Construction site accident analysis using text mining and natural language processing techniques

Article · January 2019

4 authors, including:

Fan Zhang
National University of Singapore
7 PUBLICATIONS **365** CITATIONS
SEE PROFILE

Hasan Fleyeh
Dalarna University
72 PUBLICATIONS **723** CITATIONS
SEE PROFILE

Xinru Wang
University of Nottingham Ningbo China
15 PUBLICATIONS **87** CITATIONS
SEE PROFILE

Automation in Construction 99 (2019) 238–248

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Construction site accident analysis using text mining and natural language processing techniques

Fan Zhang[a,*], Hasan Fleyeh[a], Xinru Wang[b], Minghui Lu[c]

[a] Dalarna University, Department of Computer Engineering, Falun 79188, Sweden
[b] Research Center for Fluids and Thermal Engineering, University of Nottingham Ningbo, 315100, China
[c] Shanghai Jiao Tong University, Department of Computer Science and Engineering, 200030, China

ARTICLE INFO

Keywords:
Construction site accident analysis
Machine learning
Text mining
Natural language processing
Sequential quadratic programming
Optimization

ABSTRACT

Workplace safety is a major concern in many countries. Among various industries, construction sector is identified as the most hazardous work place. Construction accidents not only cause human sufferings but also result in huge financial loss. To prevent reoccurrence of similar accidents in the future and make scientific risk control plans, analysis of accidents is essential. In construction industry, fatality and catastrophe investigation summary reports are available for the past accidents. In this study, text mining and natural language process (NLP) techniques are applied to analyze the construction accident reports. To be more specific, five baseline models, support vector machine (SVM), linear regression (LR), K-nearest neighbor (KNN), decision tree (DT), Naive Bayes (NB) and an ensemble model are proposed to classify the causes of the accidents. Besides, Sequential Quadratic Programming (SQP) algorithm is utilized to optimize weight of each classifier involved in the ensemble model. Experiment results show that the optimized ensemble model outperforms rest models considered in this study in terms of average weighted F1 score. The result also shows that the proposed approach is more robust to cases of low support. Moreover, an unsupervised chunking approach is proposed to extract common objects which cause the accidents based on grammar rules identified in the reports. As harmful objects are one of the major factors leading to construction accidents, identifying such objects is extremely helpful to mitigate potential risks. Certain limitations of the proposed methods are discussed and suggestions and future improvements are provided.

1. Introduction

Construction industry remains globally the most dangerous work place [1,2]. There are > 2.78 million deaths every year caused by occupational accidents according to the International Labor Organization (ILO) [3]. Among which approximately one of six fatal accidents occur in the construction sector. Construction accidents not only cause severe health issues but also lead to huge financial loss. To prevent occurrence of similar accidents and promote workplace safety, analysis of past accidents is crucial. Based on the results of cause analysis, proper actions can be taken by safety professionals to remove or reduce the identified causes. It is also noted that one major factor contributing to the risk of an accident is the presence of harmful objects [4] such as misused tools, sharp objects nearby, damaged equipment. Mitigating strategies can be made accordingly after identification of such objects. For example, raising awareness, performing mandatory regular checks before operation of the machine which went wrong and caused the accident earlier.

In construction industry, a catastrophe investigation report is generated after a fatal accident which provides a complete description of the accident, such text data can be utilized for further analysis.

Studies of text mining, NLP and ensemble techniques for the analysis of construction accidents report are rare. Motivation of this paper is to fill this research gap. In this study, text mining and NLP techniques are applied to analyze the construction site accidents using the data from Occupational Safety and Health Administration (OSHA). Aan ensemble model is proposed to classify the causes of accidents. While in conventional majority voting mechanism, equal weights are assigned to each base classifier involved in the ensemble model. In this study, the weight of each base classifier is optimized by Sequential Quadratic Programming (SQP) algorithm. Moreover, a rule based chunker is developed to identify common objects which cause the accidents. Neither SQP optimization nor chunker algorithm is found to be applied in this field in any existing literatures.

[*] Corresponding author.
E-mail address: fzh@du.se (F. Zhang).

https://doi.org/10.1016/j.autcon.2018.12.016
Received 21 June 2018; Received in revised form 13 December 2018; Accepted 18 December 2018
Available online 24 December 2018

Major contributions of this work are:

- Various texting mining and NLP techniques are explored with respect to construction site accidents analysis.
- Ensemble algorithm which has not been well studied in this field is proposed to classify the causes of accidents and SQP algorithm is utilized to search for optimal weighs of the ensemble model.
- A rule based chunker is developed for dangerous objects extraction. Neither SQP optimization algorithm nor rule based chunker with regard to this field is found in the state of the art.
- Case studies are designed using OSHA dataset and effectiveness of the proposed approaches is verified by the experiment results.

2. Literature review

There are several studies which utilize text mining or natural language process (NLP) approaches for occupational accidents analysis. Bertke et al. [5] developed a Naïve Bayesian model to classify the compensation claims causation due to work related injuries. The proposed model achieved an overall accuracy of approximately 90%, however the accuracy of claims belongs to minor injury categories dropped. Taylor et al. [6] applied Naïve Bayesian and Fuzzy models to categorize the injury outcome and mechanism of injury for fire service incident reports extracted form from the National Firefighter Near-Miss Reporting System. Results showed that Fuzzy model achieved a sensitivity of 0.74 while sensitivity of Naïve Bayesian model is 0.678. Wellman et al. [7] proposed a Fuzzy Bayesian model to classify injury narratives into external-cause-of-injury and poisoning (*E*-code) categories. Data used in this study is the injury reports from US National Health Interview Survey (NHIS) during 1997 and 1998. The proposed model achieved an accuracy of 87.2%. Abdata et al. [8] applied Bayesian network to extract recurrent serious Occupational Accident with Movement Disturbance (OAMD) scenarios from narrative texts. It is noted that data pre-processing of this approach is time consuming and expert knowledge is required. Wellman et al. [9] proposed an approach which combined manual coded rules with machine learning algorithms for injury narratives classification. Results showed that using Logistic Regression (LR) and filtering out the bottom 30% of its predictions reviewed manually resulted an overall sensitivity of 0.89. Bertke et al. [10] compared the methods of Naïve Bayesian and Regularized Logistic Regression for auto coding the causation of injury narratives. Dataset used was from Occupational Injury and Ill-ness Classification System (OIICS), and results showed that the logistic model achieved an overall accuracy of 71% for 2-digit OIICS event/exposure classification system and 87% for first digit respectively.

In terms of the analysis of construction related accidents, Tixier et al. [11] applied Random Forest (RF) and Stochastic Gradient Tree Boosting (SGTB) algorithms to predict type of energy involved in the accident, injury type, body part affected, and injury severity using construction injury reports. Rank Probability Skill Score (PRSS) of the proposed methods ranked from 0.236 to 0.436. Tixier et al. [12] proposed a NLP approach based on hand crafted rules and keywords dictionary to extract outcomes and precursors from unstructured injury reports and achieved a recall of 0.97 and precision of 0.95, however the proposed approach was not robust to unanticipated situations. Goh et al. [13] applied support vector machine (SVM), linear regression (LR), random forest (RF), K-nearest neighbor (KNN), decision tree (DT) and Naive Bayes (NB) algorithms for construction accident narrative classification. Among which, SVM achieved a F1 score ranged from 0.45 and 0.92 and outperformed the other classifiers. The author further presented an ensemble approach for construction accident narrative classification [14]. Chokor et al. [15] applied a K-means based approach to classify injury reports. Four clusters were identified and each cluster represented a type of accident. Identified accident types were 'falls', 'struck by objects', 'electrocutions' and 'trenches collapse' respectively. Fan et al. [16] compared the text mining approach with case-based reasoning approach for accidents documents retrieval. Result showed that the text mining approach is superior in terms of recall and precision. Zou et al. [17] proposed a NLP approach based on semantic query expansion and Vector Space Model (VSM) techniques to retrieve similar accident cases. Recall of the proposed method ranged from 0.5 to 1.

3. Methodology

3.1. Text mining and natural language processing

Text mining, also referred to as text data mining, is defined as the process of deriving information from text data which is not previously known and not easy to be revealed [18]. It involves transforming text into numeric data which can be used in data mining algorithms then [19]. Natural language processing (NLP) involves the techniques of multiple areas in artificial intelligence, computational linguistics, mathematics and information science, it the approach to make computer understand natural language and perform certain tasks [20]. NLP can be utilized to analyze semantic and grammatical sutures of text while such analysis cannot be performed by text mining. In this work, five single classifiers are evaluated along with the proposed ensemble model for accident causes classification and a rule based chunking approach is proposed to identify common objects which cause the accident.

