


Article

Relational Graph Convolutional Network for Text-Mining-Based Accident Causal Classification

Zaili Chen ^{1,2,†}, Kai Huang ^{2,3,†}, Li Wu ^{1,*}, Zhenyu Zhong ¹ and Zeyu Jiao ^{2,*} 

¹ Faculty of Engineering, China University of Geosciences, Wuhan 430074, China; zl.chen@cug.edu.cn (Z.C.); zy.zhong@giim.ac.cn (Z.Z.)

² Guangdong Key Laboratory of Modern Control Technology, Institute of Intelligent Manufacturing, Guangdong Academy of Sciences, Guangzhou 510070, China; kirehuang@gmail.com

³ School of Economics and Management, Beihang University, Beijing 100191, China

* Correspondence: lwu@cug.edu.cn (L.W.); zy.jiao@giim.ac.cn (Z.J.)

† These authors contributed equally to this work.

Abstract: Accident investigation reports are text documents that systematically review and analyze the cause and process of accidents after accidents have occurred and have been widely used in the fields such as transportation, construction and aerospace. With the aid of accident investigation reports, the cause of the accident can be clearly identified, which provides an important basis for accident prevention and reliability assessment. However, since accident record reports are mostly composed of unstructured data such as text, the analysis of accident causes inevitably relies on a lot of expert experience and statistical analyses also require a lot of manual classification. Although, in recent years, with the development of natural language processing technology, there have been many efforts to automatically analyze and classify text. However, the existing methods either rely on large corpus and data preprocessing methods, which are cumbersome, or extract text information based on bidirectional encoder representation from transformers (BERT), but the computational cost is extremely high. These shortcomings make it still a great challenge to automatically analyze accident investigation reports and extract the information therein. To address the aforementioned problems, this study proposes a text-mining-based accident causal classification method based on a relational graph convolutional network (R-GCN) and pre-trained BERT. On the one hand, the proposed method avoids preprocessing such as stop word removal and word segmentation, which not only preserves the information of accident investigation reports to the greatest extent, but also avoids tedious operations. On the other hand, with the help of R-GCN to process the semantic features obtained by BERT representation, the dependence of BERT retraining on computing resources can be avoided.

Keywords: accident causal classification; accident investigation reports; text mining; R-GCN; BERT



Citation: Chen, Z.; Huang, K.; Wu, L.; Zhong, Z.; Jiao, Z. Relational Graph Convolutional Network for Text-Mining-Based Accident Causal Classification. *Appl. Sci.* **2022**, *12*, 2482. <https://doi.org/10.3390/app12052482>

Academic Editors: Nikos D. Lagaros and Vagelis Plevris

Received: 30 January 2022

Accepted: 24 February 2022

Published: 27 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Accident investigation reports are usually text documents formed by professional investigators or teams through visits, conversations, viewing video surveillance and analyzing recorded data after accidents occur [1] and have been widely used in aviation, construction, transportation and other fields [2]. The process and consequences of the accident recorded in the reports can be leveraged by experts to analyze the cause of the accident, which is of great significance for preventing the recurrence of the accident or forming the accident response plan [3]. However, the current analysis of accident investigation reports mainly relies on expert experience to manually determine the cause of the accident, which requires a lot of work, and the accuracy is affected by the subjective experience of experts [4]. On 29 October 2018, an Indonesian Lion Air Boeing 737 MAX8 plane carrying 189 passengers and crew was flying from Jakarta's Soekarno Hatta International Airport to Penang Port, Bangka Belitung Province. The plane lost contact 13 min after takeoff and was later confirmed to have crashed in the waters off Karawang, West Java province [5].

Although experts have been investigating the cause of the accident as soon as possible after the accident, unfortunately, on 10 March 2019, another Ethiopian Boeing 737 MAX8 with 157 passengers and crew on board suffered the same accident [6]. If the causes of some accidents can be identified as early as possible, for example, the cause of the accident can be preliminary determined based on the records of the accident and it is possible to take appropriate measures in advance to avoid the occurrence of the accident [7].

Although the possible causes of the accident are hidden in the accident investigation report, analyzing the possible accident causes from textual records is extremely challenging and usually requires an analysis performed by an expert team composed of scholars, engineers, designers, etc., which, to a certain extent, leads to the long process of accident cause analysis [8]. Therefore, a naive idea is to build an expert system to automatically analyze textual records in accident investigation reports, which is essentially a text classification problem, that is, by constructing suitable models to mine the information in the text and classify the text into different categories [9]. Text mining is the process of extracting effective, novel, useful, understandable, valuable knowledge scattered in text documents and using this knowledge to better organize information [10].

The rapid development of artificial intelligence technology [11–13], especially natural language processing (NLP) and text mining technology, makes it possible to analyze accident investigation reports on a large scale and automatically [14]. With the help of these emerging technologies, time consumption and human error in determining the accident causes would be minimized [15] and the efficiency of analyzing would be significantly improved. A great deal of work has been conducted in existing studies to apply different models to the accident causal classification. According to the different ways of constructing models, existing research can be divided into the methods based on statistics and machine learning [16–19] and the methods based on deep neural networks [20–23]. The methods based on statistics and machine learning are mainly utilized to manually determine a series of text features, such as the term frequency, keyword search, N-grams [24], etc. These methods transform the original unstructured text data into structured feature vectors by artificially determining some features that can represent the key information of the document and, at the same time, create new features based on the existing data. The authors of [25] adopted a variety of machine learning and text mining methods, such as support vector machine (SVM) and Naive Bayes (NB). By combining them into a more powerful learning algorithm through ensemble learning methods, results showing an accuracy of 1.0, a recall rate of 0.96 and a F1-score of 0.96 were obtained. Zhang et al. [26] utilized five baseline models to classify the cause of the accident, including SVM, linear regression (LR), K-nearest neighbor (KNN), decision tree (DT) and NB, and the weight of each classifier in the integrated model was optimized by the sequential quadratic programming (SQP) algorithm. In general, the classification results of simple statistical and machine learning methods, such as keyword search or SVM, largely depend on the quality of feature selection and have a high misidentification rate in the analysis of accident causes [27].

