist 3 miner move grizzly component place maintain vertical position component rehook component become unstable shift's effort maintain upright fail lean pin elbow rib bend back break left wrist
Accident type	Caught in, under or between a moving and a stationary object (CIMS)

3.2. Random Forest Classifier

Random forest (RF) methods are popular among the other classification methods due to their robustness and accuracy [23], hence their choice for the past research presented in Ganguli et al., 2021 [6]. It is an ensemble method that uses a set of decision trees in order to classify narratives into their appropriate accident categories. During the "training" process, the RF method learns from the training set (50% of 40,649 narratives total used in this paper) and builds decision trees based on each column ("feature") of the data set. Then, the fitted decision tree model is applied to the narratives in the "test set" (the other 50% of narratives) to predict their respective accident categories. The method is described more in Ganguli et al., 2021 [6] and Mitchell, 1997 [36]. The 50:50 split used in the paper is consistent with the past research.

In order to train the RF model, the following steps are performed in advance. Each narrative is at first converted to lower case and tokenized (split into unique words or unigram tokens). In the next step, certain common words that do not add much value to modeling, called "stop words" (in, the, between, employee, EE, etc.), along with certain symbols (", &, /, etc.) and spaces are removed. Subsequently, the words are lemmatized or reduced to their root forms, for example, "pinched" and "pinching" will become "pinch". Word vectorization is the next step to represent the words in number forms. After processing the narratives in the above steps, the accident classification model was implemented using RandomForestClassifier () in the scikit-learn [37] toolkit. Table 2 shows the major (in terms of counts) accident groups that were used in the RF analysis. Some narrow group accident categories are abbreviated as given below. It should also be noted that the acronym "NEC" stands for "not elsewhere classified". The reason for undertaking the narrow groups (as opposed to major) is to study how the RF model performs on such small groups. Narrow groups are much harder to classify due to the small amount of narratives available for training purpose and the common vocabulary they share with the major groups.

- Over-exertion in lifting objects (OEL).
- Over-exertion in pulling or pushing objects (OEP).
- Fall to the walkway or working surface (FWW).
- Caught in, under or between a moving and a stationary object (CIMS), and
- Struck by flying object (SFO).

Table 2. The four "accident type" groups modeled in the previous paper (Ganguli et al., 2021 [6]).

Type Group: Caught in	Type Group: Fall	Type Group: Over-Exertion	Type Group: Struck
Caught in, under, or between a moving and a stationary object	Fall down raise, shaft or manway	Over-exertion in lifting objects	Struck by concussion
Caught in, under, or between collapsing material or buildings	Fall down stairs	Over-exertion in pulling or pushing objects	Struck by falling object
Caught in, under, or between NEC	Fall from headframe, derrick, or tower	Over-exertion in wielding or throwing objects	Struck by flying object
Caught in, under, or between running or meshing objects	Fall from ladders	Over-exertion NEC	Struck by powered moving object
Caught in, under, or between two or more moving objects	Fall from machine	-	Struck by rolling or sliding object
-	Fall from piled material	-	Struck by... NEC
-	Fall from scaffolds, walkways, platforms	-	-
-	Fall on same level, NEC	-	-
-	Fall onto or against objects	-	-
-	Fall to lower level, NEC	-	-
-	Fall to the walkway or working surface	-	-

Table 3 shows how the training and test sets have been split among different accident categories for RF-based analysis in the past research.

Table 3. Various accident categories in the training and testing subsets. Each subset has 40,649 samples (Ganguli et al., 2021 [6]).

Subset	Type Group: OE	Type Group: Caught in	Type Group: Struck by	Type Group: Fall	OEP	OEL	FWW	CIMS	SFO
Training	8909	4563	10,216	4802	1290	2838	2130	3337	1586
Testing	8979	4524	10,226	4926	1275	2961	2130	3310	1590

Table 4 provides a summary of the modeling results within the MSHA test set. In order to understand the table, the over-excretion (OE) type group is presented as an example. The following formulae are used to arrive at the various results in the table.

- Total samples (n_samples): 40,649;
- Total samples in target category (n_target): 8979;
- Total samples in other categories (n_other): n_samples − n_target = 31,670;
- Samples from target category predicted accurately (n_target_accurate): 7248;
- Samples from “other category” predicted wrongly as target (false_predicts): 1331;
- Samples from “other category” predicted correctly as other (other_accurate): 31,670 − 1331 = 30,339;
- Percentage of targets accurately predicted: 100 × n_target_accurate/n_target = 100 × 7248/8979 = 81%;
- False positive rate: false_predicts/n_other = 1331/31,670 = 4%;
- Total correct predictions (total_correct): n_target_accurate + other_accurate = 7248 + 30,339 = 37,587;
- Overall success rate (%) = 100 × total_correct/_samples = 100 × 37,587/40,649 = 92%.

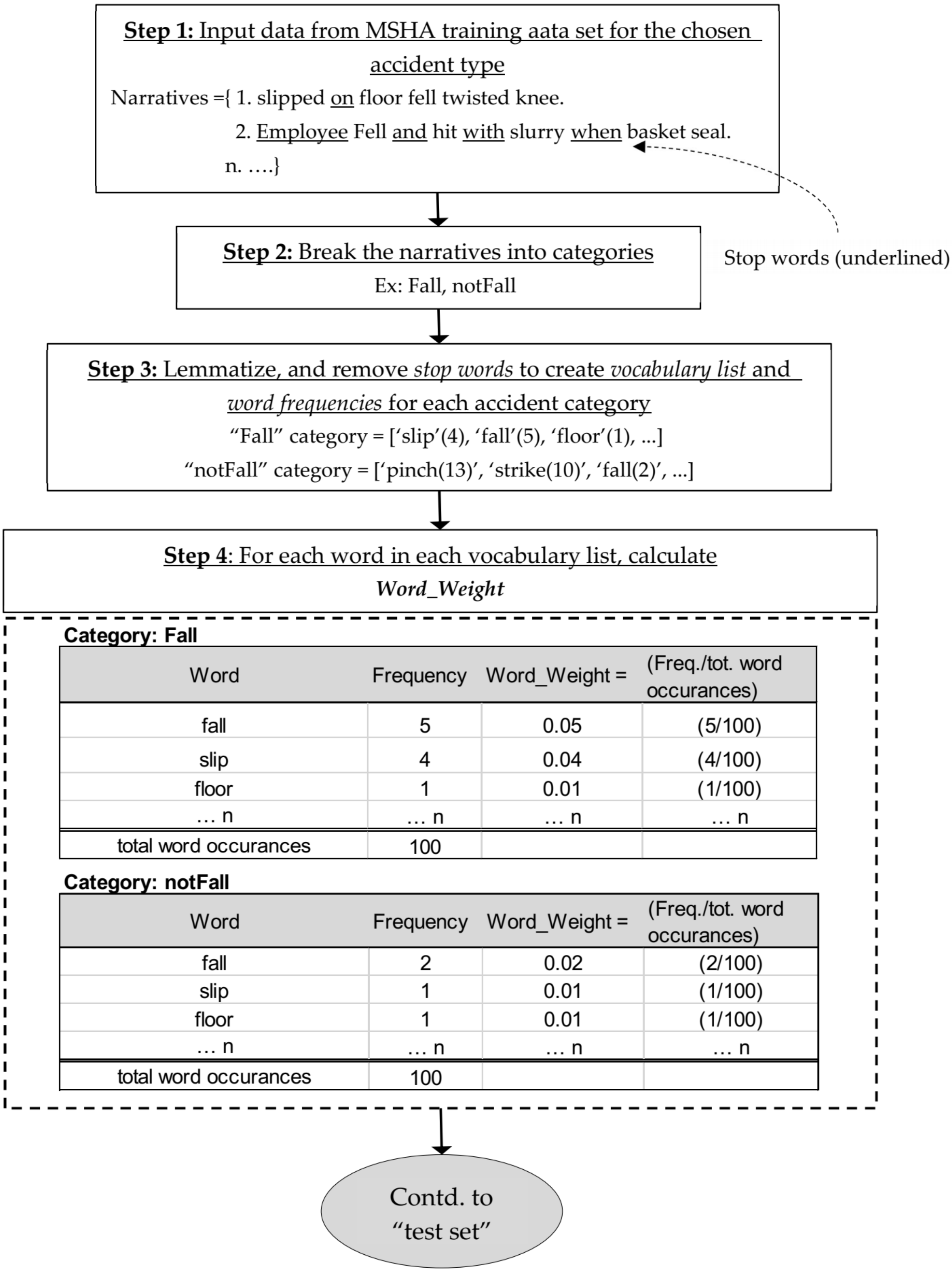
Table 4. Results from RF model-based analysis on the MSHA test set (Ganguli et al., 2021 [6]).