Before applying the aforementioned classifiers to text data, certain pre-processing and feature extraction steps are needed. Common steps to process text are:

Lower case and punctuation removal: This step transforms the text into lower case which reduces variation of same word, e.g., after transformation 'Employee' and 'employee' are treated as the same word. Punctuations increase the size of training data and usually do not contribute much to text analysis, thus are removed.

Stopwords removal: Stopwords are extremely common words which are of little value in helping select documents [21] and such words are excluded. Some published stopwords lists are available for example in Snowball stop word list [22] published with the Snowball Stemmer and Terrier stop word list [23] published with the terrier package. However, stopwords of different domains are different. For medical domain, words like 'pill', 'patient' occur in most documents and such words are considered stopwords while for computer product domain, potential stopwords list consists words such as 'CPU', 'memory', etc. Generally, common stopwords list does not cover such terms, a domain specific stopwords list can be complied base on acquired domain knowledge.

Tokenization: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, certain characters, such as punctuation is filtered out during the process [24].

Stemming and lemmatization: In a document, same word can be expressed in different forms, e.g. 'kill', 'kills', 'killing'. Moreover, words can be represented in different syntactic categories that have the same root form and are semantically related, e.g. 'irony', 'ironic'. The two aforementioned scenarios are common due to grammatical reasons. Stemming and lemmatization are used to reduce inflectional and derivationally related form of a word and converting it to a base form [25]. E.g. 'am', 'is', 'are' are converted to 'be', 'dog', 'dogs', 'dog's' are converted to 'dog'.

Part of speech tagging: (POS tagging) is the process of assigning parts of speech tag to each token, such as noun, verb, adjective, etc. A comprehensive list of part of speech tags can be found in Penn Treebank [26]. More details of POS tagger can are described in the books of Kristina et al. [27].

N-grams: N-grams of texts are a set of co-occurring words within a given window size n, i.e. window size of unigrams, bigrams,

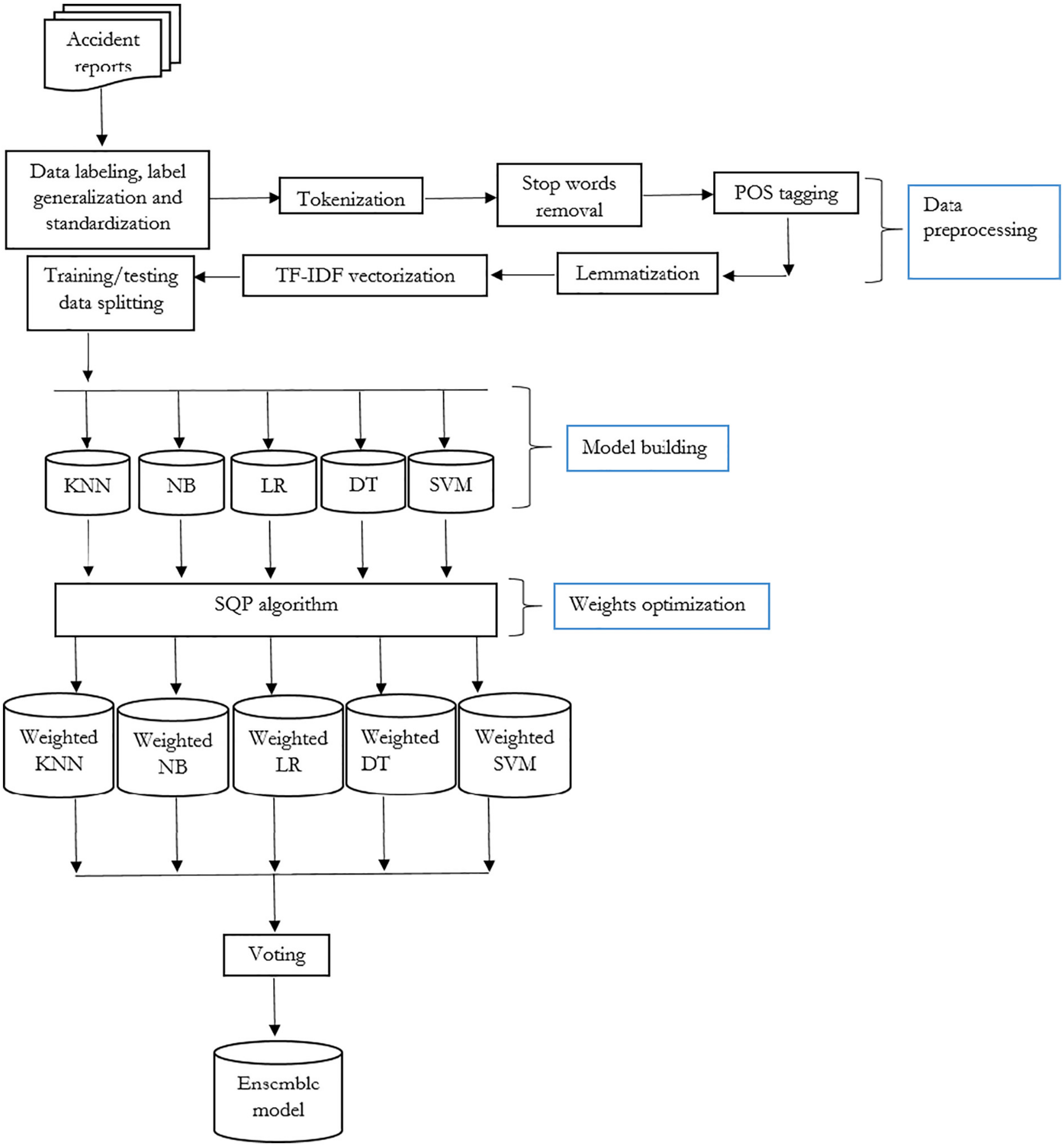

Fig. 1. Workflow of the ensemble model.

trigrams is one, two, three respectively. An example for the sentence 'he is a boy', unigrams are 'he', 'is', 'a', 'boy' while bigrams are 'he is', 'is a', 'a boy' and trigrams are 'he is a', 'is a boy'.

Term Frequency – Inverse Documents Frequency (TF-IDF) is first proposed by Karen [28]. The main idea of TF – IDF is that terms appear frequently in a large number of documents are considered less important than high frequency words appear within one document. TF – IDF is calculated by Eq. (1).

$$\mathrm{TF}-\mathrm{IDF}=\frac{n_t}{N}\cdot log\frac{K}{K_n} \tag{1}$$

In this equation, n_t is the occurrence of term t within a document, N is the number of terms in the document, K is the total number of documents and k_n is the number of documents which contain term t.

3.2. Methodology overview and the proposed approach

Five single classifies, SVM, KNN, decision tree, logistic regression, Naive Bayesian are adopted in this study and an ensemble model is proposed. To improve the conventional majority voting based mechanism, SQP algorithm is utilized to optimize the weights of the ensemble model. An overview of the five single classifiers, SQP algorithm and details of proposed ensemble model are discussed in below sections.

3.2.1. Overview of SVM

Support Vector Machine (SVM) algorithm was proposed by Vapnik [29], kernel and optimizer are two major components of SVM. Kernel enables SVM to map non-linear separable data in low dimensional space to a high dimensional feature space, in which the data can be separated linearly. Learning algorithm of SVM is based on optimization theory,

