Received 19 November 2023, accepted 26 November 2023, date of publication 7 December 2023,
date of current version 14 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3339774*

RESEARCH ARTICLE

# Extracting Fallen Objects on the Road From Accident Reports Using a Natural Language Processing Model-Based Approach

**SEUNG-SEOK LEE[1], SO-MI CHA[1], BONGGYUN KO[1], AND JE JIN PARK[2]**

[1]Department of Mathematics and Statistics, Chonnam National University, Gwangju 61186, South Korea
[2]Department of Civil Engineering, Chonnam National University, Gwangju 61186, South Korea

Corresponding authors: Bonggyun Ko (bonggyun.ko@jnu.ac.kr) and Je Jin Park (jinpark@jnu.ac.kr)

This research was supported by the BK21 FOUR (Fostering Outstanding Universities for Research, NO.5120200913674) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF).

**ABSTRACT** Keyword extraction is an effective way to quickly identify key elements in text. It can accelerate the identification of key factors that play a role in accidents when applied to incident report analysis. Our research presents an innovative process for extracting keywords from incident reports with the pre-trained natural language processing models. We utilized fine-tuning techniques to integrate a BiLSTM-CRF with a fully-connected layer and pre-trained natural language models. The process of extracting keyphrases is approached as a task of labeling sequences. To analyze incident reports from Korea, we employ pre-trained models customized for the Korean context, such as KoBERT and KoELECTRA. Our approach is assessed using a range of metrics, including accuracy, area under the curve (AUC), F1-score, slot error rate (SER), and simple matching coefficient (SMC). In contrast to traditional approaches which mainly concentrate on document summarization, our research provides a distinct method tailored to identifying falling objects as the main cause of accidents. Our findings demonstrate that the ELECTRA-based model with a BiLSTM-CRF outperforms other models, achieving an accuracy of 0.943, an AUC of 0.991, and a low SER of 0.075. The F1-score and SMC closely resemble the BERT-based model with a BiLSTM-CRF, with no significant differences observed within the 95% confidence interval. These results underscore the potential of fine-tuning pre-trained models for post-hoc traffic accident analysis. This method offers a swift preliminary step to identify the key factors before human analysis, presenting a multifaceted strategy to enhance road safety and prevent accidents.

**INDEX TERMS** BiLSTM-CRF, ELECTRA, keyphrase extraction, safety incident analysis, sequence labeling, fine-tuning.

## I. INTRODUCTION

The frequency of traffic accidents caused by road debris has been steadily increasing, presenting a pressing societal issue. Extensive research conducted by the AAA Foundation for Traffic Safety demonstrated that road debris played a prominent role in over 200,000 crashes, leading to more than 39,000 injuries and 500 fatalities in the United States between 2011 and 2014 [1]. Accidents involving debris were found to be approximately four times more likely to occur on interstate highways compared to accidents not involving debris [1]. Moreover, the AAA reported that a significant portion of the $11.5 billion spent on litter disposal nationwide in the US was allocated to the removal of unsecured debris from loads, accounting for up to 40% of the total expenditure [2]. Consequently, road debris imposes a substantial social cost on society. According to the National Highway Traffic Safety Administration, incidents caused by unsecured loads resulted in 90,226 cases, claiming 683 lives and causing 19,663 injuries in 2016 alone [2]. In this study, we define

The associate editor coordinating the review of this manuscript and approving it for publication was Agostino Forestiero.

fallen objects as encompassing a range of items, including road debris resulting from human error such as unsecured loads and vehicle parts, as well as natural objects like tree branches. Recognizing the significance of mitigating road hazards and minimizing associated societal costs, it is crucial to comprehend the patterns and trends of accidents related to road debris. This knowledge can serve as a foundation for developing effective policies and systems aimed at enhancing road safety.

This paper describes a comprehensive analysis of road safety incident reports using deep learning based methods. The analysis was conducted on a dataset consisting of 3,971 accident reports provided by the Ministry of Transportation of the Korea Expressway Corporation (KEC), covering the period from 2015 to 2019. Each accident was individually documented and included both direct impacts from fallen objects and accidents resulting from evasive actions to avoid fallen objects. The reports contained detailed textual descriptions of accidents written in Korean, along with 108 variables related to the accidents. These variables encompassed various aspects, including the extent of damage (e.g., damage rating, vehicle damage, and casualties), environmental conditions (e.g., weather, day/night, lighting conditions, road pavement conditions), operational factors (e.g., pre-accident speed, number and type of vehicles involved, vehicle size), primary causes of accidents, and information regarding the fallen objects. Additionally, the number of injuries and fatalities associated with each accident was also recorded. This study highlights the importance of leveraging deep learning analysis to gain valuable insights from road safety incident reports, contributing to the field of road safety research. Complex natural language processing models based on deep learning have made remarkable progress in processing large amounts of textual data. However, the usefulness of deep learning-based models is not limited to large-scale data. In this study, we used deep learning-based complex modeling to analyze a limited number of documents for the following reasons. First, natural language models pre-trained on large corpora can be used to gain semantic understanding of complex language in documents. Second, complex modeling is necessary to understand the context of an incident report and the relationships between words to ensure that the interpretation is accurate and relevant to the context. Third, by fine-tuning a pre-trained model for a target task, we can leverage the knowledge gained from the large corpus on which the model was pre-trained. Even when specific data for a target task is limited, transferring knowledge from a pre-trained complex model can help to understand the language of a particular domain. Finally, simple modeling methods, including traditional machine learning approaches, often require extensive feature engineering to achieve good results. Complex deep learning models reduce manual effort by automatically finding relevant features in text data. To enhance road safety and prevent accidents, it is important to identify the main causes of accidents by analyzing road incident reports. In this study, we propose a deep-learning-based approach for keyphrase extraction from accident reports to gain insight into past accidents and prevent future incidents. This research concentrates on identifying the root cause of the accident rather than the outcome. By automating the process of keyphrase extraction from text data, we aim to identify accident trends and take appropriate measures to prevent accidents. This method has the potential to significantly reduce the occurrence of accidents on roads by identifying and addressing key factors in a timely manner.