The methods based on deep neural networks usually map the terms in the text to the word vector space, process the word vector and classify it with the help of the structure of the neural network, which has gradually become the mainstream of sequential data processing in recent years. Zhang et al. [20] exploited Word2Vec to skip the gram model to learn the word embedding from the corpus of a specific domain and embedded the learned words into the mixed structured deep neural network for accident report classification. Zhong et al. [22] proposed a latent Dirichlet assignment (LDA) algorithm model to identify risk topics and utilized convolutional neural networks (CNNs) to automatically classify hazards. Meanwhile, a word co-occurrence network (WCN) was generated to determine the relationship between hazards and word cloud (WC) technology was used for the quantitative analysis of keywords to provide a visual overview of hazard accident records. Heidarysafa et al. [28] employed deep learning methods and powerful word embedding (such as Word2Vec and GloVe) to classify accident cause values in the main cause field using text in the narrative. The results show that these methods not only can accurately

classify the causes of accidents according to the report description, but can also find the important inconsistencies in the accident report. A deep neural network is essentially a polynomial regression model, which is better characterized by the stacking of multi-layer neural units than a shallow classifier such as SVM [29]. This superb characteristic also enables the model to have the ability of processing text data and implement accident causal classification.

However, the existing methods either rely on large corpus and data preprocessing methods, which are cumbersome, or extract text information based on bidirectional encoder representation from transformers (BERT) [30], but the computational cost is extremely high. These shortcomings make it still a great challenge to automatically analyze accident investigation reports and extract the information therein. To address the aforementioned problems, this study proposes a text-mining-based accident causal classification method based on a relational graph convolutional network (R-GCN) and pre-trained BERT. On the one hand, the proposed method avoids preprocessing such as stop word removal and tokenization, which not only preserves the information of accident investigation reports to the greatest extent, but also avoids tedious operations. On the other hand, with the help of a R-GCN to process the semantic features obtained by BERT representation, the dependence of BERT retraining on computing resources can be avoided. The main contributions can be summarized as follows:

- A text-mining-based accident causal classification method based on a R-GCN and pre-trained BERT is proposed.
- The pre-trained BERT was adopted to avoid preprocessing in traditional text mining and ensure efficient text feature extraction.
- The R-GCN was utilized to avoid the expensive retraining of BERT and enable classification of accident investigation reports.
- To eliminate prediction errors that may be caused by domain GAP when embedding text features based on BERT, a gate mechanism was introduced into the R-GCN architecture.
- The proposed method gets rid of preprocessing such as tokenization and stop word removal and can quickly classify accident causes without relying on expert experience.

2. Methodology

2.1. Overall Scheme of the Proposed Method

The overall scheme of the proposed text-mining-based accident causal classification method is shown in Figure 1, which mainly includes two stages, the text feature extraction stage and the text classification stage. The text feature extraction stage is mainly based on the pre-trained BERT to map the text into a high-dimensional space to obtain a series of embedded text features. In the text classification stage, on the basis of the extracted text features, the R-GCN is utilized to obtain the corresponding category of accident causes.

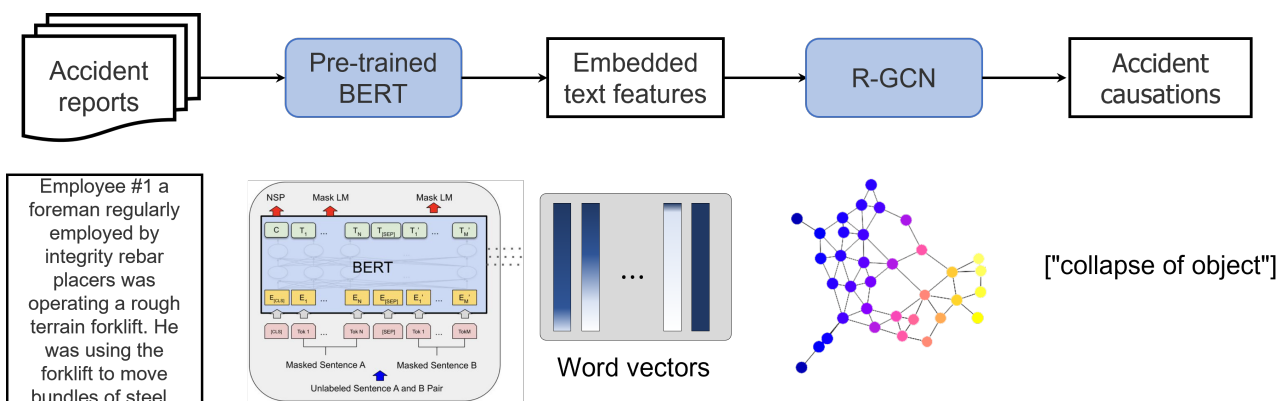


Figure 1. Overall scheme of the proposed method.