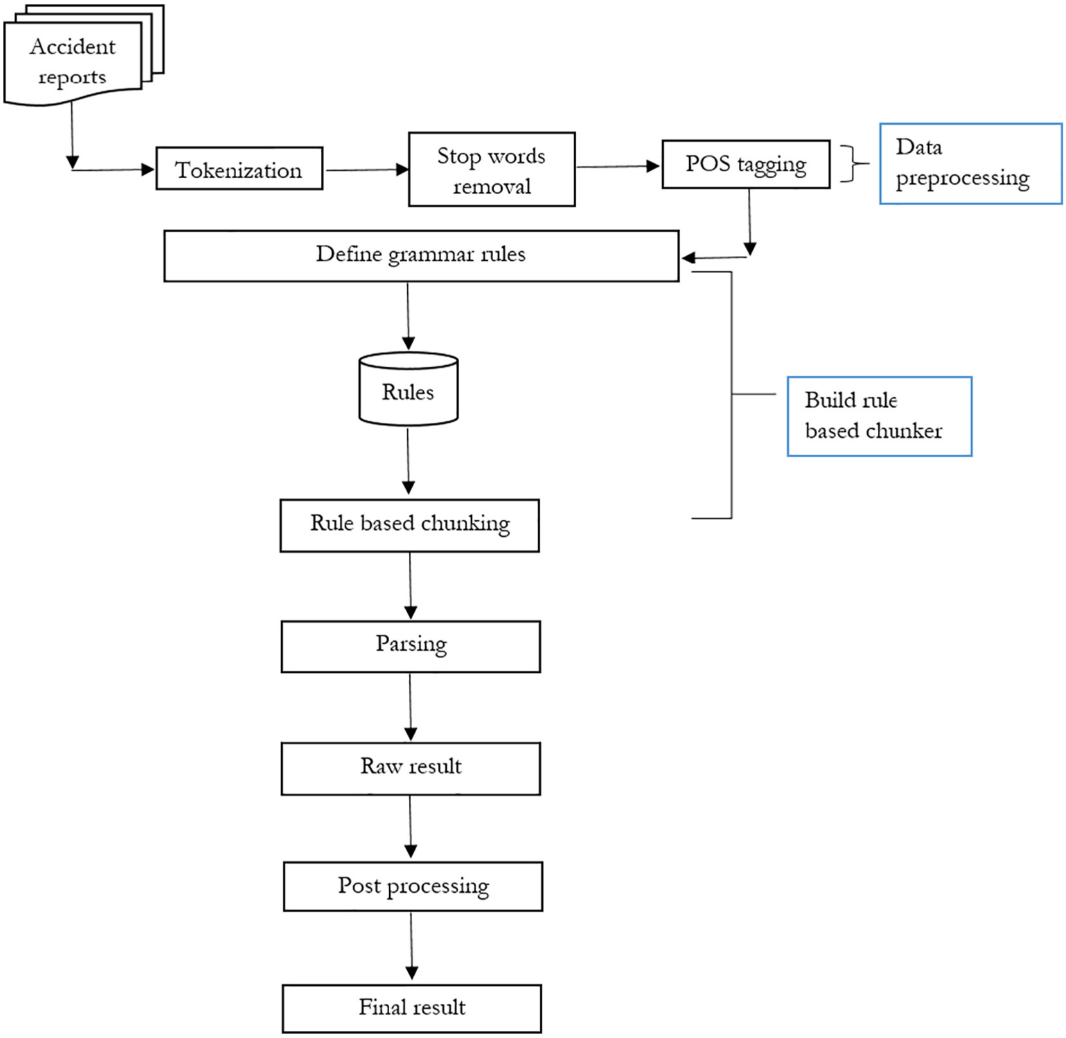

Fig. 2. Workflow of rule based chunker.

Table 1
Sample labeled case.

Title	Firefighter dies after being struck by collapsing wall
Summary	On June 17 2011 Employee #1 a volunteer firefighter with the Du Quoin Fire, Department was fighting a fire in a brick structure. He was wearing full turnout gear. While Employee #1 and another firefighter were removing a 35 ft. extension ladder from the side of the building its second story collapsed. Bricks struck Employee #1's head and neck. He was transported to a hospital where he died later that day. The second firefighter was not injured.
Cause (manually labeled)	Collapse of object

the optimizer component is utilized to solve the optimization problem. To be more specific, SVM is based on the structural risk minimization (SRM) inductive principle, it seeks to minimize an upper bound of the generalization error consisting of the sum of the training error and a confidence level. This makes SVM superior to commonly used empirical risk minimization (ERM) principle, which only minimizes the training error. Thus, generalization ability of SVM is usually better than other machine learning techniques. Details of the theory of SVM can be found in [30].

3.2.2. KNN

K-Nearest Neighbor (KNN) algorithm is widely used for pattern classification based on feature similarity. For a given unclassified sample point, it is classified by a majority vote of its neighbors, with the point being assigned to the class most common among its K nearest neighbors.

KNN is a lazy algorithm. Unlike most statistical methods which elaborate a model from the information available in the historic data, KNN considers the training set as the model itself. Thus there is no explicit training phase for KNN algorithm and during the testing phase, all training data is needed due to the lack of generalization. A KNN algorithm is characterized by issues such as number of neighbors, adopted distance, etc. More details of the KNN fundamental theory can be found in [31].

3.2.3. Decision tree

Decision tree is a hierarchical tree-based classifier. It is represented by a set of nodes, a directional graph that starts at the base with a single node and extends to many leaf nodes that represent the categories that the tree can classify. It classifies a given sample data by applying a series of rules to features of the sample data. Each rule is represented by a node and each internal node points to one child node for each possible

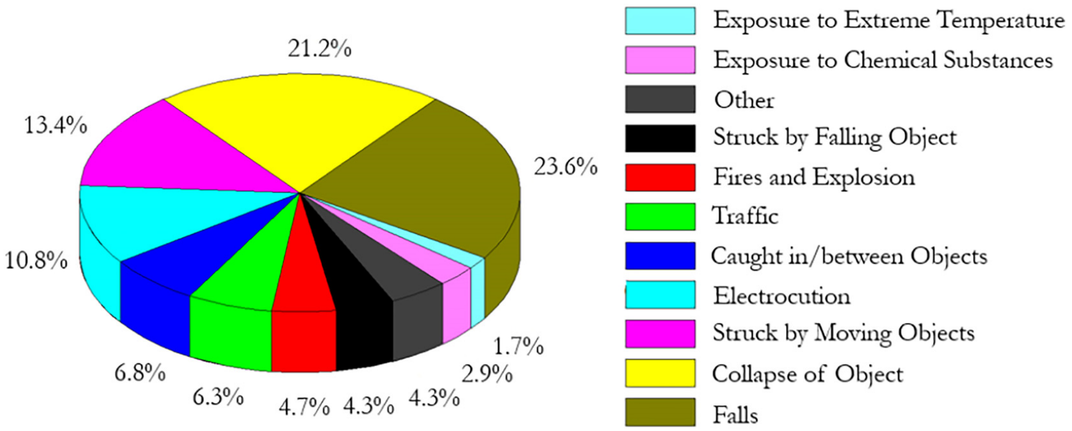

Fig. 3. Distribution of the different causes of accidents.

outcome corresponding to the applied rule. Such process is repeated until the sample data is sorted into a class by following the path from root node to leaf node. The sample data is assigned to the class which is the leaf it reaches. More details of decision tree theory can be found in [32].

3.2.4. Logistic regression

Logistic regression is an algorithm used to describe relationship between a response variable and other explanatory variables. In most cases, the response variable is discrete and consists of more than one possible value. In terms of classification problems, logistic regression method can be applied to predict the probability of a given sample data being assigned to a certain category. More details can be found in [33].

3.2.5. Naive Bayesian

Naïve Bayesian classifier is a member of probabilistic classifiers family which has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s [34]. It is widely used in text categorization tasks such as documents classification and spam email detection.

3.2.6. Sequential quadratic programming

SQP was first proposed by Wilson [35]. The algorithm relies on a profound theoretical foundation and has been proved to be both global convergent and locally super linearly convergent. Due to such advantages, SQP is one of the most efficient algorithms for solving constrained nonlinear optimization problems [36,37]. Suppose the form of a nonlinear programming problem is given by Eq. (2):

$$\text{Minimize } f_{objective}(x) \tag{2}$$

Subject to $h_i(x) = 0$, $i = 1, 2, 3\ldots, n_e$ and $g_j(x) \geq 0$, $j = 1, 2, 3\ldots, n_{ie.}$where $f_{objective}(x)$ denotes the cost function to be minimized while $h_i(x)$ and $g_j(x)$ denote the equality and inequality constraint, respectively. The corresponding Lagrangian function is given in Eq. (3).

$$L_{(x,\gamma)} = f_{objective}(x) + \sum_{i}^{n_e} \gamma_i h_i(x) + \sum_{j}^{n_{ie}} \mu_j g_j(x) \tag{3}$$

where γ_i and μ_j are Lagrangian multipliers. The corresponding Hessian matrix of Lagrangian function is given in Eq. (4).

$$H_k = H_{k-1} + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_{k-1} s_k s_k^T H_{k-1}}{s_k^T H_{k-1} s_k} \tag{4}$$

where $s_k = x_k - x_{k-1}$ and $q_k = f_{objective}(x_k) + \sum_i^{n_e} \gamma_i h_i(x_k) + \sum_j^{n_{ie}} \mu_j g_j(x_k) - f_{objective}(x_{k-1}) - \sum_i^{n_e} \gamma_i h_i(x_{k-1}) - \sum_j^{n_{ie}} \mu_j j(x_{k-1})$

During the optimization process, an approximated quadratic sub problem given by Eq. (5) is solved. As a result, monotonic descent of $f_{objective}(x)$ is ensured at each step.

$$\text{Minimize} \left(\frac{1}{2}\Delta x^T H_k \Delta_x + \nabla f_{objective}(x_k)^T \Delta_x\right) \tag{5}$$

Subject to $h_i(x_k) + \nabla h_i(x_k)^T \Delta x = 0$, $\Delta x = 0$ and $g_j(x_k) + \nabla g_j(x_k)^T \Delta x \geq 0$, $\Delta x \geq 0$

According to Eq. (5), x_{k+1} can be calculated by Eq. (6).

$$x_{k+1} = x_k + \partial_k \tag{6}$$

where $\partial_k \in (0,1)$ denotes the step size. Such process is iterated until Karush–Kuhn–Tucker (KKT) [38,39] condition is reached.