Metrics	Type Group: OE	Type Group: Caught in	Type Group: Struck by	Type Group: Fall	OEP	OEL	FWW	CIMS	SFO
Records from Category	8979	4524	10,226	4926	1275	2961	2130	3310	1590
Overall Success	92%	96%	90%	95%	98%	96%	96%	95%	97%
% from Category Accurately Predicted	**81%**	**71%**	**75%**	**71%**	**37%**	**59%**	**34%**	**55%**	**25%**
False Positive	**4%**	**1%**	**5%**	**2%**	**<1%**	**<1%**	**<1%**	**2%**	**<1%**

3.3. Similarity Score (SS) Model

As stated earlier, the main aim of the SS model is to improve upon the success rates and false positive rates achieved by the RF method on the MSHA database, shown as “% from Category Accurately Predicted” and “False Positives”, respectively, in highlighted text (Table 4). Hence, the SS model uses the same set of narratives collected for the RF model. The model flowchart in Figure 1 shows the steps involved in processing the training set of narratives. Like the RF model, the SS model uses a 50:50 split between training and test sets. As a first step (Step 1 in Figure 1), the algorithm selects narratives from the chosen accident type (such as “fall”) from the total set of narratives (81,298). In Step 2, it separates narratives that do not belong to the chosen accident type into the “notFall” category. Note that the first two steps can be repeated when several accident categories are involved. In Step 3, narratives from the “fall” and “notFall” categories are lemmatized. At this stage, each accident category is left with a unique set of words or tokens (vocabulary sets) with their corresponding frequency (number of occurrences) scores. Thus, a vocabulary list created for each accident category (in this case, Fall and notFall) is unique in its length and constituent words. In a similar manner, the training set can be divided into several accident types, such as “caught in”, “struck by”, CIMS, and so on, and ultimately can be reduced to their respective unique “vocabulary sets”. The whole process can be described as “word vectorization”.

In Step 4, each unique word in a vocabulary set will be given a weight, calculated by dividing the number of times that word occurred in the set (frequency) by the sum of all words’ occurrences. In this manner, each word in a vocabulary list belonging to an accident category has its own unique “weight”, depending on its frequency in that list. For instance, the word “fall” can occur in different accident type vocabulary lists. However, it will most likely carry additional weight in the “fall” type accident list, since its frequency of occurrence can be high in such a list. The numbers (word frequencies, weights, etc.) used in Step 4 of the flowchart are for demonstration purposes only.


Step 1: Input data from MSHA training aata set for the chosen accident type
Narratives ={ 1. slipped on floor fell twisted knee.
2. Employee Fell and hit with slurry when basket seal.
n. ….}
Stop words (underlined)
Step 2: Break the narratives into categories
Ex: Fall, notFall
Step 3: Lemmatize, and remove *stop words* to create *vocabulary list* and *word frequencies* for each accident category
"Fall" category = ['slip'(4), 'fall'(5), 'floor'(1), ...]
"notFall" category = ['pinch(13)', 'strike(10)', 'fall(2)', ...]
Step 4: For each word in each vocabulary list, calculate *Word_Weight*
Category: Fall
Category: notFall
Contd. to "test set"

Category: Fall

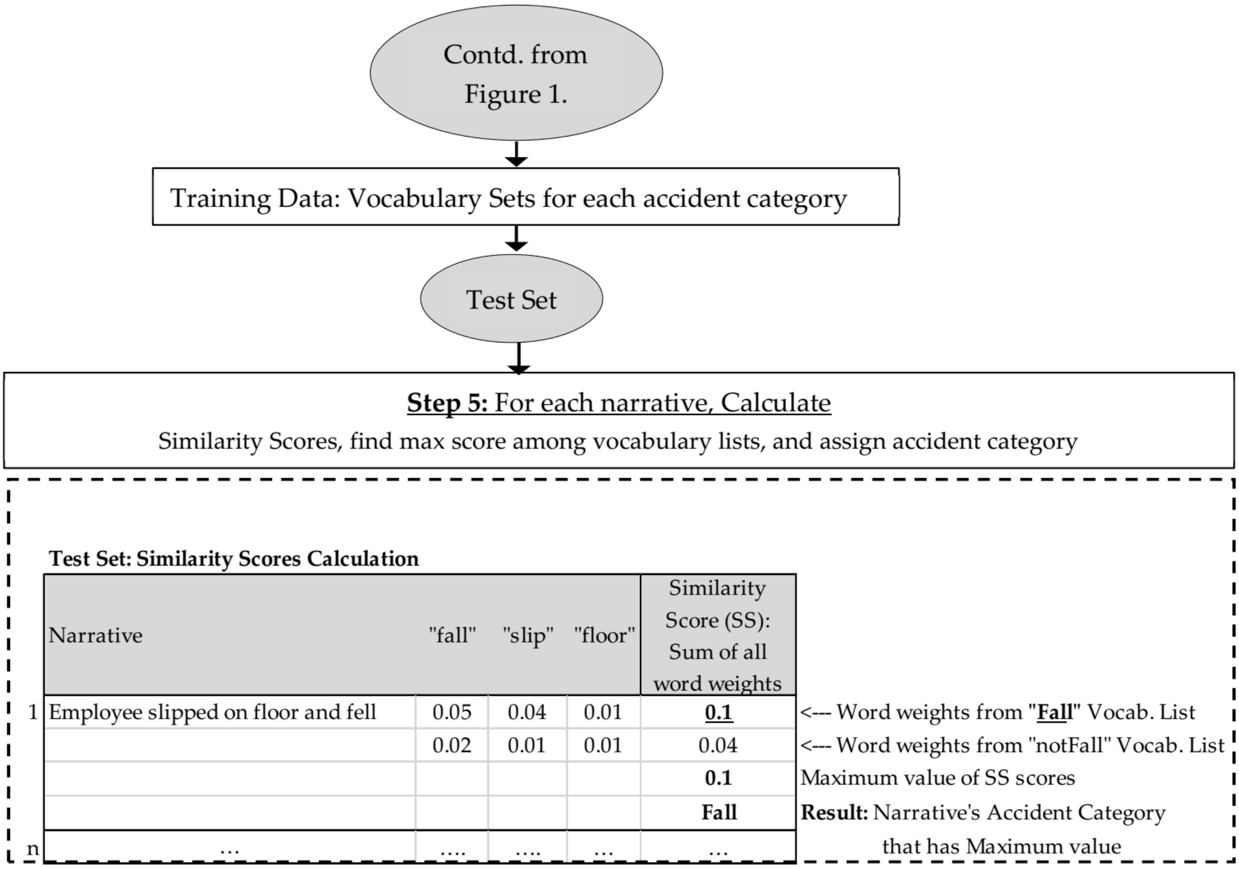
Word	Frequency	Word_Weight =	(Freq./tot. word occurances)
fall	5	0.05	(5/100)
slip	4	0.04	(4/100)
floor	1	0.01	(1/100)
… n	… n	… n	… n
total word occurances	100		

Category: notFall

Word	Frequency	Word_Weight =	(Freq./tot. word occurances)
fall	2	0.02	(2/100)
slip	1	0.01	(1/100)
floor	1	0.01	(1/100)
… n	… n	… n	… n
total word occurances	100		

Figure 1. Flowchart for "training" the similarity score (SS) model.