2.2. Pre-Trained BERT

Text data usually contain a lot of symbols and numbers to make it easier for readers to understand the meaning of the text, but it is difficult for computers to process and understand them [31]. In traditional methods [32–34], data cleaning is usually performed through a series of preprocessing methods, such as text cleaning, stop word removal, tokenization, data division and word embedding, in order to extract key information in the text. These preprocessing methods not only rely on pre-built corpora, but also lead to the loss of contextual semantic information in the original sentence during the preprocessing. With the excellent performance of the transformer model [35] in NLP, text information is mapped into a high-dimensional space to achieve the quantitative representation of text features. On this basis, BERT is proposed as a pre-trained language representation model. It emphasizes that the traditional one-way language model or the method of shallow splicing of two one-way language models for pre-training is no longer used as before, but a new masked language model (MLM) is exploited to generate deep bidirectional linguistic representation, as shown in Figure 2. BERT aims to pre-train deep bidirectional representations by jointly conditioning the context in all layers. Therefore, the pre-trained BERT representation can be fine-tuned with an additional output layer, suitable for the construction of state-of-the-art models for a wide range of tasks, such as question answering and language inference, without requiring significant architectural modifications for specific tasks.

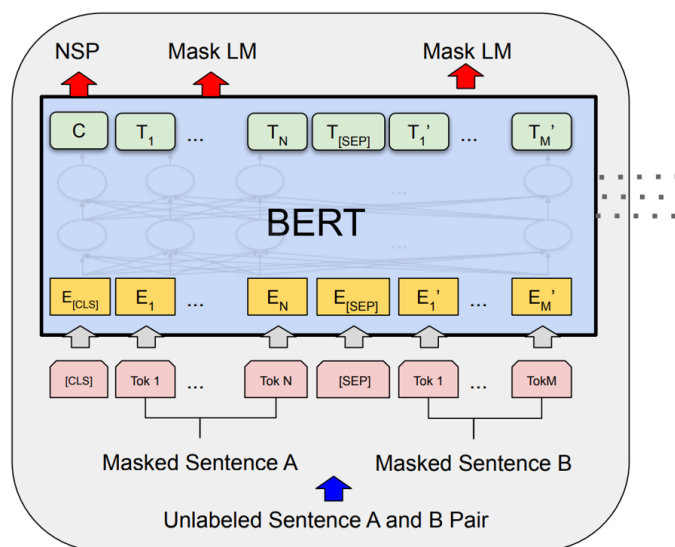


Figure 2. Architecture of BERT in [30].

Although BERT can adaptively learn word-to-word association information in texts in an unsupervised manner, retraining BERT on new datasets is expensive and computationally intensive [36]. While considering the number of accident investigation reports, it is unrealistic to repeatedly retrain BERT, but a pre-trained BERT on large datasets can cover common accident investigation report texts. All that remains to be conducted is to use an appropriate method to mine the text features output by BERT and obtain the accident cause category from the accident investigation reports.

2.3. R-GCN

A graph convolutional network (GCN) [37,38] is a topological network model based on graph theory, which was originally proposed to deal with non-Euclidean data. On the basis of a graph neural network (GNN), the convolution operation in GCNs is performed to realize the differentiable information transfer process of adjacent graph nodes. The transmitted information is usually the hidden state of the node itself, which is essentially high-dimensional feature vectors. GCNs naturally have the advantage of processing text

data [39]. Every word, symbol and datum in the text can be regarded as a node of the network. Based on the word co-occurrence relationship and the relationship between document words, a text graph can be built for a specific corpus and then a text graph convolutional network (text-GCN) model can be built. Let us suppose that a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has nodes $v_i \in \mathcal{V}$ and edges $(v_i, v_j) \in \mathcal{E}$. According to the definition by Kipf et al. [37], each node v_i contains a self-loop edge, namely, $(v_i, v_i) \in \mathcal{E}$. Let $X \in R^{n \times m}$ be a matrix containing the eigenvectors of n nodes, where m is the dimension of the eigenvectors and each row of $x_v \in R^m$ is the eigenvector of node v . Let A be the adjacency matrix of graph \mathcal{G} and D be the degree matrix of \mathcal{G} , where $D_{ii} = \sum_j A_{ij}$. The diagonal element of A is 1 due to the presence of self-loops. One convolutional layer of the GCN can only capture near-domain information. When multiple GCN layers are stacked, larger domain information is aggregated. For a single-layer GCN, the k -dimensional node feature matrix $L^{(1)} \in R^{n \times k}$ is calculated as follows:

$$L^{(1)} = \rho(\tilde{A}XW_0), \quad (1)$$

where $\tilde{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix and $W_0 \in R^{m \times k}$ is the weight matrix. As mentioned earlier, higher-order neighborhood information can be incorporated by stacking multiple GCN layers.

$$L^{j+1} = \rho(\tilde{A}L^jW_j), \quad (2)$$

where j represents the number of layers and $L^0 = X$.

Therefore, the forward propagation process in the R-GCN can be defined as

$$h_i^{(l+1)} = \text{ReLU} \left(\sum_{u \in \mathcal{N}(v_i)} \frac{1}{c_i} W^{(l)} h_u^{(l)} \right), \quad (3)$$

where $\mathcal{N}_r(v_i)$ represents the set of neighbor nodes whose relationship is r for node i , l denotes the layer number and c_i is a normalization constant. It should also be noted that the bias term is ignored in the formula and the bias is added to the calculation to promote the convergence of the model during training.