3.2.7. Ensemble model and the proposed approach

Ensemble methods combine the results of multiple base classifiers which in general are weak classifiers [40–45] to improve the robustness and the ability of generalization over a single classifier. Detailed theory of ensemble methods can be found in [46–48]. Three types of ensemble strategies are commonly used, i.e. averaging, voting and boosting. For averaging method, each classifier is built independently and then average of the predictions is calculated. As a result, variance of the combined classifier is reduced, thus a combined classifier often outperforms a single classifier. Voting is similar to averaging, while averaging is for regression, voting is used to solve classification problem. For boosting method, base classifiers are built sequentially and more weight is given to misclassified data by the weak learners of last round. Thus, the total bias is reduced during training process.

In this study, five aforementioned single classifiers, SVM, decision tree, logistic regression, KNN and Naive Bayesian are combined together to form an ensemble model for accident causes classification. Moreover, instead of assigning equal weigh to each base classifier for majority voting, SQP algorithm is utilized to optimize the weigh for each base classifier. Cross-entropy loss calculated by Eq. (7) serves as the objection function of SQP to be minimized.

$$L_{log} = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{k=0}^{K-1} y_{i,k} \log p_{i,k} \tag{7}$$

where N denotes the number of samples, K denotes the number of labels, $y_{i,\ k}$ is binary indicator if k is the correct label for sample i and $p_{i,\ k}$ denotes the predicted probability of sample i labeled as class k. Constraints are all weights values range from zero to one and sum of all weights should be one. Then the optimized weights are assigned to v corresponding base classifier for voting. Workflow of the proposed method is shown in Fig. 1. Detailed steps of will be described in the Section 3.

3.3. Chunking overview and the proposed approach

Chunking is used to analyze the constituents of a given sentence and extract wanted information such as named entities from it. A chunk is a

Table 2
Performance measures of each model.

Causes	Decision tree			KNN			Support
	Precision	Recall	F1 score	Precision	Recall	F1 score	
Caught in/between objects	0.47	0.53	0.50	0.58	0.47	0.52	15
Collapse of object	0.50	0.45	0.48	0.47	0.55	0.51	44
Electrocution	0.68	0.71	0.70	0.67	0.95	0.78	21
Exposure to chemical substances	0.00	0.00	0.00	0.00	0.00	0.00	8
Exposure to extreme temperatures	1.00	0.67	**0.80**	1.00	0.33	0.50	3
Falls	0.60	0.76	0.67	0.58	0.73	0.65	49
Fires and explosions	0.70	0.70	0.70	0.75	0.60	0.67	10
Struck by moving objects	0.20	0.17	0.18	1.00	0.08	0.15	12
Struck by falling objects	0.17	0.11	0.13	0.33	0.11	0.17	9
Traffic	0.43	0.47	0.45	0.39	0.47	0.43	19
Others	0.64	0.70	0.67	0.78	0.70	0.74	10
Weighted average/total	0.51	0.54	**0.52**	0.56	0.56	**0.53**	200

Causes	Naive Bayesian			SVM			Support
	Precision	Recall	F1 score	Precision	Recall	F1 score	
Caught in/between objects	0.50	0.07	0.12	0.56	0.33	0.42	15
Collapse of object	0.37	0.50	0.43	0.43	0.55	0.48	44
Electrocution	0.77	0.81	0.79	1.00	0.86	**0.92**	21
Exposure to chemical substances	0.00	0.00	0.00	1.00	0.12	0.22	8
Exposure to extreme temperatures	0.00	0.00	0.00	0.00	0.00	0.00	3
Falls	0.47	0.86	0.61	0.69	0.71	0.70	49
Fires and explosions	1.00	0.50	0.67	0.91	1.00	**0.95**	10
Struck by moving objects	0.00	0.00	0.00	0.80	0.33	0.47	12
Struck by falling objects	0.00	0.00	0.00	0.25	0.11	0.15	9
Traffic	0.44	0.42	0.43	0.34	0.68	0.46	19
Others	1.00	0.50	0.67	1.00	0.70	0.82	10
Weighted average/total	0.46	0.50	**0.44**	0.64	0.59	**0.58**	200

Causes	Logistic regression			Ensemble			Support
	Precision	Recall	F1 score	Precision	Recall	F1 score	
Caught in/between objects	1.00	0.13	0.24	0.71	0.33	0.45	15
Collapse of object	0.39	0.52	0.45	0.42	0.57	0.49	44
Electrocution	0.83	0.95	0.89	0.80	0.95	0.87	21
Exposure to chemical substances	0.00	0.00	0.00	0.00	0.00	0.00	8
Exposure to extreme temperatures	0.00	0.00	0.00	0.00	0.00	0.00	3
Falls	0.57	0.88	0.69	0.57	0.84	0.68	49
Fires and explosions	1.00	0.50	0.67	0.86	0.60	0.71	10
Struck by moving objects	0.00	0.00	0.00	0.00	0.00	0.00	12
Struck by falling objects	0.00	0.00	0.00	0.00	0.00	0.00	9
Traffic	0.43	0.63	0.51	0.43	0.53	0.48	19
Others	1.00	0.70	0.82	1.00	0.70	0.82	10
Weighted average/total	0.53	0.56	**0.50**	0.50	0.57	**0.52**	200

Causes	Optimized ensemble			Support
	Precision	Recall	F1 score	
Caught in/between objects	0.64	0.47	**0.54**	15
Collapse of object	0.59	0.61	**0.60**	44
Electrocution	0.86	0.90	0.88	21
Exposure to chemical substances	1.00	0.50	**0.67**	8
Exposure to extreme temperatures	0.67	0.67	0.67	3
Falls	0.77	0.73	**0.75**	49
Fires and explosions	0.77	1.00	0.87	10
Struck by moving objects	0.64	0.75	**0.69**	12
Struck by falling objects	0.40	0.22	**0.29**	9
Traffic	0.46	0.63	**0.53**	19
Others	0.89	0.80	**0.84**	10
Weighted average/total	0.69	0.68	**0.68**	200

syntactically correlated phrase which comprises a set of tokens [49].

For example, in sentence ‘I have a white cat’, the noun phrase ‘a white cat’ is a noun phrase chunk. Two major approaches to build a chunker are rule based and statistical based. For statistical based approach, a chunker can be trained using supervised learning. The overall process of training a chunker is similar to the training of a classifier which means labeled dataset is required. To be more specific, the training data should be annotated with POS tag along with the chunk tag. Moreover, the training data need to be represented in a certain format, one commonly used format is presented by sequences of token, POS tag and Inside Outside Begin (IOB) tag tuples. Chunk tag ‘B’ is assigned to a token if it is identified as the beginning of a chunk,

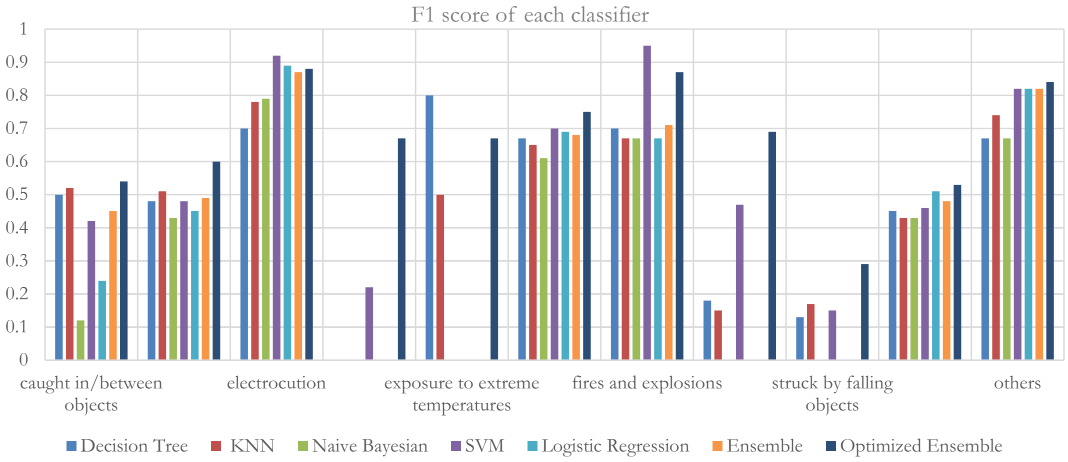

Fig. 4. F1 score for each cause of the accident for each classifier.