The accident-specific vocabulary sets and their word weights created in the training stage thus far will be used to calculate the similarity score for each test set narrative (Figure 2). The narratives (40,649) from the test set will be processed through the rest of the algorithm shown in the Figure 2 flowchart. In Step 5, each test set narrative is reduced to its lemmatized word list. For example, the narrative “Employee slipped on floor and fell” is reduced to the words “fall”, “slip”, and “floor”. Then, from the “Fall” accident category vocabulary list created during training process, weights are assigned to each of these three words. Likewise, the process is repeated for the “notFall” category.


Contd. from Figure 1.
Training Data: Vocabulary Sets for each accident category
Test Set
Step 5: For each narrative, Calculate
Similarity Scores, find max score among vocabulary lists, and assign accident category

Test Set: Similarity Scores Calculation

	Narrative	"fall"	"slip"	"floor"	Similarity Score (SS): Sum of all word weights	
1	Employee slipped on floor and fell	0.05	0.04	0.01	**0.1**	<--- Word weights from "**Fall**" Vocab. List
		0.02	0.01	0.01	0.04	<--- Word weights from "notFall" Vocab. List
					0.1	Maximum value of SS scores
					Fall	**Result:** Narrative's Accident Category that has Maximum value
n	…	….	….	…	…	

Figure 2. Flowchart for test data processing with SS model.

Based on the algorithms (Figures 1 and 2), Table 5 demonstrates how an accident category is assigned to a test narrative from the MSHA database. For instance, the test narrative number 1 is scored according to different accident-specific vocabulary lists (accident categories OE through SFO). The narrative’s category is assigned as CIMS, since the category scored maximum among the others (highlighted text under accident class). On the other hand, the narrative’s actual category from the MSHA database is “Stuck by”, making the assignment a false positive. The “SS” model performance can be found in the results section.

Table 5. Similarity scores criteria to classify “accident category” of a narrative (lemmatized) from the test set.

	Accident Category and Similarity Scores													
	OE	**OEP**	**OEL**	**Struck by**	**Caught in**	**Fall**	**FWW**	**notFWW**	**CIMS**	**SFO**	**Max Value**	**Accident Class**	**Actual Class**	**False Positive?**
1	**Narrative:** use mill bar clean chute bring bar back strike’s hand cross member bar cause laceration’s right hand receive six suture close wound													
	0.067	0.076	0.064	0.120	0.127	0.081	0.078	0.090	**0.130**	0.088	0.130	**CIMS**	Struck by	Yes (1)
2	**Narrative:** work unit # 1238 air conditioner step cab foot slip off step catch right hand later day say pain shoulder work 7/12/13													
	0.127	0.110	0.103	0.066	0.083	0.127	0.128	0.069	0.081	0.049	0.128	FWW		
3	**Narrative:** drill last bolt hole cut undetected slick sided rock slip hit top canopy break one piece rock strike left side back result contusion 1.5” long laceration stitch require													
	0.067	0.068	0.069	0.165	0.099	0.101	0.096	0.108	0.101	0.126	0.165	Struck by		

3.4. Accident Specific Expert Choice Vocabulary (ASECV) Model

The ASECV model is essentially a heuristic approach that compensates for some of the challenges encountered by SS model. For instance, SS approach suffers from high false positive rates (which will be further discussed in Section 4.1). This is due to the fact that its accident specific vocabulary lists have too many common words, raising ambiguity in classification. Unfortunately, the false positive rates did not improve even when vocabulary lists were repopulated with words exclusive to a particular accident type. When such classification challenges occur where traditional or learned approaches fail, application of heuristic methods is not uncommon [34,35]. ASECV draws inspiration from such heuristic approaches.

In the same context, use of experts familiar with safety vocabulary, specific to an industry, in selecting certain phrases for the purpose of training the accident classification tools is not uncommon [4,12]. ASECV also uses such a strategy in identifying key phrases that are important for the classification of "caught in" type accidents. For the purpose, authors, being the experts in the mining industry, will identify such key phrases. For instance, when a "caught in" type accident occurs, the majority of the time, the narrative contains the words "between" or "under" or "in", followed by some object names. In other words, looking for such prepositions after the word "catch" can help identify "caught in" type narratives. Likewise, depending on the narrator's style—which can be observed through random examination of narratives from the training set—verbs that best describe key actions in an accident type can be deduced. This apparently saves lot of computer memory, since a typical training vocabulary set involves thousands of frequently occurring words.

At first, the ASECV model processes narratives from the training set to form the word clusters; then, the clusters will be utilized to score and classify test set narratives into the targeted ("caught in") category. In conformance with the previous models, a 50:50 training-to-test-set data split is used.

3.4.1. Training the Model

One important aspect of the ASECV model that differs from the rest of the models is that, unlike in the RF and SS models, the stop words are not removed when lemmatization of a narratives is performed. In order to form the word clusters, the algorithm follows certain heuristic rules (or steps) set forth by the industry experts as given below.

(i). Tokenization and sorting: At first, the training set narratives from the "caught in" category are lemmatized and converted into tokens, and then sorted based on their frequency of occurrence.

(ii). Expert choice words: From the first 100–200 tokens of the sorted list, certain key words are selected by experts. This is done heuristically by observing the link between the word's occurrence in a narrative and the chance (probability) that the narrative is categorized into the "caught in" category.

(iii). Word clusters: Then, the words are arranged into clusters of importance. For instance, when words such as squeeze, crush, pinch occurred, it was observed that there is a high probability (95–100%) that the narrative belongs to the "caught in" category. Hence, these words are sorted into a high-importance cluster (1). Likewise, there is a medium-importance word cluster (2) and a low or complementary-importance word cluster (3). All the words that do not belong to word clusters will be given a default score of zero.

(iv). Word weights in clusters (high importance)*:* Each word in a cluster is given a "score weight". A score weight of 100 is given to high-importance words. Although the scoring choice is arbitrary, it roughly corresponds to the chance (95–100%) that a narrative will be classified into the "caught in" category when such high-importance words occur. For the same reason, the qualifying score is set at 100 for test set narratives.

(v). Word weights in clusters (medium and low or complementary importance): Since the high-importance words are set at score 100, others are scaled down to 80, 60, and 20, approximately reflecting the probability that the narrative fits into the intended category.

(vi). Word weights in clusters (other rules): The prepositions, such as "between", "under", and so on, played an important role in identifying the "caught in" category. For instance, the word "catch" (score: 80) followed by "between" (score 20) within 5 word spaces can qualify (total score: 100) for the "caught in" category.

3.4.2. Testing the Model

During the testing process of the algorithm, the following steps take place, which are depicted in the flowchart in Figure 3.