By constructing a large heterogeneous text graph containing word nodes and document nodes, global word co-occurrences can be explicitly modeled and graph convolutions can be easily applied. The number of text graph nodes $|v|$ is equal to the number of documents (corpus size) plus the number of distinct words in the corpus (vocabulary size). The text-GCN simply lets the feature matrix $X = I$ be the identity matrix, meaning that each word or document is represented as a one-hot vector as input to the text-GCN. Edges are established between nodes based on word occurrences in the document (document node–word node edges) and word co-occurrences in the entire corpus (word node–word node edges). The weight of an edge between a document node and a word node is the term frequency–inverse document frequency (TF–IDF) of that word in the document. However, due to the prior information of the syntactic structure between sentences, the traditional GCN can only represent text as an isomorphic graph and the relationship between different words may be different, which also means that the topological structure of the text is essentially heterogeneous graph [40]. Schlichtkrull et al. [41] proposed an R-GCN structure to address this heterogeneous graph problem, where different edges have different definitions of relations. Based on the above method, the update method of node vi in the graph is as follows:

$$h_i^{(l+1)} = \text{ReLU} \left(\sum_{r \in R} \sum_{u \in \mathcal{N}_r(v_i)} \frac{1}{c_{i,r}} W_r^{(l)} h_u^{(l)} \right), \quad (4)$$

where $c_{i,r}$ is a regularization constant, where the value of $c_{i,r}$ is $|N_i^r|$; $W_r^{(l)}$ is a linear transformation function, which transforms the neighbor nodes of the same type of edge using a parameter matrix $W_r^{(l)}$. Following the definition of text syntactic structure relationship in Reference [42], the relationship between texts in accident investigation reports can be divided into three types, including related, irrelevant and self-loop. Figure 3 gives an example of an analytic syntactic structure. When constructing a syntactic graph, information is also allowed to flow in the opposite direction of the syntactic dependency arc, i.e., from the dependency arc to the head.

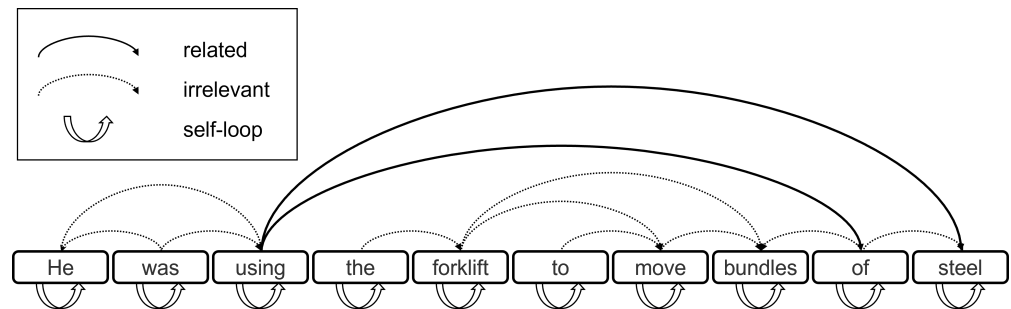


Figure 3. Example of an analytic syntactic structure. It should be noted that the syntactic structure in the figure is only an illustration and the relationships between words are not all listed. Moreover, in practical use, the input of the R-GCN is not the original text itself, but the text features embedded in the text after pre-training BERT.

Considering that the predicted grammatical information may be wrong due to the domain gap when embedding text features based on the pre-trained BERT, some mechanisms are needed to reduce the influence of false dependent edges. To this end, the gate mechanism [43,44] was introduced into the R-GCN architecture. The gate mechanism dynamically assigns a weight between 0 and 1 to the dissemination of information from different nodes. By multiplying this weight into the forward pass, the impact of incorrectly embedded features on the final result is reduced. The weight of the gate mechanism can be calculated as follows:

$$g_{u,v}^{(l)} = \text{Sigmoid}\left(h_u^{(l)} \cdot W_{r,g}\right). \tag{5}$$

Updating these weights by backpropagation, the R-GCN with a gate mechanism can be computed by

$$h_i^{(l+1)} = \text{ReLU}\left(\sum_{r \in R} \sum_{u \in N_r(v_i)} g_{u,v_i}^{(l)} \frac{1}{c_{i,r}} W_r^{(l)} h_u^{(l)}\right). \tag{6}$$

2.4. Pre-Trained BERT Combined with R-GCN

The pre-trained BERT and the R-GCN with the gate mechanism were introduced in the previous article. How to combine the two has become the only unsolved problem. Marcheggiani and Titov [45] utilized a GCN to integrate syntactic information into sequential neural networks and transformed the syntactic prior into a syntactic dependency graph, which was digested using the GCN. This architecture combines syntactic structure with BERT embeddings for text classification tasks. Following this idea, by concatenating the text features and syntactic structure information of the pre-trained BERT embedding, it can be regarded as containing all the information of the text. As shown in Figure 4, on the basis of the original BERT structure, by placing the R-GCN with gate mechanism in parallel, the embedded text features and syntactic structures can be extracted simultaneously, concatenating these two together to form features that can be used for text classification.