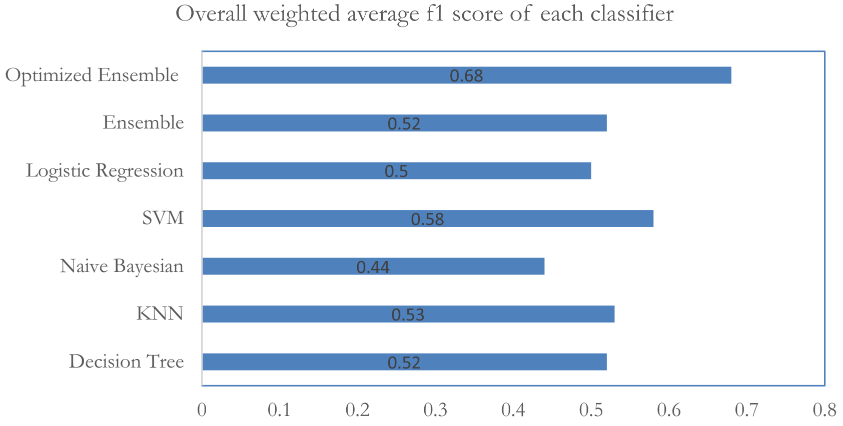

Fig. 5. The overall F1 score for each classifier.

Table 3
Time measures of the different models.

Classifier	Training time (s)	Predicting time (s)	Optimization time (s)	Total (s)
Decision tree	0.253	0.001	NA	0.254
KNN	0.003	0.019	NA	0.022
Naive Bayesian	0.009	0.001	NA	0.01
SVM	4.735	0.112	NA	4.847
Logistic regression	0.529	0.001	NA	0.53
Ensemble	5.351	0.142	NA	5.493
Ensemble with optimized weights	5.4	0.139	0.945	6.484

subsequent tokens within the chunk are labeled by tag ‘I’ and tag ‘O’ is assigned to tokens outside the chunk. For example, in the earlier example sentence, the verb ‘have’ is tagged with ‘O’, the article ‘a’ is tagged with ‘B’, while ‘white’ and ‘cat’ are assigned with ‘I’ tags.

On the contrary, data used to build a rule based chunker is not necessary to be annotated with chunk labels. POS tag information should be present though. The core step to build a rule based chunker is defining chunk grammars which are rules indicating how text should be chunked. For example, to extract the noun phrase chunk (NP chunk) from the sentence represented by a list of token and POS tag tuples: [(“a”, “DT”), (“little”, “JJ”), (“white”, “JJ”), (“dog”, “NN”), (“barked”, “VBD”), (“at”, “IN”), (“a”, “DT”), (“cat”, “NN”)]. The grammar rule can be: a NP chunk comprises of an optional determiner (DT) followed by any number of adjectives (JJ) and a noun (NN). By applying this

Table 4
Sample POS tagged title data.

Title	POS tagged title
1. Employee is burned by forklift radiator fluid	[(‘Employee’, ‘NNP’), (‘is’, ‘VBZ’), (‘burned’, ‘VBN’), (‘by’, ‘IN’), (‘forklift’, ‘NN’), (‘radiator’, ‘NN’), (‘fluid’, ‘NN’)]
2. Two workers are struck by motor vehicle and one is killed	[(‘Two’, ‘CD’), (‘workers’, ‘NNS’), (‘are’, ‘VBP’), (‘struck’, ‘VBN’), (‘by’, ‘IN’), (‘motor’, ‘NN’), (‘vehicle’, ‘NN’), (‘and’, ‘CC’), (‘one’, ‘CD’), (‘is’, ‘VBZ’), (‘killed’, ‘VBN’)]
3. Employee is struck by bales of wire and killed	[(‘Employee’, ‘NNP’), (‘is’, ‘VBZ’), (‘struck’, ‘VBN’), (‘by’, ‘IN’), (‘bales’, ‘NNS’), (‘of’, ‘IN’), (‘wire’, ‘NN’), (‘and’, ‘CC’), (‘killed’, ‘VBD’)]
4. Employee is splashed with hot water and is burned	[(‘Employee’, ‘NNP’), (‘is’, ‘VBZ’), (‘splashed’, ‘VBN’), (‘with’, ‘IN’), (‘hot’, ‘JJ’), (‘water’, ‘NN’), (‘and’, ‘CC’), (‘is’, ‘VBZ’), (‘burned’, ‘VBN’)]
5. Employee falls from flatbed trailer and later dies	[(‘Employee’, ‘NNP’), (‘falls’, ‘NNS’), (‘from’, ‘IN’), (‘flatbed’, ‘VBN’), (‘trailer’, ‘NN’), (‘and’, ‘CC’), (‘later’, ‘JJ’), (‘dies’, ‘NNS’)]
6. Worker falls from roof and breaks arm	[(‘Worker’, ‘NNP’), (‘falls’, ‘VBZ’), (‘from’, ‘IN’), (‘roof’, ‘NN’), (‘and’, ‘CC’), (‘breaks’, ‘NNS’), (‘arm’, ‘NN’)]

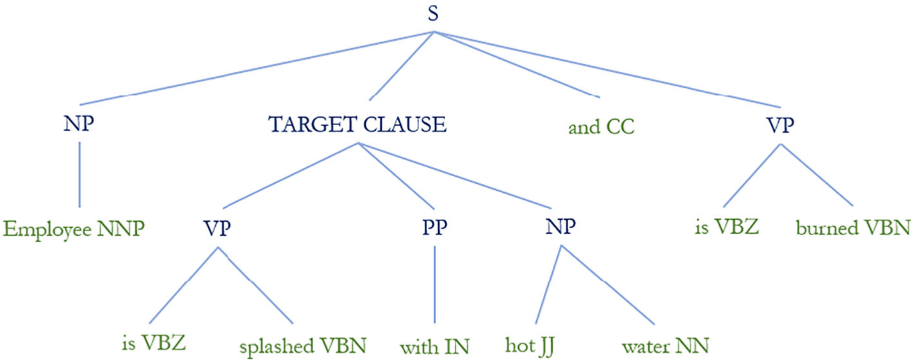

Fig. 6. Sample parsed text tree.

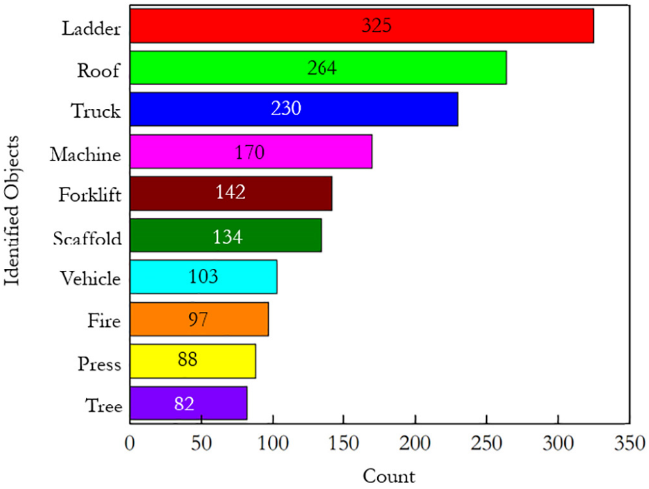

Fig. 7. The 10 common objects causing the accidents.

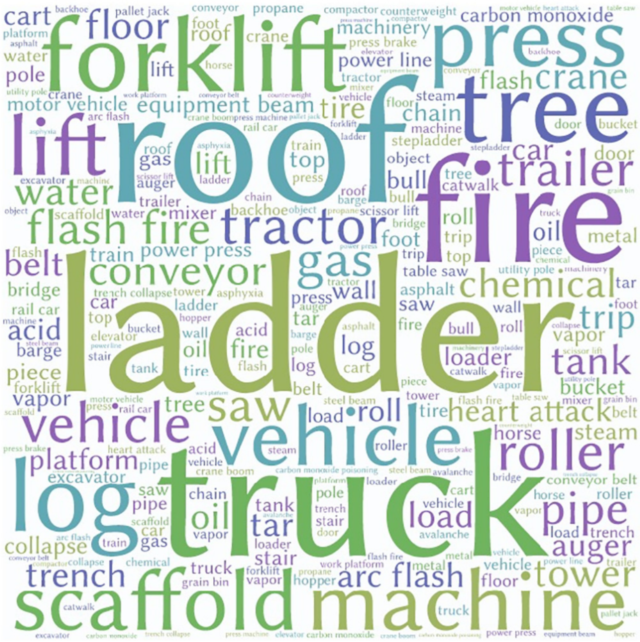

Fig. 8. Word cloud of the objects causing the accidents.

grammar rule, ‘a little white dog’ and ‘a cat’ are extracted as two NP chunks from the sample text. Such grammar rules can be coded using regular expressions.