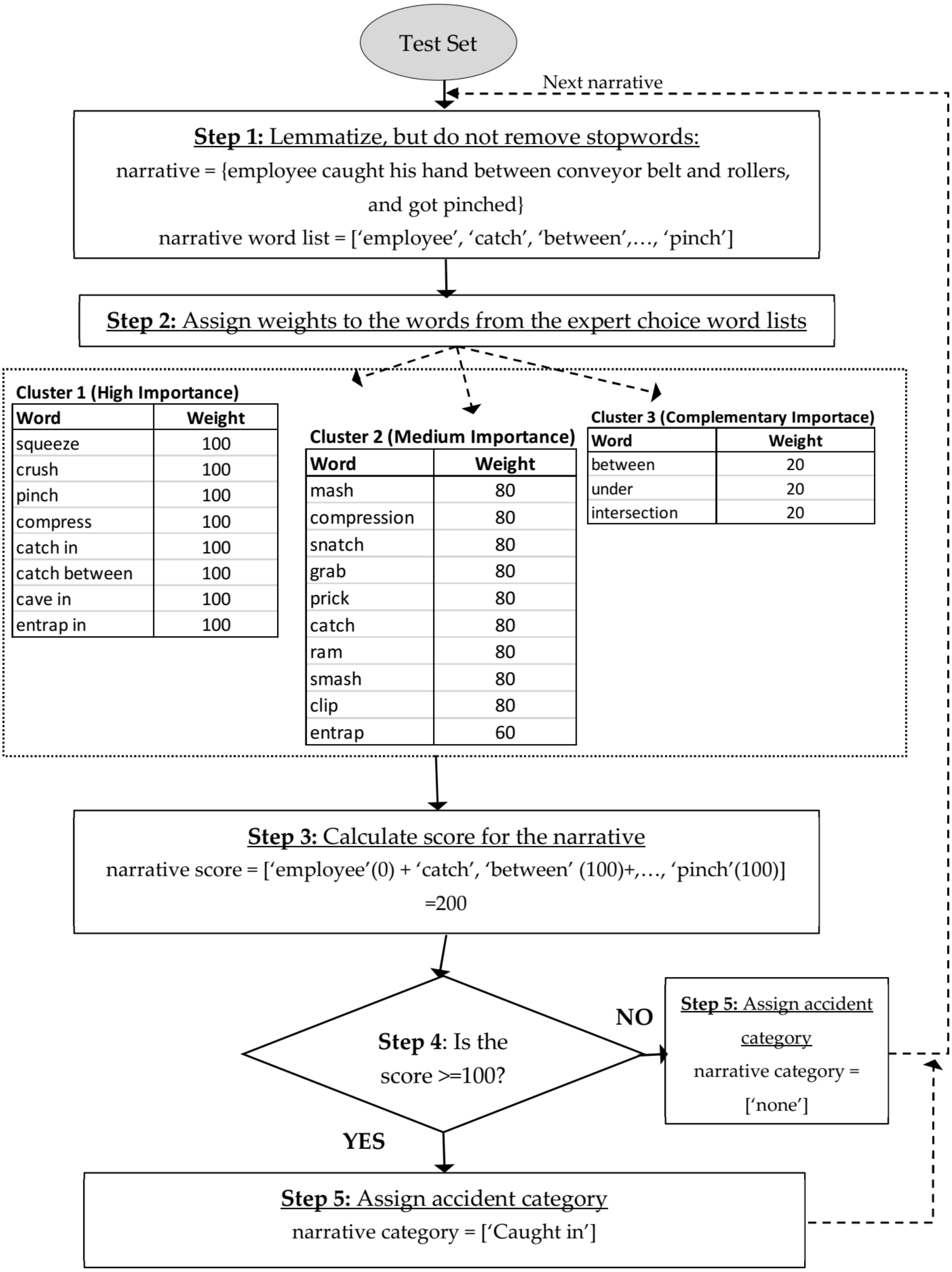
(i). A narrative from a test set is broken into its token list (Step 1).

(ii). Then, the algorithm looks for words matching tokens from the clusters and assigns corresponding weights to tokens (Step 2). As part of Step 2, duplicate tokens will be eliminated to avoid scoring the same token multiple times.

(iii). Heuristic rules are applied at this stage to calculate the score of a narrative (Step 3). The sum of individual token weights in a narrative is the score of the narrative.

(iv). Qualifying score of narrative: As a rule, if the score of a narrative exceeds or equals to a qualifying score (100), the narrative can be classified into the intended accident category (Step 4).

Once the word clusters are compiled from the training process, they were tested to find the impact of the presence (or absence) of each word in a cluster on the overall performance of the model. By heuristically (and systematically) adding and dropping words to the clusters, the performance of the ASECV algorithm was monitored on test sets in terms of accuracy and false positive rates. Word clusters with the best model performance are retained for the purpose of the "stacking approach." These experiments (iterations) are presented in detail in Section 4.


Test Set
Next narrative
Step 1: Lemmatize, but do not remove stopwords:
narrative = {employee caught his hand between conveyor belt and rollers, and got pinched}
narrative word list = ['employee', 'catch', 'between',..., 'pinch']
Step 2: Assign weights to the words from the expert choice word lists
Step 3: Calculate score for the narrative
narrative score = ['employee'(0) + 'catch', 'between' (100)+,..., 'pinch'(100)] =200
Step 4: Is the score >=100?
NO
YES
Step 5: Assign accident category
narrative category = ['none']
Step 5: Assign accident category
narrative category = ['Caught in']

Cluster 1 (High Importance)

Word	Weight
squeeze	100
crush	100
pinch	100
compress	100
catch in	100
catch between	100
cave in	100
entrap in	100

Cluster 2 (Medium Importance)

Word	Weight
mash	80
compression	80
snatch	80
grab	80
prick	80
catch	80
ram	80
smash	80
clip	80
entrap	60

Cluster 3 (Complementary Importace)

Word	Weight
between	20
under	20
intersection	20

Figure 3. Flowchart for ASECV Model.

3.5. Stacking Approach

The stacking option chosen for the paper can be explained as follows (Figure 4). If the RF model from past research classifies a test narrative in to "caught in" category, the narrative is assigned with a value of "1". If any of the new models (such as "SS" or

"ASECV") have comparable performance with RF (1% false positive rate), we can stack the model to RF. This means, as shown in the flowchart (Figure 4), that after processing test set narratives (Step 1) with RF, all the narratives that are not classified by RF (0 s) will be tried as a test set by the new model (Step 2). Any success achieved in "accurate" classification of the narratives by the new model will be added to the overall success of the RF model (Step 3). Thus, the stacked performance of the RF and the new model together can be superior to RF performance alone.