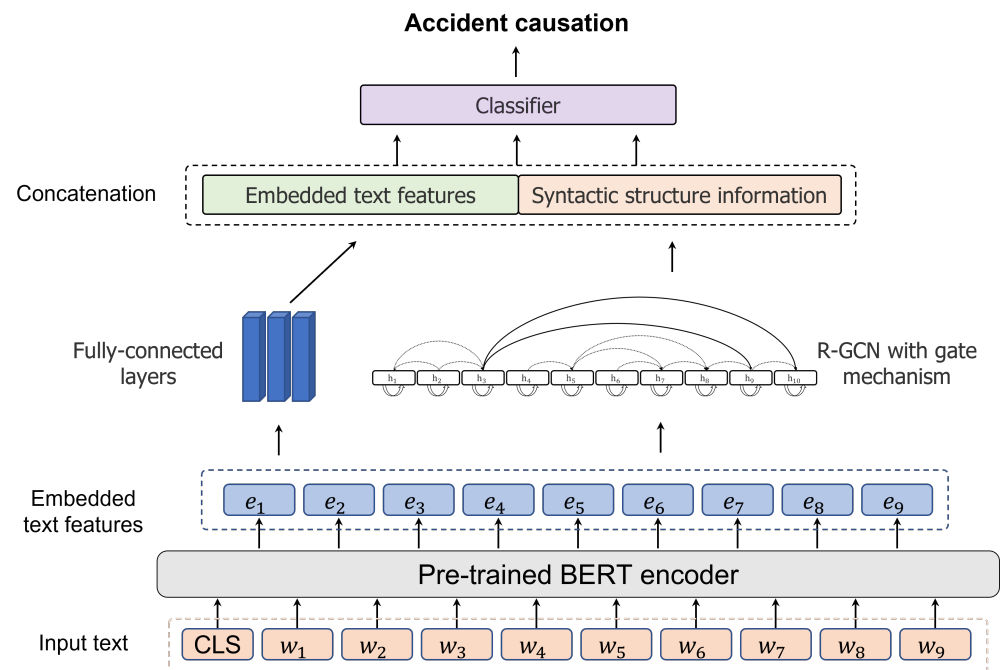


Figure 4. Structure of pre-trained BERT combined with R-GCN.

The text input to BERT adds a classification mark, such as ‘CLS’ in Figure 4, before the first sentence, so that the corresponding vector of this bit in the last layer of BERT can be used as the semantic representation of the whole sentence, which can be used for downstream classification tasks. Compared with other words in the text, this symbol without obvious semantic information more “fairly” fuses the semantic information of each word in the text, so as to better represent the semantics of the whole sentence. In addition, the text feature vectors embedded by the pre-trained BERT are coupled through a fully connected network, so that the vectors of all the words are weighted and fused to obtain features that can represent text word information. The reason for concatenating the output of the R-GCN with the embedded text features is that the graph convolution of the GCN model is actually a special form of Laplacian smoothing [46], which may mix features of vertices and make them indistinguishable.

3. Experimental Details

This section gives details of the experiment in this study, including the datasets and pre-trained models used, training settings, evaluation metrics and experimental platforms.

3.1. Dataset and Pre-Trained Model

The primary accident investigation report data used in this study were construction site accident data collected from the Occupational Safety and Health Administration (OSHA) open source database [47]. It contains the textual records of 16,323 construction site accidents that occurred from 1983 to 2016. However, the document only provides a detailed description of the event, including the causal factors and events that led to the incident. Therefore, this study adopts the labeled dataset provided by Goh and Ubeynarayana [9]. Goh and Ubeynarayana manually annotated parts of the original OSHA dataset. A new construction site accident was created with 1000 accident causal categories annotated and a total of 11 construction accident causes were derived. This dataset has been widely used in accident cause analysis [48]. The 11 accident causes were assigned different indexes and the number of various accident causes in the 1000 data were also counted; they are shown in Table 1.

Table 1. Labeled cause distribution of dataset provided by Goh and Ubeynarayana [9].

Index	Cause	Labeled Number
1	Traffic	63
2	Collapse of object	212
3	Falls	236
4	Caught in/between objects	68
5	Struck by moving objects	134
6	Others	43
7	Exposure to chemical substance	29
8	Fires and explosion	47
9	Electrocution	108
10	Struck by falling object	43
11	Exposure to extreme temperatures	17

When building the pre-trained BERT model, a large-scale general language understanding evaluation (GLUE) benchmark [49] is adopted, which is an ensemble of multiple natural language understanding tasks. Based on the work by Devlin et al. [30], a pre-trained BERT model was directly used for the text representation of accident investigation reports in this study.

3.2. Training Settings

During the training, Adam [50] was exploited as the optimizer with a $\beta_1 = 0.9$ and $\beta_2 = 0.999$ and we applied a high weight decay of 0.1. The initial learning rate was set to 10^{-4} and the batch size for the training was set to 512. It should be noted that the BERT used in the model was not fine-tuned and retrained, but directly adopted with its network parameters fixed—the ‘bert-large-uncased’ version [30] of BERT to generate raw embedded text features. Besides, batch normalization and drop out were also leveraged in all fully connected layers. Following the setting of [42], a layer of the R-GCN with a gate mechanism was utilized to capture immediate syntactic neighbor information. In addition, given that the data were still very limited, five-fold cross-validation was utilized to achieve better generalization performance and more accurate model performance estimates.

3.3. Evaluation Metrics

In the training stage of the model, the performance evaluation criteria used were the precision of accident report classification, recall rate, F1-score and average weighted F1-score. Precision is the ultimate criterium of the predicted result. It can be calculated by using Equation (7) and is obtained by dividing the true result by the sum of the true and false positive values.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

Recall is a measure of how well each unique label fits into the predicted results. It can be seen, from Equation (8), that the recall rate is the sum of the real result divided by the real value and the false negative value.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

The F1-score is the harmonic mean of precision and recall rate, in which the F1-score reaches the best value when 1 and the worst value when 0. Formula for obtaining F1-score is shown in Equation (9).

$$\text{F1-score} = \frac{2(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (9)$$

When unbalanced classes appear in the dataset, an average weighted F1-score is required. Count the number of cases and the total number of case classes that involve support for a particular tag. The average weighted F1-score can be computed by

$$\text{avg F1}_{\text{weighted}} = \sum_{i=1}^N \left(\frac{S_i}{T} * F1_i \right), \quad (10)$$

where S_i is the number of cases supported by label i and T is the total number of the dataset.

When constructing the model, in order to balance the precision and recall of the model, the training objective of the model is selected to maximize the average weighted F1 score.