Due to the lack of properly formatted and labeled data, a rule based chunking approach is used to identify common objects which cause the accident in this study. Workflow of the proposed method is shown in Fig. 2. Detailed steps of will be described in the Section 3.

4. Experiments and results

4.1. Experiment tools and data description

Two experiments are designed in this study. In the first experiment, an ensemble classifier is developed to classify the cause of construction accident while the second experiment is designed to identify common objects which cause the accident. Developing tool used is Python 2.7, main packages used for algorithms design are sklearn v0.19.1, pandas v0.22.0, nltk v3.2.5 and matplotlib v2.1.2 package for visualization. The original dataset from the Occupational Safety and Health Administration (OSHA) website [50] is free to download. It contains 16,323 records of construction site accidents (happened between 1983 and 2016) without labeling the cause of accidents. The report provides a detailed description of the incident, including causal factors and events which lead to the incident. In this study, case summary are used for classification while for analyzing objects caused the accidents, only case title information of the dataset is used.

Data pre-processing process for accident cause classification and object identification are different. The major difference is that supervised learning approach used for classification task requires labeled data while for object identification task an unsupervised rule based chunking approach is adopted, and hence the dataset is not necessarily to be annotated. Thus, details of the data pre-processing steps are discussed in the corresponding experiment sections separately.

4.2. Accident classification

Since classification requires labeled data and labeling the whole OSHA dataset is a tedious work due to resource construction, another dataset is invoked. In an early study of Goh [13], a processed dataset which consists of 1000 labeled records was published [51]. Therefore, this dataset is utilized instead of using the original dataset from OSHA website. A sample case is depicted in Table 1. The cases are annotated according to labels used in Workplace Safety and Health Institute (2016) [52]. Furthermore, to avoid having a case with multiple categories, the label is assigned according to the first incident if multiple incidents leading to one accident. For example, in the case summary in Table 1, the first incident is “second story collapsed” followed by the second incident “Bricks struck Employee #1's head and neck”, thus this case is labeled as “Collapse of object” accordingly. Meanwhile to reduce the number of labels representing similar causes, the cases are annotated in a more general and standard fashion. For example, the cause of the case in Table 1 is labeled as “Collapse of object” instead of “Collapse of building story”.

As a result, the dataset is labeled with 11 causes of accidents:

"caught in/between objects", "collapse of object", "electrocution", "exposure to chemical substances", "exposure to extreme temperatures", "falls", "fires and explosions", "struck by moving objects", "struck by falling objects", "traffic", and "others".

Distribution of the 11 causes of accidents is shown in Fig. 3.

After labeling, tokenization is performed to break the long sentence into separated word. However, common words across sentences with little lexical content, i.e. stop words should be removed, as stop words introduce noise to the model leading to poor accuracy and consumes more computational time.

Then, the tokens are tagged with POS tags using POS tagging based on the POS tagging result. Lemmatization is performed to remove inflectional endings and to return the base or dictionary form of a word. During the feature extraction process, n-gram representation which consists of contiguous sequence of n tokens are vectored to tf-idf matrix. In this study, uni-gram is used. As a result of data pre-preprocessing and feature extraction, documents are represented by a matrix in which each row represents one document while each column represents a word. Values within the tf-idf matrix indicate the importance of each word to a document in the dataset.

To train the classifier and evaluate the performance, the labeled dataset is randomly split into 80% and 20% for training and testing respectively. Therefore, there are 800 samples for training and 200 samples for testing. Performance of each single model, ensemble model without the optimization of weights is evaluated and compared with ensemble model based on optimized weights.

4.2.1. Results and discussions

To evaluate the model performance, F1 score proposed by Buckland et al. [53] has been widely adopted in literatures. However, support which denotes the number of true instances for each label is not considered in conventional F1 score calculation. Therefore, the average weighted F1 score given by Eq. (8) is adopted for performance measure.

$$Avg\ F1_{weighted} = \sum_{i=1}^{N} \left(\frac{S_i}{T} * F1_i \right) \tag{8}$$

where N denotes the total number of labels, S_i denotes the support of label i, T denotes the support of all labels and $F1_i$ denotes the $F1$ score of label i.

Other performance measures such as precision, recall of each model, and support for each label is also measured. Detailed results are presented in Table 2. The $F1$ score for each cause of the accident for each classifier is depicted in Fig. 4, while Fig. 5 depicts the overall $F1$ score for each classifier.

In Table 2, the weighted average F1 score of each model and the highest F1 score for each label are highlighted in bold. It is noted that the highest weighted average F1 score for labels is 0.68 achieved by the proposed ensemble model with optimized weights. While the second best model is SVM, the overall performance of Naïve Bayesian model is the worst. The result also shows that ensemble model using simple majority voting mechanism without optimization doesn't effectively improve the overall performance. Besides, it can be seen from the result that the highest F1 score is achieved by the proposed ensemble model for almost all labels. Except label 'electrocution', 'fires and explosions' and 'exposure to extreme temperatures'. To be more specific, for label 'electrocution' and 'fires and explosions' classification, SVM outperforms the rest models. For 'exposure to extreme temperatures', the highest F1 score is achieved by decision tree model. It is worth noticing that, although support of this label is extremely low, decision tree achieves the most satisfying classification result followed by the proposed model. The results also show that the performance of both Naïve Bayesian model and Logistic Regression model are poor when classifying cases with a low support while the proposed ensemble model is more robust to the value of support. It is worth noting that the proposed model achieves high F1 score for most labels. However, model performance of classifying certain labels such as caught in/between objects, truck by falling objects and traffic are not a satisfactory. This is due to the fact that natural language is imprecise. Different expressions exist for the same sentence which leads to various interpretation. In many cases, human fail to classify the correct causes which has been discussed in [13]. It is also suggested in this study that for labels with low F1 scores, additional manual checks need to be performed.

Moreover, training time, prediction time and the time of running SQP optimization algorithm is measured and presented in Table 3. The experiment is ran on a 64bit Windows 10 platform with 16GB memory and an Intel core i5 vPro 8th generation processor.

The results show that though the average weighted F1 score of Naive Bayesian classifier is the lowest, it is the most efficient in terms of the total time spent in training and predicting. On the contrary, the proposed ensemble model is the most time consuming model with extra 0.945 s spent by SQP optimization algorithm. The final weigh optimized by SQP is 0.16, 1.54e-17, 2.60e-18, 0.84 and 1.76e-17 for Decision Tree, KNN, Naïve Bayesian, SVM and Logistic Regression model respectively. Another interesting finding derived from the optimized weights is that classifier with a relatively high average weighted F1 score is assigned more weight, such as SVM and Decision Tree. It is worth noting that though the average weighted F1 score of KNN is slight higher than Decision Tree, the latter achieves the highest F1 score for 'exposure to extreme temperatures' label classification which can be a potential reason of being assigned a higher weight than KNN by SQP algorithm.

4.3. Identification of common objects causing accidents

In this experiment, rule based chunking approach is adopted to extract common objects which cause accidents from 'title' data. As it is an unsupervised learning approach, unlabeled original dataset is invoked.

Data preprocessing involves three steps, tokenization, stop words removal and POS tagging which are the same as used for cause classification experiment. After the POS tagging step, it is noted that a few words which can be critical to identifying objects in the context. Are annotated with wrong POS tags, e.g. 'injures', 'breaks', 'crush', 'swings' are tagged with 'NN' by the POS tagger. Such errors are manually corrected.

Table 4 shows the sample title data after POS tagging. After examining the 'title' data, certain syntactic structure is observed. From sample POS tagged title data 1,2,3,4 shown in Table 4, the target object is a noun or noun phase appears after a past tense verb followed by a proposition and from title 5,6, the target object is a noun or noun phase appears after a verb followed by a proposition. A chuncker is built using regular expressions according to the identified rules. Then the text data is parsed into a tree consists of a set of connected labeled nodes. A sample parsed tree using title data is shown in Fig. 6. The original text before parsing is 'Employee is splashed with hot water and is burned'.

The root node 'S' represent sentence, leaves of 'NP' node which is under the 'TARGET CLAUS' node compose the target object, i.e. 'hot water' is the object which causes the accident in this context.

After extracting the target objects using the proposed chunker, it is found some extracted noun phases are actually not legitimate objects, e.g. 'height', 'exposure', 'fall'. Thus, a post process is performed to filter out such words from the result.

4.3.1. Results and discussions

The 10 most common objects, which are 'ladder', 'root', 'truck', 'machine', 'forklift', 'scaffold', 'vehicle', 'fire', 'press', 'tree', are shown in Fig. 7. The corresponding word cloud is depicted in Fig. 8.