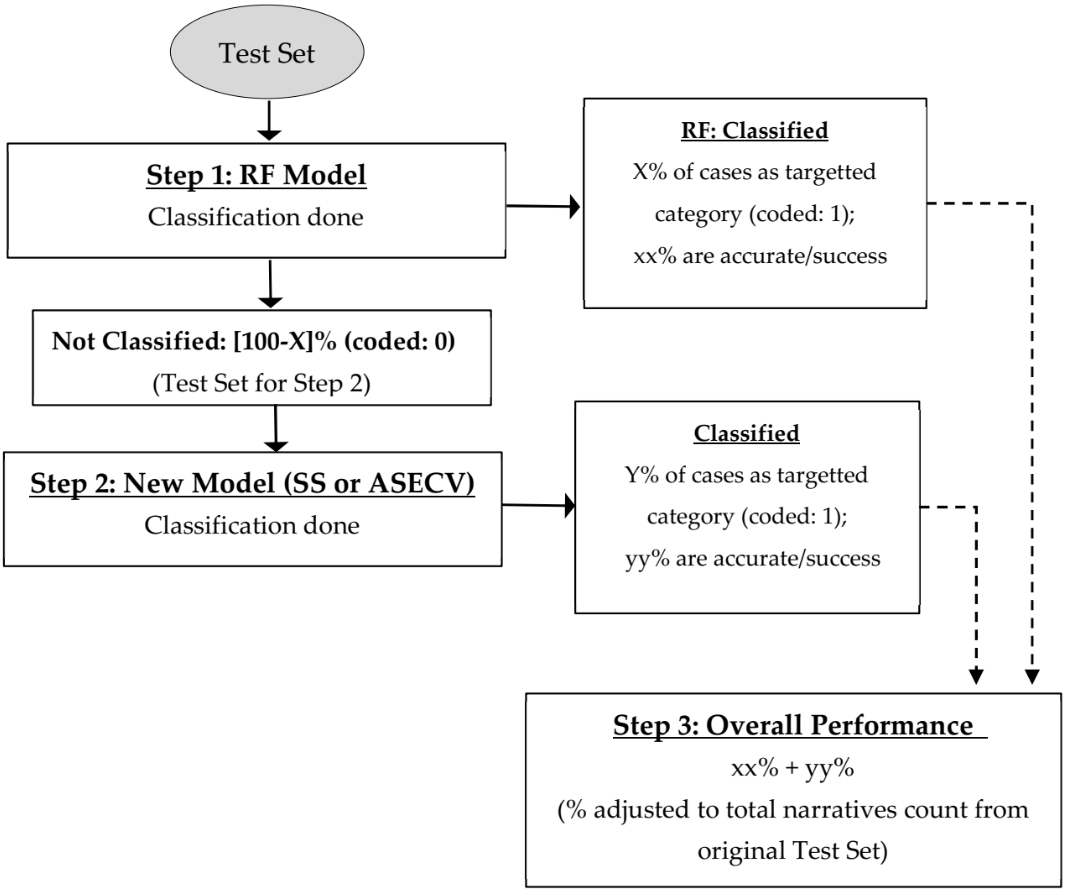

Test Set
Step 1: RF Model
Classification done
RF: Classified
X% of cases as targetted category (coded: 1);
xx% are accurate/success
Not Classified: [100-X]% (coded: 0)
(Test Set for Step 2)
Step 2: New Model (SS or ASECV)
Classification done
Classified
Y% of cases as targetted category (coded: 1);
yy% are accurate/success
Step 3: Overall Performance
xx% + yy%
(% adjusted to total narratives count from original Test Set)

Figure 4. Flowchart for stacking model.

4. Results

The results from each of the three approaches and their corresponding experiments are described in the following sections. The aim of the experiments was to find high-performance model parameters between the SS and ASECV methods so that the parameters can be used when the stacking approach is deployed.

4.1. Results: SS Model

The "vocabulary strategy" column of Table 6 shows the type of strategy followed for SS model training exercises to create the accident-specific "vocabulary sets". As stated in the previous sections, these vocabulary sets will be used to score and classify test set narratives. The "SS Criteria" column provides targeted accident categories for narrative classification. In order to compare with RF results from past work, the CIMS accident category was chosen. The aim is to find the combination of vocabulary strategy and criteria at which the SS model provides the best performance in terms of high "Success within (accident) category" with low "false positive rates". The aspiration is that if the results from SS modeling for the CIMS category outperform the RF model, the criteria can be extended (and applied) to other accident categories to get the similar results. It should be noted that all the results presented in this section are obtained by applying the strategies on test set narratives. The following are some dataset parameters used.

- Total "test set" samples (*n*_samples) used = 40,649;
- Total samples in target category (*n*_target) = 3310;
- Total samples in other categories (*n*_other) =37,339.

Table 6. RF vs. SS: Classification performance for CIMS type accidents.

			Target Category Predicted Accurately (*n*_target_accurate)		**Success within Category: *n*_target_accurate/*n*_target**		**False Positive Rate: false_predicts/*n*_other**	
	Vocabulary Strategy	**SS Criteria**	**RF**	**SS**	**RF**	**SS**	**RF**	**SS**
1	All words	Multiple Catg. (9) Splitting	1815	290	55%	8.8%	2%	15%
2	All words	Few Catg. (FWW/CIMS/SFO/Other) Splitting	1815	170	55%	5.1%	2%	12%
3	Excl. words (100)	Few Catg. (FWW/CIMS/SFO) Splitting	1815	3005	55%	90.8%	2%	24%
4	Excl. words + 25freq. Words	CIMS, notCIMS, Neither Class	1815	3305	55%	99.8%	2%	96%
5	Excl. words + 25freq.+ Adj. Wts.	CIMS, notCIMS, Neither Class	1815	3208	55%	96.9%	2%	58%
6	Excl. words + Diff. qualify strategy	CIMS, notCIMS, Neither Class	1815	159	55%	4.8%	2%	58%

At first, all words in their respective vocabulary sets for nine accident categories were tried in narrative classification (Strategy 1). The success rate was poor (8.8%) compared to RF (55%), and the false positive rate was high, at 15%, when compared to RF's 2%. In strategy 2, the number of accident categories was minimized (FWW, CIMS, SFO, and other) to observe if it can help improve the performance. The "other" category includes all narratives in the training set outside of the three categories noted. FWW, CIMS, and SFO are very narrow categories, as opposed to the rest of the categories such as "caught in" or "struck by" demonstrated in the RF model (Table 2). The success and false positive rates for Strategy 2 went slightly down to 5.1% and 12%, respectively, when compared to Strategy 1.

Even though common words such as "employee" and "EE" were eliminated as part of the stop words, it was observed that all accident categories share vocabulary to a certain extent. For instance, the tokens "pain", "hurt", and "fall" are common in each accident type. This could be one reason that Strategies 1 and 2 registered such low success rates to begin with. This prompted a change of strategy to eliminate such common vocabulary among all accident categories. Therefore, Strategies 3–7 include only "exclusive" vocabulary sets for each accident category. With the new strategy, it was hoped that the SS score of a narrative would be high for a particular accident type when its exclusive vocabulary was present in the narrative. As an experiment, for Strategy 3, the top 100 most frequent words of the vocabulary list were used. The success rates for CIMS improved significantly (90.8%), but with increased false positive rates (24%), which is not desirable.

At this stage, it can also be noticed that the CIMS category performance depended upon how well the narratives are split between several other categories. Similar to the RF models developed in the past research, one model to classify one category avoids such dependencies. Moreover, there will only be two major vocabulary sets at any given point of time, which is less complex for modeling. A CIMS category classification model, for instance, will have two vocabulary sets from training, that is, all vocabulary that belongs to CIMS and the rest of the vocabulary ("notCIMS"). In the testing process, when the SS scores of a narrative for CIMS and notCIMS become equal, it sometimes creates ambiguity for the algorithm. In such cases, the narrative will be classified into "Neither Class", which is counted in the notCIMS category for calculation purposes. For Strategies 4 and 5, all exclusive words along with the 25 most frequent "common" words were used. Although the success rates were high (>95%), the false positive rates were high as well (96% and 58%, respectively), which is not desirable. The high false positive rates are due to the presence of words, patterns, or certain elements from multiple accident categories in one narrative. For instance, the narrative "employee fell and caught his hand between the moving conveyor belt and stationary guard” can be classified by MSHA as a “Fall” accident. However, models or algorithms can interpret and classify the same narrative in a few different ways. For instance, due to the presence of words “caught” and “in between”, it can be classified in the “caught in” category or the “caught in between moving or stationary objects” category due to the presence of objects such as “conveyor belt”, a moving object, and a “guard”, a stationary object.

Contrary to other strategies, it should be noted that for Strategy 5, the weights of words in vocabulary lists are adjusted proportionately to the 25 most frequent words. For Strategy 6, all exclusive words were used along with a “difference” strategy. The strategy can be explained as follows. For the training set, it is possible that the difference between the CIMS and notCIMS scores for the narratives can have a correlation with accurate prediction (CIMS) rates. In particular, a score above the 95th percentile of the “difference” was observed to be highly correlated with correct prediction of the accident type. This is used as a “qualifying criteria” for Strategy 6. In the test set, whenever the difference between the scores for a narrative “qualifies”, it will be automatically categorized as CIMS. The strategy, however, has limited success (4.8%), with high false positive rates (58%).