3.4. Experimental Platforms

All the experiments were conducted on an Intel i7-6700 CPU at 4.0 GHz with a 16 GB RAM and a Nvidia P100 GPU with a 16 GB memory. The programming language was Python 3.6 and the integrated development environment was Anaconda 3. Several open source libraries, including SpaCy, Jieba and Deep Graph Library (DGL), were also used. Among them, DGL was used to convert each dependency graph into a DGL graph object. The R-GCN model was also implemented based on the DGL.

4. Experimental Results

Through five-fold cross-validation, the original labeled dataset was equally divided into the same quintiles, i.e., each part contained 200 accident records. For each training, four of them were used as the training set and the remaining one was regarded as the test set, ensuring that each part was treated as the test set throughout the validation process. The results of each cross-validation were measured through the average F1-score to evaluate the performance of the whole model and the model with the highest average F1-score was adopted as the final built model combining the pre-trained BERT and a R-GCN. The results of the five-fold cross-validation are illustrated in Table 2.

Table 2. Results of the five-fold cross-validation.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
AvgF1-score	0.69	0.64	0.72	0.77	0.75

As shown in Table 2, the accident causal classification model combining the pre-trained BERT and the R-GCN could achieve an average F1-score of up to 0.77. In order to more intuitively show the performance of the constructed model on each type of accident, we show the confusion matrix of the adopted model in Table 3. The corresponding procedure for calculating the average F1-score of Fold 4 is shown in Table 4.

It can be seen, from the confusion matrix in Table 3, that the model used can achieve accurate classification of most texts, but there is still a certain error for types that also contain fall or object. This shows that the proposed text-mining-based accident cause analysis could be roughly classified, but the specific accident cause analysis results still need to be further improved. To further demonstrate the advantages of the proposed model, comparative experiments were performed to numerically evaluate the improvement of the proposed method over previous methods. Traditional text mining methods, including decision tree, k-nearest neighbors (KNN), Naive Bayes and logistic regression, were also adopted to classify text in the OSHA dataset. Deep-learning-based networks, including long short-term memory (LSTM), gate recurrent unit (GRU) and symbiotic organisms search-gate recurrent unit (SGRU) [48], were also compared. Furthermore, to explore the role of the gate mechanism in the proposed method, the results of an ablation experiment were also analyzed. It should be noted that all experiments were performed on Fold 4. Due to space constraints, only the final average results, not class-by-class results, are shown in Table 5.

Table 3. Confusion matrix of the adopted model.

		Prediction											Total	TP	FN	Recall
		1	2	3	4	5	6	7	8	9	10	11				
Ground truth	1	9	1	0	1	0	0	0	0	0	1	0	12	9	3	0.75
	2	0	34	0	1	4	0	1	0	0	0	0	40	34	6	0.85
	3	1	0	39	0	0	0	1	0	0	8	1	50	39	11	0.78
	4	0	0	1	12	1	0	0	1	0	0	0	15	12	3	0.80
	5	0	2	0	0	19	0	0	0	0	2	0	23	19	4	0.83
	6	0	1	0	2	0	3	0	1	1	1	0	9	3	6	0.33
	7	0	1	0	0	0	0	5	0	0	0	0	6	5	1	0.83
	8	0	0	1	0	0	0	0	7	0	0	2	10	7	3	0.70
	9	0	0	0	0	2	2	2	0	15	0	0	21	15	6	0.71
	10	0	0	4	0	0	0	0	0	0	6	0	10	6	4	0.60
	11	0	0	0	0	1	0	0	0	0	0	3	4	3	1	0.75
Total	10	39	45	16	27	5	9	9	16	18	6	200				
TP	9	34	39	12	19	3	5	7	15	6	3					
FP	1	5	6	4	8	2	4	2	1	12	3					
Precision	0.90	0.87	0.87	0.75	0.70	0.60	0.56	0.78	0.94	0.33	0.50					

Table 4. Corresponding procedure for calculating the average F1-score of Fold 4.

	Precision	Recall	F1-Score	Number of Cases	AvgF1-Score
1	0.90	0.75	0.82	12	0.77
2	0.87	0.85	0.86	40	
3	0.87	0.78	0.82	50	
4	0.75	0.80	0.77	15	
5	0.70	0.83	0.76	23	
6	0.60	0.33	0.43	9	
7	0.56	0.83	0.67	6	
8	0.78	0.70	0.74	10	
9	0.94	0.71	0.81	21	
10	0.33	0.60	0.43	10	
11	0.50	0.75	0.60	4	

Table 5. Results of the comparison experiment and the ablation experiment.

	Average Precision	Average Recall	AvgF1-Score
Decision trees	0.48	0.55	0.51
KNN	0.49	0.52	0.50
Naive Bayes	0.57	0.54	0.55
Logistic regression	0.47	0.87	0.61
LSTM	0.58	0.64	0.61
GRU	0.70	0.61	0.65
SGRU	0.73	0.69	0.71
Ours w/o gate mechanism	0.74	0.72	0.73
Ours	0.79	0.76	0.77

From the results in Table 5, it can be seen that, although the proposed method still has some limitations, it achieved a 6% improvement of the average F1-score compared to existing research. At the same time, it can also be found from the results of the ablation experiments that the gate mechanism played a key role in the entire model. By eliminating the possible errors of the pre-trained BERT, the model achieved a 4% improvement.

5. Conclusions and Future Works

This study proposes a text mining method combining the pre-trained BERT and a R-GCN to automatically explore accident causal information in accident investigation reports. The proposed method avoids the tedious preprocessing steps of previous text mining methods and extracts text features by employing a pre-trained BERT to embed words from text reports into a high-dimensional vector space. Then, by using a R-GCN with

a gate mechanism, the syntactic structure in the text is also processed into high-dimensional vectors. By concatenating these two features and with the help of the classifier, it is possible to understand both the word and syntax of the accident investigation reports. Compared with methods such as text frequency alone, it is more accurate and concise. Compared with retraining BERT to extract text and syntax features at the same time, it is very cheap and fast. The experimental results show that the proposed method could achieve an average F1-score as high as 0.77, which exceeds existing methods and has important practical significance for accident causal classification.