It is noted that the proposed approach involves certain manual inspections and corrections to improve the results.

Due to the dynamic characteristics of the natural language, sentences of same meaning can be expressed differently in terms of the structure or wording. Thus, developing exhaustive rules to all cover variations is not feasible. As a consequence, certain objects in the documents are missed out and some extracted objects are actually not legitimate. Moreover, vagueness of natural language is common and results in various interpretations from different people. In fact, it is challenging even for a human to identify the object which cause the accidents in some cases. For example, for sentence 'Employee dies of brain aneurism', the cause of accident is 'brain aneurism', however, 'brain aneurism' is not an object. For sentence 'Employee faints in trench', it is difficult to tell if 'trench' is the actual object that causes the accident without giving more context.

Apart from exhaustive rules, need to be hand crafted when dealing with dynamic structured cases. Another challenge like other unsupervised learning approaches is that the correct result is not available. In other words, the result needs to be manually checked.

Advantages of the proposed approach is that it doesn't required labeled data which reduce huge manual effort and it is an ideal approach when applied to extracting patterns from a small set of documents with similar grammar structures. Possible strategy of utilizing the proposed method is categorizing the cases with similar syntactic structure first, then designing rules to cover major categories consist of relatively good number of cases. For categories with less number of cases or cases found to be challenging to be categorized, it is suggested to review such cases manually.

It is noticed that objects extracted from this experiment present some close linkage to the causes of accidents discussed in Section 3.2. For example, objects such as ladder, fire and roof are very likely to be involved in an accident caused by collapse of object, such information can be served to further improve the performance of the classifier in the first experiment.

5. Conclusions and future work

Analyzing the construction accident reports leads to valuable knowledge of what went wrong in the past in order to prevent future accidents. To be more specific, accident causes classification is essential as prevention strategies should be developed based on different causes accordingly. Besides, identification of dangerous objects plays a crucial role in improving the safety of the working environment as well, as preventive actions can be implemented to eliminate or mitigate the potential risks of identified objects. However, manual classification of accident reports and investigation of dangerous objects involved in accidents are time consuming and labor intensive. In this work, an ensemble model with optimized weights is proposed for construction accident causes classification. The results show that the proposed model outperforms other single model in terms of the average weighted F1 score. Further, the proposed model is proved to be more robust to the cases of low support. Moreover, a rule based chunker approach is explored to identify the common objects which cause the accidents. Therefore, the aforementioned labor intensive tasks are effectively automated by the proposed approaches. Besides, the proposed approaches support the informed culture and play an important role in improving the safety information system proposed by Reason [54] which enhance the construction site safety in the long run.

Several possible future improvements can be considered, for example, data balancing [55] techniques such as under sampling, oversampling or a combination of both can be applied. Compiling a stop words list specific to construction accident domain which reduces stop words more accurately is also an approach can be considered to improve the data quality. Besides, missing corresponding context information between tokens can also cause the misclassification problem. In this study, only unigrams is used when building the classifiers, while bigrams and trigrams can preserve more context information and probably lead to a better performance of the classifier. Optimization algorithms such as GA, PSO, DE [56] can be utilized to better select the weight and model parameters of each single classifier when forming the ensemble model. Besides, instead of ensemble of weak learners, more advanced recurrent neural network model such as long short term memory (LSTM) neural network [57] can be explored in a future study. It is also noted some POS tags are not annotated properly by the published POS tagger, as POS tags are most critical information for chunking, utilizing a domain specific POS tagger is also benefit to the performance of built chunker eventually. To chunk an unlabeled large dataset, supervised learning approach requires large amount of annotated data while rule based approach requires manual checks of the results. One potential technique to explore is semi supervised [58] learning approach. Last but not the least, more NLP frameworks such as Natural Node/natural [59], Erelsgl/limdu [60] and Stanford NLP [61] can be explored in the future research.

Acknowledgement

The dataset used in this study is published and processed by Yang Miang Goh and C.U. Ubeynarayana [13]. Download link is: https://github.com/safetyhub/OSHA_Acc.git. The original data can be downloaded from below link: https://www.osha.gov/pls/imis/accidentsearch.html (Occupational Safety and Health Administration, 2016).

References

[1] H.M. Al-Humaidil, F.H. Tan, Construction safety in Kuwait, J. Perform. Constr. Facil. 24 (1) (2010) 70–77, https://doi.org/10.1061/(ASCE)CF.1943-5509.0000055.
[2] R. Navon, R. Sacks, Assessing research issues in automated project performance control (APPC), Autom. Constr. 16 (4) (2007) 474–484, https://doi.org/10.1016/j.autcon.2006.08.001.
[3] International Labor Organization (ILO), Safety and Health at Work, http://www.ilo.org/global/topics/safety-and-health-at-work/lang–en/index.html (Accessed: Oct. 2nd, 2018).
[4] R.A. Haslam, et al., Contributing factors in construction accidents, Appl. Ergon. 36 (4) (2005) 401–415, https://doi.org/10.1016/j.apergo.2004.12.002.
[5] S.J. Bertke, A.R. Meyers, S.J. Wurzelbacher, J. Bell, M.L. Lampl, D. Robins, Development and evaluation of a Naïve Bayesian model for coding causation of workers compensation claims, J. Saf. Res. 43 (5–6) (2012) 327–332, https://doi.org/10.1016/j.jsr.2012.10.012.
[6] J.A. Taylor, A.V. Lacovara, G.S. Smith, R. Pandian, M. Lehto, Near-miss narratives from the fire service: a Bayesian analysis, Accid. Anal. Prev. 62 (2014) 119–129, https://doi.org/10.1016/j.aap.2013.09.012.
[7] H.M. Wellman, M.R. Lehto, G.S. Sorock, G.S. Smith, Computerized coding of injury narrative data from the National Health Interview Survey, Accid. Anal. Prev. 36 (2) (2004) 165–171, https://doi.org/10.1016/S0001-4575(02)00146-X.
[8] F. Abdat, S. Leclercq, X. Cuny, C. Tissot, Extracting recurrent scenarios from narrative texts using a Bayesian network: application to serious occupational accidents with movement disturbance, Accid. Anal. Prev. 70 (2014) 155–166, https://doi.org/10.1016/j.aap.2014.04.004.
[9] H.R. Marucci-wellman, H.L. Corns, M.R. Lehto, Classifying injury narratives of large administrative databases for surveillance - a practical approach combining machine learning ensembles and human review, Accid. Anal. Prev. 98 (2017) 359–371, https://doi.org/10.1016/j.aap.2016.10.014.
[10] S.J. Bertke, A.R. Meyers, S.J. Wurzelbacher, A. Measure, M.P. Lampl, D. Robins, Comparison of methods for auto-coding causation of injury narratives, Accid. Anal. Prev. 88 (2016) 117–123, https://doi.org/10.1016/j.aap.2015.12.006.
[11] A.J. Tixier, M.R. Hallowell, B. Rajagopalan, D. Bowman, Application of machine learning to construction injury prediction, Autom. Constr. 69 (2016) 102–114, https://doi.org/10.1016/j.autcon.2016.05.016.
[12] A.J. Tixier, M.R. Hallowell, B. Rajagopalan, D. Bowman, Automation in construction automated content analysis for construction safety: a natural language processing system to extract precursors and outcomes from unstructured injury reports, Autom. Constr. 62 (2016) 45–56, https://doi.org/10.1016/j.autcon.2015.11.001.
[13] Y.M. Goh, C.U. Ubeynarayana, Construction accident narrative classification: an evaluation of text mining techniques, Accid. Anal. Prev. 108 (2017) 122–130, https://doi.org/10.1016/j.aap.2017.08.026.
[14] C.U. Ubeynarayana, Y.M. Goh, An Ensemble Approach for Classification of Accident Narratives ASCE International Workshop on Computing in Civil Engineering 2017, (2017), pp. 409–416, https://doi.org/10.1061/9780784480847.051.
[15] A. Chokor, H. Naganathan, W.K. Chong, M. El, Analyzing Arizona OSHA injury reports using unsupervised machine learning, Procedia Eng. 145 (2016) 1588–1593, https://doi.org/10.1016/j.proeng.2016.04.200.
[16] H. Fan, H. Li, Retrieving similar cases for alternative dispute resolution in construction accidents using text mining techniques, Autom. Constr. 34 (2013) 85–91, https://doi.org/10.1016/j.autcon.2012.10.014.
[17] Y. Zou, A. Kiviniemi, S.W. Jones, Retrieving similar cases for construction project risk management using Natural Language Processing techniques, Autom. Constr. 80

(2017) 66–76, https://doi.org/10.1016/j.autcon.2017.04.003.

[18] G. Miner, J. Elder, T. Hill, D. Delen A Fast, Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications Academic Press, (2012) (ISBN: 9780123870117).