Overall, the strategies yielded mixed results, with success rates varying between 4.8% to 96.9%, and false positive rates between 12% to 96%. Since the false positive rates are higher than the desired levels (<5%), the SS model is not considered suitable for stacking over the RF model.

From the SS model results, it is observed that improving the success rates and false positive rates for the CIMS category in comparison to RF is difficult. This is because CIMS is a very narrow category of the broad “Caught in” group of accidents, and it shares lot of vocabulary with the group. Hence, for the experiments (iterations) in the ASECV model, the “Caught in” category was chosen. The aspiration is to build a successful model with strategies that can be applied to other models at later stage.

4.2. Results: ASECV Model

Table 7 shows the results for various ASECV vocabulary criteria in the “caught in” accident category. The clusters of vocabulary lists created from the training set are provided in Table 7 as well. The clusters are classified according to the importance of the words they contain with respect to narrative classification. For instance, high, medium, and low or complementary type clusters have word weights of 100, 80, and 20, respectively, except for “entrap”, which has been given a weight score of 60.

Table 7. RF vs. ASECV: "caught in" category performance for various criteria.

		Vocabulary Sets by Importance			**Target Category Predicted Accurately (*n*_target_accurate)**		**Success within Category: *n*_target_accurate/*n*_target**		**False Positive Rate: false_predicts/*n*_other**	
		High Score: "100"	**Medium Score: "80", Except 'entrap': "60"**	**Complementary/Low Score: "20"**						
	Accident Category	**Cluster 1**	**Cluster 2**	**Cluster 3**	**RF**	**ASECV**	**RF**	**ASECV**	**RF**	**ASECV**
1	Caught in	squeeze, crush, pinch, compress, catch in, cave in, entrap in	clip, ram, smash, mash	between, under, intersection	3212	2941	71%	65%	1%	1.76%
2	Caught in	squeeze, crush, pinch, compress, catch in, cave in, entrap in	clip, ram, smash, mash, **entrap**	between, under, intersection	3212	2036	71%	45%	1%	1.19%
3	Caught in	squeeze, crush, pinch, compress	clip, ram, smash, mash		3212	1629	71%	36%	1%	0.85%
4	Caught in	squeeze, crush, pinch, compress, **clip, ram, smash, mash, catch, entrap**			3212	1699	71%	38%	1%	1.02%
5	Caught in	squeeze, crush, pinch, compress, **clip, ram, smash, mash, catch, entrap, catch in, cave in, entrap in.**			3212	3480	71%	77%	1%	6.21%
6	Caught in	squeeze, crush, pinch, compress	clip, ram, smash, mash		3212	1614	71%	36%	1%	0.80%
7	Caught in	squeeze, crush, pinch, compress, clip, ram, smash, mash			3212	2294	71%	49%	1%	1.84%
8	Caught in	squeeze, crush, pinch, compress			3212	1606	71%	35%	1%	0.79%
9	Caught in	squeeze, crush, pinch			3212	1602	71%	35%	1%	0.76%
10	Caught in	squeeze, crush			3212	166	71%	0.04%	1%	0.31%
11	Caught in (**ASECV * performance when stacked on RF**)	squeeze, crush, pinch			3212	103	71%	**8%**	1%	**0.42%**

* The model in the iteration uses 36,953 narratives that were not classified by RF model as test set, as opposed to the regular 40,649 narratives test set used by all other iterations. Hence, the calculations are proportional to the test set length.

The experiments or iterations (1–11) involve the adding and dropping of words to the clusters. This is to observe how different word combinations affect the success rates, as well as false positive rates, when applied to test set narratives. When new words are added to the clusters for an iteration, they were highlighted in the text (Table 7). The process of adding or dropping words is first done systematically, by sorting the words according to their importance (using scores), and then by dropping one word at a time from the bottom of the list. Later, several combinations of words from each cluster are performed both systematically and heuristically based on the expert knowledge. The 11 iterations presented are the result of exploring many permutations and combinations. The success and false positive rates of the ASECV model are compared to the RF model side by side in the table.

For iteration 1, the full extent of the words list for each cluster was used. The success rate was 65%, with a false positive rate of 1.76%, which are low numbers when compared to the RF model rates of 71% and 1%, respectively. The overall focus is to reduce the ASECV model's false positive rates to below 1%, which is comparable to RF performance in the same area (1%). This way, when ASECV is stacked over RF, there is minimal compromise in accuracy. In this context, the ASECV model outperforming the RF model is desirable. For iteration 2, the word "entrap" was added to cluster 2. The success rate dramatically reduced to 45%, while the false positive rate was slightly reduced to 1.19%. Removing the words (or prepositions) from cluster 3 ("between", "under" and "intersection") altogether in iteration 3 resulted in a lowering of the success rate to 36%. However, the false positive rate was brought down to 0.85%. It is interesting to see in iteration 4 that moving cluster 2 (clip, ram, smash, mash) and the word "entrap" to cluster 1 improved the success rate to 38%. The false positive rate, however, was only slightly increased (1.02%) from iteration 3. In iteration 5, certain phrases such as "catch in", "cave in", "entrap in" are added. The resulting success rates improved to 77%, which is better than the RF model, but the false positive rates stayed higher, at 6.21%, when compared to RF's 1%. Iterations 6–9 demonstrated that the false positive rates can be reduced below 1% by gradually eliminating words to ultimately keep the words "squeeze", "crush", and "pinch" in cluster 1. The success rate at this point was 35%, but the false positive rate was reduced to 0.76%, which is highly desirable for the stacking approach. Dropping the word "pinch" in iteration 10 drastically reduced the success rate to 0.04%. This shows how important the word is to the model. In this context, the vocabulary set in iteration 9 is selected for stacking the RF model due to its low false positive rate.

Ultimately, in iteration 11, the narratives that the RF model failed to classify (assigned as "0") into the "caught in" category (36,953 out of 40,649) were analyzed by the ASECV model. Hence, as a stacked model on the RF model, ASECV alone achieved an 8% success rate, with only 0.42% false positive rate, which is highly desirable (see highlighted text in iteration 11). The following are some dataset parameters used.

- Total "test set" samples (*n*_samples) used = 40,649;
- Total samples in target category (*n*_target) = 4524 (Caught in);
- Total samples in other categories (*n*_other) = 37,339 (Caught in).

4.3. Results: Stacking Approach

Out of all the experimentations performed for the SS and ASECV models, iteration 11 of the ASECV model looked promising. Hence, it was used in stacking with the RF model from past research. The "stacking" approach has resulted in improving the overall success rate of the RF model from 71% to 73.28%, with only 1.41% false positive rate (Table 8). Hence, it can be noticed that the "stacked model" has resulted in improving the existing success rates of the RF model.

Table 8. Overall performance of the stacked model (RF when stacked with ASECV).

	Vocabulary Sets by Importance			**Target Category Predicted Accurately (*n*_target_accurate)**		**Success within Category: *n*_target_accurate/*n*_target**		**False Positive Rate: false_predicts/*n*_other**	
	High Score: “100”	**Medium Score: “80”, Except ‘entrap’: “60”**	**Complementary/Low Score: “20”**						
Accident Category	**Cluster 1**	**Cluster 2**	**Cluster 3**	**RF**	**Stacked**	**RF**	**Stacked**	**RF**	**Stacked**
Caught in	squeeze, crush, pinch			3212	3315 (3212 RF + 103 ASECV)	71%	**73.28%** (71% RF + 8% ASECV)	1%	**1.41%**

5. Discussion

When RF model performance is compared with that of the SS model (Table 6), certain interesting results can be observed. The success (5.1–99.8%) and false positive rates (1–96%) were poor in general and fluctuated in a wide range. In addition, high success rates were not always accompanied by low false positive rates. When narratives were classified into all nine (9) categories using all the vocabulary of each accident-related set (iterations 1 and 2 of Table 5), the success rates were still poor (<10%). In iteration 3, when few categories were tried with exclusive words criteria, success rates improved (91%), while the false positive rates did not (stayed high at 24%). Through iterations 3–5, several combinations of exclusive (top 100) and frequent (top 25) words with a qualifying strategy that implements a limit on SS score “difference” between CIMS and notCIMS (rest of narratives) was tried. The 95-percentile limit on the difference should reduce the false positive rates. However, the false positive rates were still high (58%), despite the high success rates. The experiments prove that when it comes to narrow accident categories such as CIMS, the lack of exclusive vocabulary, specific to the category, seems to be a problem with the SS model—as it shares a lot of vocabulary with the broader “caught in” category and to some extent with other accident categories. Therefore, it is apparent that it becomes difficult to achieve high success rates while keeping the false positive rates below 5%.