However, it is undeniable that, although the existing methods have made certain breakthroughs compared with previous studies, the classification accuracy still needs to be improved. The methods proposed at present can only assist in the analysis of accident causes from accident investigation reports to a certain extent and cannot completely replace experienced experts. Especially in accident cause analyses, once the accident causal classification is wrong, it may bring unnecessary investment or mislead accident prevention. This is a key breakthrough in future work. In the future, the accuracy of accident causal classification can be improved by enriching the accident causal dataset and adding relevant labels. At the same time, on the basis of this study, we can further explore the BERT-based text information encoding method to build a more efficient expert system.

Author Contributions: Conceptualization, Z.J.; methodology, K.H. and Z.C.; validation, L.W. and Z.Z.; investigation, L.W., Z.J. and Z.Z.; writing—original draft preparation, K.H. and Z.C.; writing—review and editing, L.W. and Z.Z.; funding acquisition, Z.J. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: Figure 2 was modified from ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’ (<https://aclanthology.org/N19-1423/>, accessed on 15 January 2022), licensed under a Creative Commons Attribution 4.0 International License. This work was financially supported by GDAS’ Project of Science and Technology Development (grant no. 2021GDASYL-20210103090) and GDAS’ Project of Science and Technology Development (grant nos. 2019GDASYL-0502007 and 2020GDASYL-20200302015).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: All authors extend their sincerest thanks to the reviewers. Thanks to Yingjie Cai, Department of Electrical Engineering, Chinese University of Hong Kong, for her guidance.

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

References

1. Williams, H.; Edwards, A.; Hibbert, P.; Rees, P.; Evans, H.P.; Panesar, S.; Carter, B.; Parry, G.; Makeham, M.; Jones, A.; et al. Harms from discharge to primary care: Mixed methods analysis of incident reports. *Br. J. Gen. Pract.* **2015**, *65*, e829–e837. [[CrossRef](#)] [[PubMed](#)]
2. Reason, J. *Managing the Risks of Organizational Accidents*; Routledge: London, UK, 2016.
3. Nixon, J.; Braithwaite, G.R. What do aircraft accident investigators do and what makes them good at it? Developing a competency framework for investigators using grounded theory. *Saf. Sci.* **2018**, *103*, 153–161. [[CrossRef](#)]
4. Jiao, Z.; Lei, H.; Zong, H.; Cai, Y.; Zhong, Z. Potential Escalator-related Injury Identification and Prevention Based on Multi-module Integrated System for Public Health. *arXiv* **2021**, arXiv:2103.07620.
5. Kahfie, I.; Ramadan, M.; Rafi, S.; Perawati, D. The Crash Of Boeing 737 Max 8 And It’s Effect On Costumer Trust: Case On Lion Air Passenger. *Adv. Transp. Logist. Res.* **2019**, *2*, 764–769.
6. Johnston, P.; Harris, R. The Boeing 737 MAX saga: Lessons for software organizations. *Softw. Qual. Prof.* **2019**, *21*, 4–12.
7. Zhang, J.; Wan, C.; He, A.; Zhang, D.; Soares, C.G. A two-stage black-spot identification model for inland waterway transportation. *Reliab. Eng. Syst. Saf.* **2021**, *213*, 107677. [[CrossRef](#)]
8. Topuz, K.; Delen, D. A probabilistic Bayesian inference model to investigate injury severity in automobile crashes. *Decis. Support Syst.* **2021**, *150*, 113557. [[CrossRef](#)]
9. Goh, Y.M.; Ubeynarayana, C. Construction accident narrative classification: An evaluation of text mining techniques. *Accid. Anal. Prev.* **2017**, *108*, 122–130. [[CrossRef](#)] [[PubMed](#)]