[19] T.P. Williams, J. Gong, Predicting construction cost overruns using text mining, numerical data and ensemble classifiers, Autom. Constr. 43 (2014) 23–29, https://doi.org/10.1016/j.autcon.2014.02.014.

[20] G.G. Chowdhury, Natural language processing, Annu. Rev. Inf. Sci. Technol. 37 (1) (2003) 51–89, https://doi.org/10.1002/aris.1440370103.

[21] nlp.stanford.edu, Introduction to Information Retrieval, Dropping Common Terms: Stop Words, https://nlp.stanford.edu/IR-book/html/htmledition/dropping-common-terms-stop-words-1.html (Accessed: Oct. 2nd, 2018).

[22] snowball.tartarus.org, Snowball Stop Word List, http://snowball.tartarus.org/algorithms/english/stop (Accessed: Oct. 2nd, 2018).

[23] bitbucket.org, Terrier Stop Word List, https://bitbucket.org/kganes2/text-mining-resources/downloads (Accessed: Oct. 2nd, 2018).

[24] nlp.stanford.edu, Introduction to Information Retrieval, Tokenization, https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html (Accessed: Oct. 2nd, 2018).

[25] nlp.stanford.edu, Introduction to Information Retrieval, Stemming and Lemmatization, https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html (Accessed: Oct. 2nd, 2018).

[26] ling.upenn.edu, Alphabetical List of Part-of-speech Tags Used in the Penn Treebank Project, https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html (Accessed: Oct. 2nd, 2018).

[27] K. Toutanova, D. Christopher, Manning Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger in Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), (2000), pp. 63–70 (ISBN: 0302-9743).

[28] K.S. Jones, A Statistical Interpretation of Term Specificity and its Application in Retrieval, 60, 5 MCB University Press, 2004, pp. 493–502 (ISSN: 0022-0418).

[29] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, Berlin, Heidelberg, 1995 (ISBN:0-387-94559-8).

[30] Vladimir Vapnik, Steven E. Golowich, Alex Smola, Support vector method for function approximation, regression estimation, and signal processing NIPS'96, Proceedings of the 9th International Conference on Neural Information Processing Systems Pages 281–287 Denver, Colorado, 1996.

[31] B.V. Dasarathy, Nearest Neighbour (NN) Norms: NN Pattern Classification Techniques, IEEE Computer Society Press, 1991 (ISBN: 0818689307).

[32] Leo Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, (1984) (ISBN-13: 978-0412048418).

[33] D.W. Hosmer, S. Lemeshow, Applied Logistic Regression, John Wiley & Sons, Inc, 2000, https://doi.org/10.1002/0471722146 (Print ISBN: 9780471356325).

[34] Stuart J. Russell, Peter Norvig, Artificial Intelligence: a Modern Approach, 2nd ed., Prentice Hall, 2003 (ISBN 13: 9780137903955).

[35] R.B. Wilson, A Simplicial Algorithm for Concave Programming PhD Diss, Graduate School of Business Administration, George F. Baker Foundation, Harvard University, 1963.

[36] Woongrae Roh, Youdan Kim, Trajectory optimization for a multi-stage launch vehicle using time finite element and direct collocation methods, Eng. Optim. 34 (1) (2002) 15–32, https://doi.org/10.1080/03052150210912.

[37] H.S. Kim, Y. Kim, Trajectory optimization for unmanned aerial vehicle formation reconfiguration, Eng. Optim. 46 (1) (2014) 84–106, https://doi.org/10.1080/0305215X.2012.748048.

[38] W. Karush, Minima of functions of several variables with ineqalities as side conditions, in: G. Giorgi, T. Kjeldsen (Eds.), Traces and Emergence of Nonlinear Programming, Birkhäuser, Basel, 2014, pp. 217–245, , https://doi.org/10.1007/978-3-0348-0439-4_10 Print ISBN: 978-3-0348-0438-7.

[39] H.W. Kuhn, A.W. Tucker, Nonlinear Programming Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Calif, 1951, pp. 481–492 https://projecteuclid.org/euclid.bsmsp/1200500249.

[40] X. Zhu, C. Bao, W. Qiu, Bagging very weak learners with lazy local learning, 2008 19th International Conference on Pattern Recognition, 2008, pp. 1–4, , https://doi.org/10.1109/ICPR.2008.4761096.

[41] N. Ghasemian and M. Akhoondzadeh, Fusion of non-thermal and thermal satellite images by boosted SVM classifiers for cloud detection, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-4/W4, 2017 Tehran's Joint ISPRS Conferences of GI Research, SMPR and EOEC 2017, 7–10 October 2017, Tehran, Iran. DOI: https://doi.org/10.5194/isprs-archives-XLII-4-W4-83-2017.

[42] J.A. Aledo, J.A. Gámez, D. Molina, Tackling the supervised label ranking problem by bagging weak learners, Inf. Fusion 35 (2017) 38–50, https://doi.org/10.1016/j.inffus.2016.09.002.

[43] J. Abellán, J.G. Castellano, C.J. Mantas, A new robust classifier on noise domains: bagging of Credal C4.5 trees, Complexity 2017 (2017), https://doi.org/10.1155/2017/9023970.

[44] H. Kaur, S. Batra, HPCC: an ensembled framework for the prediction of the onset of diabetes, 4th International Conference on Signal Processing, Computing and Control (ISPCC), 2018, pp. 216–222, , https://doi.org/10.1109/ISPCC.2017.8269678.

[45] M. Farooq, E. Sazonov, Detection of chewing from piezoelectric film sensor signals using ensemble classifiers, 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2016 2016, pp. 4929–4932, , https://doi.org/10.1109/EMBC.2016.7591833.

[46] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45, https://doi.org/10.1109/MCAS.2006.1688199.

[47] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1–2) (2010) 1–39, https://doi.org/10.1007/s10462-009-9124-7.

[48] D. Opitz, R. Maclin, Popular ensemble methods: an empirical study, J. Artif. Intell. Res. 11 (1999) 169–198, https://doi.org/10.1613/jair.614.

[49] nltk.org.book, Natural language processing with Python, Extracting Information from Text https://www.nltk.org/book/ch07.html (Accessed: Oct. 2nd, 2018).

[50] Original dataset Obtained from Occupational Safety and Health Administration (OSHA), Fatality and Catastrophe Investigation Summaries, https://www.osha.gov/pls/imis/accidentsearch.html, (2016) (Accessed: Oct. 2nd, 2018).

[51] Processed OSHA Dataset Published by Goh, https://github.com/safetyhub/OSHA_Acc (Accessed: Oct. 2nd, 2018).

[52] Workplace Safety and Health Institute, Workplace Safety and Health Report 2015, https://www.wsh-institute.sg/, (2016) (Accessed: Oct. 2nd, 2018).

[53] M. Buckland, F. Gey, The relationship between recall and precision, J. Am. Soc. Inf. Sci. 45 (1) (1994) 12–19, https://doi.org/10.1002/(SICI)1097-4571(199401)45:1 < 12::AID-ASI2 > 3.0.CO;2-L.

[54] J.T. Reason, Managing the Risks of Organizational Accidents Aldershot, Hants, England; Brookfield, Vt., USA: Ashgate, (1997) (ISBN: 1840141050).

[55] R. Oqab, G. López, L. Garach, Bayes classifiers for imbalanced traffic accidents datasets, Accid. Prev. 88 (2016) 37–51, https://doi.org/10.1016/j.aap.2015.12.003.

[56] F. Zhang, C. Deb, S.E. Lee, J. Yang, K.W. Shah, Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique, Energy Build. 126 (2016) 94–103, https://doi.org/10.1016/j.enbuild.2016.05.028.

[57] M. Sundermeyer, H. Ney, R. Schlüter, From Feedforward to Recurrent LSTM Neural Networks for Language Modeling, 23, 3 (2015), pp. 517–529, https://doi.org/10.1109/TASLP.2015.2400218.

[58] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, Proceedings of the Eleventh Annual Conference on Computational Learning Theory, ACM. Madison, Wisconsin, USA, 1998, pp. 92–100, , https://doi.org/10.1145/279943.279962.

[59] NaturalNode/Natural, a General Natural Language Facility for Nodejs, https://github.com/NaturalNode/natural (Accessed: Oct. 2nd, 2018).

[60] Erelsgl/limdu, a Machine-learning Framework for Node.js Supporting Multi-label Classification, Online Learning, and Real-time Classification, https://github.com/erelsgl/limdu (Accessed: Oct. 2nd, 2018).

[61] Standford Core NLP, a Python Interface for Stanford CoreNLP, https://github.com/stanfordnlp/python-stanford-corenlp (Accessed: Oct. 2nd, 2018).