When it comes to the ASECV model, from Table 7, it can be observed that the false positive performance for the “caught in” category steadily improved up to iteration 9 (1.76 to 0.76%, which is on par with the RF rate of 1%). It can also be observed that when vocabulary lists (clusters) are reduced, they remained critical to classification until iteration 10. During the process of reducing the false positive rates (iterations 1–9), the success rates were also reduced (from 65% to 35%). Since it is important to keep false positive rates below 1%, compromise in success rates can be justifiable. Moreover, it can help improve the success rates of RF, if “stacking” of the ASECV model is performed. With stacking, the narratives that the RF model failed to categorize as “caught in” were sent to the ASECV model, anticipating that it can reclassify some of the narratives into the correct category. At iteration 11, when stacking was performed in such a fashion, the ASECV model improved its false positive rate performance (0.42%). Even though the success rate achieved was not a high number (8%), the very low false positive rates are highly desirable for stacking. For this reason, in the “stacked model” (Table 7), the RF success rate was improved by 2.28% (from 71% to 73.28%). It can also be observed that at the 0.42% false positive rate, the ASECV vocabulary set only included three words, that are, “squeeze”, “crush”, and “pinch”. This demonstrates how important these words are to the model prediction accuracy on the test set. Even elimination of one word (“pinch”) can result in dramatic reduction in success rates to 0.04% (iteration 10). This again shows how important the above-noted three words are to the success rates in the classification process, as demonstrated in iterations 9 and 11. For this reason, the three-word set in cluster 1 is used in the stacking process (iteration 11). This shows how the field-specific expert knowledge can be leveraged in terms of using the proper set of words that can describe the key mechanisms of the accident occurrences. Accident-related vocabulary changes from industry to industry and often depends on the narrator’s style. If the vocabulary sets used can reflect these aspects, the classification success could be improved.

It can be understood that given the scope of this paper, to reduce the overall time expended and complexity in past modelling (RF and SS), the ASECV method is proposed. The ASECV algorithm operates on heuristics (linguistic rules) that are based on expert knowledge in mine safety. This approach dramatically reduced the size of the vocabulary sets used by past methods. In addition, as found by the authors, stacking the RF method with ASECV is an optimal method in terms of model performance compared to previous standalone RF approach.

6. Conclusions

NLP tools, if used strategically, can help process vast amounts of text into meaningful information. Accident narratives are concise descriptions of accidents that can help mines, as well as federal agencies, in analyzing accident data. Accident classification is the first and foremost of the steps involved in finding root causes for accidents. However, the process is manually intensive due to the sheer volume of text to deal with. In their previous application of NLP techniques and RF methods, the authors were able to successfully classify MSHA and non-MSHA (a surface metal mine) narratives. It was found that multiple RF classification models—one for each accident category—were effective in classification when compared to one model that can perform all classification tasks. The prediction success achieved was 75% and 96% (across the board) for MSHA and non-MSHA narratives, respectively. Minimizing the false positive rates to within 5% is of great importance for the accuracy of the model, and the RF models previously developed were able to achieve such rates. The insights provided (from previous work in Ganguli et al., 2021 [6]) related to how often certain accidents occur at the partnering mine site (non-MSHA) helped the mine operator in taking preventive measures.

Furthering the research in the area, and in an attempt to improve upon the previous success rates (from RF), three approaches were presented in this paper. Two were novel approaches named, the similarity scores (SS) model and accident-specific expert choice vocabulary (ASECV) model, respectively. The third one is a "stacking approach", where one of the successful novel methods is applied in combination with the RF approach. In the SS model, test narratives are scored based on the word weights they carry in accident-specific vocabulary lists developed during the training process. Word weights in the vocabulary lists are proportionate to their frequency of occurrence in narratives that belong to the related accident category. Narratives that scored highest among the vocabulary lists are assigned with the appropriate category. Since the model depends on the word frequencies, classification strategies were devised based on how rare or common a word is for an accident type. The model produced mixed success rates, but the false positive rates were very high, which is not desirable. This could be due to the fact most of the accident type narratives shared a certain amount of common (frequent) vocabulary. To compensate for this problem, the ASECV model was developed. The heuristically developed model's predictions are based on vocabulary sets (clusters) that best describe the accident's mechanism(s). Industry-specific expert knowledge was used in this context. By experimenting with the combination of key words—arranged in clusters of high, medium, and low importance—in predicting the targeted category, different success rates were achieved. With the ASECV model, the false positive rates were successfully reduced to below 1%, which is highly desirable. Even though the success rates of classification are moderate (39% across the board), such high accuracy rates helped the overall success of the ASECV model in "stacking" with RF model.

The ASECV classification model, when applied to the narratives (as a stacked option) —where the RF method failed to classify for "caught in" category—yielded an 8% success rate with less than 1% false positive rate. The combined stacked model (RF-ASECV) thus improved the previous RF model success rates from 71% to 73.28%. This, in turn, proves that when industry-specific knowledge is used in developing models along with powerful text-processing tools such as NLP, the accuracy of prediction can be improved. This paper demonstrates that use of domain-specific (mining industry) knowledge can improve accident classification success beyond what was achieved by RF, a popular machine learning technique. The models developed in the paper are focused on the "caught in" category due to limitations of scope. However, widening the application of the models to more accident categories, the exploration of semantic rules, and alternate performance measuring metrics will be considered for future research. Application of models developed to non-MSHA data is out of scope for the research presented in this paper; however, it can be considered for future research as well.