10. Hotho, A.; Nürnberger, A.; Paaß, G. A brief survey of text mining. In *Ldv Forum*; Citeseer: Princeton, NJ, USA, 2005; Volume 20, pp. 19–62.
11. Jiao, Z.; Jia, G.; Cai, Y. A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles. *Comput. Ind. Eng.* **2019**, *135*, 1300–1311. [[CrossRef](#)]
12. Cai, Y.; Li, B.; Jiao, Z.; Li, H.; Zeng, X.; Wang, X. Monocular 3D object detection with decoupled structured polygon estimation and height-guided depth estimation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10478–10485.
13. Cai, Y.; Chen, X.; Zhang, C.; Lin, K.Y.; Wang, X.; Li, H. Semantic Scene Completion via Integrating Instances and Scene in-the-Loop. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 324–333.
14. Baclic, O.; Tunis, M.; Young, K.; Doan, C.; Swerdfeger, H.; Schonfeld, J. Artificial intelligence in public health: Challenges and opportunities for public health made possible by advances in natural language processing. *Can. Commun. Dis. Rep.* **2020**, *46*, 161. [[CrossRef](#)]
15. Kotsiantis, S.B.; Zaharakis, I.; Pintelas, P. Supervised machine learning: A review of classification techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.
16. ZHANG, Y.k.; LI, H.j. Text classification of accident news based on category keyword. *J. Comput. Appl.* **2008**, *28*, 139–140.
17. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* **2002**, *34*, 1–47. [[CrossRef](#)]
18. Kwok, J.T.Y. Automated text categorization using support vector machine. In Proceedings of the International Conference on Neural Information Processing (ICONIP), Kitakyushu, Japan, 21–23 October 1998; Citeseer: Princeton, NJ, USA, 1998.
19. Caropreso, M.F.; Matwin, S.; Sebastiani, F. Statistical phrases in automated text categorization. *Cent. Natl. Rech. Sci.* **2000**, *47*, 1–18.
20. Zhang, F. A hybrid structured deep neural network with Word2Vec for construction accident causes classification. *Int. J. Constr. Manag.* **2019**, 1–21. [[CrossRef](#)]
21. Brown, D.E. Text mining the contributors to rail accidents. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 346–355. [[CrossRef](#)]
22. Zhong, B.; Pan, X.; Love, P.E.; Sun, J.; Tao, C. Hazard analysis: A deep learning and text mining framework for accident prevention. *Adv. Eng. Inform.* **2020**, *46*, 101152. [[CrossRef](#)]
23. Soltanzadeh, A.; Mohammadfam, I.; Mahmoudi, S.; Savareh, B.A.; Arani, A.M. Analysis and forecasting the severity of construction accidents using artificial neural network. *Saf. Promot. Inj. Prev.* **2016**, *4*, 185–192.
24. Paul, D.B. Experience with a stack decoder-based hmm csr and back-off n-gram language models. In Proceedings of the Workshop Speech and Natural Language, Pacific Grove, CA, USA, 19–22 February 1991.
25. Ubeynarayana, C.; Goh, Y. An Ensemble Approach for Classification of Accident Narratives. In *Computing in Civil Engineering 2017*; The American Society of Civil Engineers: Reston, VA, USA, 2017; pp. 409–416.
26. Zhang, F.; Fleyeh, H.; Wang, X.; Lu, M. Construction site accident analysis using text mining and natural language processing techniques. *Autom. Constr.* **2019**, *99*, 238–248. [[CrossRef](#)]
27. Chen, L.; Vallmuur, K.; Nayak, R. Injury narrative text classification using factorization model. *BMC Med. Inform. Decis. Mak.* **2015**, *15*, S5. [[CrossRef](#)]
28. Heidarysafa, M.; Kowsari, K.; Barnes, L.; Brown, D. Analysis of Railway Accidents' Narratives Using Deep Learning. In Proceedings of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, 17–20 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1446–1453.
29. Cheng, X.; Khomtchouk, B.; Matloff, N.; Mohanty, P. Polynomial regression as an alternative to neural nets. *arXiv* **2018**, arXiv:1806.06850.
30. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, L.K. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
31. Rajput, A. Natural language processing, sentiment analysis, and clinical analytics. In *Innovation in Health Informatics*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 79–97.
32. Xiong, G.; Zhang, J.; Yuan, X.; Shi, D.; He, Y. Application of symbiotic organisms search algorithm for parameter extraction of solar cell models. *Appl. Sci.* **2018**, *8*, 2155. [[CrossRef](#)]
33. Karatzoglou, A.; Jablonski, A.; Beigl, M. A Seq2Seq learning approach for modeling semantic trajectories and predicting the next location. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 6–9 November 2018; pp. 528–531.
34. Zulqarnain, M.; Ghazali, R.; Ghouse, M.G.; Mushtaq, M.F. Efficient processing of GRU based on word embedding for text classification. *JOIV Int. J. Inform. Vis.* **2019**, *3*, 377–383. [[CrossRef](#)]
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 5998–6008.
36. Vasantharajan, C.; Thayasivam, U. Towards Offensive Language Identification for Tamil Code-Mixed YouTube Comments and Posts. *SN Comput. Sci.* **2022**, *3*, 1–13. [[CrossRef](#)]
37. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
38. Jiao, Z.; Jia, G.; Cai, Y. Ensuring Computers Understand Manual Operations in Production: Deep-Learning-Based Action Recognition in Industrial Workflows. *Appl. Sci.* **2020**, *10*, 966. [[CrossRef](#)]

39. Lin, Y.; Meng, Y.; Sun, X.; Han, Q.; Kuang, K.; Li, J.; Wu, F. BertGCN: Transductive Text Classification by Combining GCN and BERT. *arXiv* **2021**, arXiv:2105.05727.
40. Cao, R.; Chen, L.; Chen, Z.; Zhao, Y.; Zhu, S.; Yu, K. LGESQL: Line Graph Enhanced Text-to-SQL Model with Mixed Local and Non-Local Relations. *arXiv* **2021**, arXiv:2106.01093.
41. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Berg, R.V.D.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 3–7 June 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 593–607.
42. Xu, Y.; Yang, J. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. *arXiv* **2019**, arXiv:1905.08868.
43. Ryu, S.; Lim, J.; Hong, S.H.; Kim, W.Y. Deeply learning molecular structure-property relationships using attention-and gate-augmented graph convolutional network. *arXiv* **2018**, arXiv:1805.10988.
44. Du, C.; Wang, J.; Sun, H.; Qi, Q.; Liao, J. Syntax-type-aware graph convolutional networks for natural language understanding. *Appl. Soft Comput.* **2021**, *102*, 107080. [[CrossRef](#)]
45. Marcheggiani, D.; Titov, I. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv* **2017**, arXiv:1703.04826.
46. Li, Q.; Han, Z.; Wu, X.M. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
47. Occupational Safety and Health Administration. *Fatality and Catastrophe Investigation Summaries*; Occupational Safety and Health Administration: Washington, DC, USA, 2016.
48. Cheng, M.Y.; Kusoemo, D.; Gosno, R.A. Text mining-based construction site accident classification using hybrid supervised machine learning. *Autom. Constr.* **2020**, *118*, 103265. [[CrossRef](#)]
49. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv* **2018**, arXiv:1804.07461.
50. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.