Author Contributions: Conceptualization, R.P. and R.G.; methodology, R.P. and R.G.; software, R.P. and R.G.; validation, R.P. and R.G.; formal analysis, R.P.; investigation, R.P.; resources, R.G.; data curation, R.P. and R.G.; writing—original draft preparation, R.P.; writing—review and editing, R.P. and R.G.; visualization, R.P. and R.G.; supervision, R.G.; project administration, R.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kaplan, R.; Berry-Rogghe, G. Knowledge-Based Acquisition of Causal Relationships in Text. *Knowl. Acquis.* **1991**, *3*, 317–337. [CrossRef]
2. Garcia, D. COATIS, an NLP System to Locate Expressions of Actions Connected by Causality Links. In Proceedings of the 10th European Workshop on Knowledge Acquisition, Modeling and Management, Sant Feliu de Guixols, Spain, 15–18 October 1997; Springer: Berlin/Heidelberg, Germany, 1997; pp. 347–352.
3. Hirschberg, J.; Manning, C.D. Advances in Natural Language Processing. *Sci. Spec. Sect. Artif. Intell.* **2021**, *349*, 261–266. [CrossRef] [PubMed]
4. Posse, C.; Matzke, B.; Anderson, C.; Brothers, A.; Matzke, M.; Ferryman, T. Extracting Information from Narratives: An Application to Aviation Safety Reports. *IEEE Aerosp. Conf. Proc.* **2005**, *2005*, 3678–3690. [CrossRef]
5. MSHA. Mine Data Retrieval System: Accident Database. Available online: https://www.msha.gov/mine-data-retrieval-system (accessed on 31 January 2021).
6. Ganguli, R.; Miller, P.; Pothina, R. Effectiveness of Natural Language Processing Based Machine Learning in Analyzing Incident Narratives at a Mine. *Minerals* **2021**, *11*, 776. [CrossRef]
7. Goldberg, D.M.; Zaman, N. Topic Modeling and Transfer Learning for Automated Surveillance of Injury Reports in Consumer Product Reviews. In Proceedings of the Annual Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2020; pp. 1016–1025. [CrossRef]
8. Zaman, N.; Goldberg, D.M.; Gruss, R.J.; Abrahams, A.S.; Srisawas, S.; Ractham, P.; Şeref, M.M.H. Cross-Category Defect Discovery from Online Reviews: Supplementing Sentiment with Category-Specific Semantics. *Inf. Syst. Front.* **2021**, 1–21. [CrossRef]
9. OSHAcademy. What Are Caught-in or-between Hazards? Available online: https://www.oshatrain.org/courses/mods/807m1.html#:~{}:text=According%2520to%2520O%2520SHA%25%25202C%25%252020caught%252Din,between%2520parts%2520of%2520an%2520object (accessed on 4 April 2022).
10. Hethmon, T.; Brnich, M.; Hebig, D.; Huber, B.; Kramer, S.; Lingenfelder, D.; Pedersen-Howard, M.; Mcnamara, B.; Rajapaske, S.; Ross, C.; et al. Body of Knowledge for Mining Safety and Health Management. Society for Mining, Metallurgy, and Exploration, Inc. (SME), Englewood, CO, USA. *Min. Eng.* **2018**, *70*, 41–43.
11. Serpella, A.F.; Ferrada, X.; Howard, R.; Rubio, L. Risk Management in Construction Projects: A Knowledge-Based Approach. *Procedia-Soc. Behav. Sci.* **2014**, *119*, 653–662. [CrossRef]
12. Xu, N.; Ma, L.; Wang, L.; Deng, Y.; Ni, G. Extracting Domain Knowledge Elements of Construction Safety Management: Rule-Based Approach Using Chinese Natural Language Processing. *J. Manag. Eng.* **2021**, *37*, 04021001. [CrossRef]
13. MathWorks. BagOfWords: Bag-of-Words Model. Available online: https://www.mathworks.com/help/textanalytics/ref/bagofwords.html (accessed on 10 April 2022).
14. MathWorks. WordEmbedding: Word Embedding Model to Map Words to Vectors and Back. Available online: https://www.mathworks.com/help/textanalytics/ref/wordembedding.html?searchHighlight=wordembedding&s_tid=srchtitle_word%20embedding_1 (accessed on 11 April 2022).
15. Firth, J.R. A Synopsis of Linguistic Theory, 1930–1955. In *Studies in Linguistic Analysis*; Basil Blackwell: Oxford, UK, 1957.
16. Ubeynarayana, C.U.; Goh, Y.M. An Ensemble Approach for Classification of Accident Narratives. In Proceedings of the ASCE International Workshop on Computing in Civil Engineering, Seattle, WA, USA, 25–27 June 2017.

17. Goldberg, D.M. Characterizing Accident Narratives with Word Embeddings: Improving Accuracy, Richness, and Generalizability. *J. Saf. Res.* **2022**, *80*, 441–455. [CrossRef]
18. Camacho-Collados, J.; Pilehvar, M.T. From Word to Sense Embeddings: A Survey on Vector Representations of Meaning. *J. Artif. Intell. Res.* **2018**, *63*, 743–788. [CrossRef]
19. Open Sourcing BERT: State-of-the-Art Pre-Training for Natural Language Processing. Available online: https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html (accessed on 1 July 2022).
20. Adewumi, T.P.; Liwicki, F.; Liwicki, M. Word2Vec: Optimal Hyper-Parameters and Their Impact on NLP Downstream Tasks. *arXiv* **2020**, arXiv:2003.11645.
21. Devlin, J.; Chang, M.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019), Minneapolis, MN, USA, 2–7 July 2019.
22. Rogers, A.; Kovaleva, O.; Rumshisky, A. A Primer in Bertology: What We Know about How Bert Works. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 842–866. [CrossRef]
23. IBM. Random Forest. Available online: https://www.ibm.com/cloud/learn/random-forest#:~{}:text=Provides%20flexibility%3A%20Since%20random%20forest,popular%20method%20among%20data%20scientists (accessed on 15 April 2022).
24. Iqbal, Z.; Khan, M.I.; Hussain, S.; Habib, A. An Efficient Traffic Incident Detection and Classification Framework by Leveraging the Efficacy of Model Stacking. *Complexity* **2021**, *2021*, 5543698. [CrossRef]
25. Zhao, Y.; Deng, W. Prediction in Traffic Accident Duration Based on Heterogeneous Ensemble Learning. *Appl. Artif. Intell.* **2022**, *36*, 2018643. [CrossRef]
26. Bokaba, T.; Doorsamy, W.; Paul, B.S. A Comparative Study of Ensemble Models for Predicting Road Traffic Congestion. *Appl. Sci.* **2022**, *12*, 1337. [CrossRef]
27. Tixier, A.J.P.; Hallowell, M.R.; Rajagopalan, B.; Bowman, D. Application of Machine Learning to Construction Injury Prediction. *Autom. Constr.* **2016**, *69*, 102–114. [CrossRef]
28. Goh, Y.M.; Ubeynarayana, C.U. Construction Accident Narrative Classification: An Evaluation of Text Mining Techniques. *Accid. Anal. Prev.* **2017**, *108*, 122–130. [CrossRef]
29. Wang, Z.; Lai, C.; Chen, X.; Yang, B.; Zhao, S.; Bai, X. Flood Hazard Risk Assessment Model Based on Random Forest. *J. Hydrol.* **2015**, *527*, 1130–1141. [CrossRef]
30. Smarra, F.; Jain, A.; de Rubeis, T.; Ambrosini, D.; D'Innocenzo, A.; Mangharam, R. Data-Driven Model Predictive Control Using Random Forests for Building Energy Optimization and Climate Control. *Appl. Energy* **2018**, *226*, 1252–1272. [CrossRef]
31. Pirneskoski, J.; Tamminen, J.; Kallonen, A.; Nurmi, J.; Kuisma, M.; Olkkola, K.T.; Hoppu, S. Random Forest Machine Learning Method Outperforms Prehospital National Early Warning Score for Predicting One-Day Mortality: A Retrospective Study. *Resusc. Plus* **2020**, *4*, 100046. [CrossRef]
32. Vomfell, L.; Härdle, W.K.; Lessmann, S. Improving Crime Count Forecasts Using Twitter and Taxi Data. *Decis. Support Syst.* **2018**, *113*, 73–85. [CrossRef]
33. Goh, Y.M.; Ubeynarayana, C.U.; Wong, K.L.X.; Guo, B.H.W. Factors Influencing Unsafe Behaviors: A Supervised Learning Approach. *Accid. Anal. Prev.* **2018**, *118*, 77–85. [CrossRef] [PubMed]
34. Ul Abedin, M.A.; Ng, V.; Khan, L. Cause Identification from Aviation Safety Incident Reports via Weakly Supervised Semantic Lexicon Construction. *J. Artif. Intell. Res.* **2010**, *38*, 569–631. [CrossRef]
35. Sanchez-Pi, N.; Martí, L.; Garcia, A.C.B. Text Classification Techniques in Oil Industry Applications. *Adv. Intell. Syst. Comput.* **2014**, *239*, 211–220. [CrossRef]
36. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 1997; Volume 45.
37. Scikit-Learn. Sklearn.ensemble.RandomForestClassifier. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed on 15 January 2021).