### A. ANALYSIS OF SAFETY INCIDENT REPORTSS

This section introduces the analysis of safety incident reports. The analysis of incident reports involves a series of structured processes to identify accidents based on previous accident reports to reduce the risk of reoccurrence and prevent accidents in advance. There are several approaches for analyzing accident reports in various domains. In the domain of highway accidents, computational analysis has been used to identify key information from accident reports [3] and similar techniques have been applied to health records [4]. A semi-automated classification approach has been suggested for railway hazard reports that focus on identifying specific text that may lead to accidents [5]. Natural Language Processing (NLP) techniques have also been applied to safety incident reports in the aviation industry to extract relevant information and classify reports [6].

Text retrieval and link analysis have been combined to study aviation accident reports from the U.S. National Transportation Safety Board (NTSB) [7]. This approach focused on detecting connections between topics across multiple documents. Traffic analysis [8] has also adopted machine learning and rule-based approaches to perform sentiment analysis in the area of traffic. Text mining-based fault diagnosis methods have been applied to maintenance data from high-speed rail systems, using Bayesian networks and probabilistic latent semantic analysis to extract consistent topics within the documents [9]. Text mining has also been used to identify the main causes of road crashes by analyzing contextual relationships and identifying common factors reported in accident reports [10].

### B. KEYPHRASE EXTRACTION

This section presents an overview of keyphrase extraction methods, which are used to automatically identify and extract significant phrases or words from a given text document [11]. The primary objective of keyphrase extraction is to identify the most relevant and representative phrases related to the subject matter of the document, which can facilitate efficient document indexing, retrieval, and summarization [12]. Keyphrase extraction generally adopts one of two types of approaches. Heuristic approaches are based on the statistical information of words such as word frequency

frequency [13], n-gram [14],tf-idf scores [15], word co-occurrence measures [16]. The second approach involves binary classification, where keyphrases are classified as either relevant or non-relevant using supervised or unsupervised methods. The keyphrase extraction process typically involves two steps: generating a set of phrases as candidate key phrases, and then determining which of these phrases is the keyphrase. We focus on the process of determining which candidate phrases are indeed keyphrases. In supervised approaches, the keyphrase classification is based on selecting features of the documents, such as Part-of-Speech (POS) tags with n-gram [17] and positional information [18]. Machine learning methods are then employed to identify which of the candidate phrases represent the main content of the document. Annotated documents are commonly used in supervised approaches to train the model for keyphrase extraction. In the field of keyphrase extraction, early supervised methods approached keyphrase extraction as a binary classification problem [19]. In this approach, documents are annotated with keyphrases to serve as labeled data for training a classifier that distinguishes between candidate phrases that are keyphrases and those that are not. Keyphrases are treated as positive examples and other phrases as negative examples. Once a set of candidate phrases is generated, the next step involves identifying which candidate phrases are keyphrases [11]. In contrast to supervised approaches that rely on labeled data to learn specific tasks, unsupervised approaches do not require labeled data and can be applied across multiple domains. Although supervised approaches can be more time-consuming and costly to train, they often achieve higher accuracy within a specific domain. As labeled data was used in this study, we exclusively concentrated on the supervised approach. Supervised keyphrase extraction can incorporate both domain-specific properties and external resources, such as WordNet and Wikipedia [17]. However, in cases where syntactic properties are combined with other types of features, they may not be useful for keyphrase extraction [20]. External resources such as citation information from citation networks [21], hyperlink information [22], and search query information [23] have been shown to be effective in determining the importance of candidate keyphrases. Semantic relatedness can also be used as a property, and statistical association among key phrases can be used to measure coherence and check semantic relatedness between candidate phrases [24]. Hulth [17] combined statistical properties with syntactic features and obtained better results. Various machine learning techniques, including decision trees [25], boosting [26], bagging [17], and support vector machine(SVM) [27], [28], have been used to classify keyphrases in the binary classification problem. A Multi-Layer Perceptron [29] with SVM has also been used for classification.

Although supervised keyphrase extraction as a binary classification problem has been used effectively, it has some limitations. One of the limitations is that comparing candidate phrases with each other is impossible since each candidate phrase is annotated independently. To overcome this limitation, sequence labeling was proposed as an alternative approach. Sequence labeling is a classification process for sequential data and can handle varying lengths of keyphrases without a step of generating candidate phrases. It has been successfully used in part-of-speech (POS) tagging tasks, which identifies the format of part-of-speech in the input sentence. Conditional random fields(CRF) [30] is a widely used algorithm in sequential labeling tasks before the emergence of deep neural networks with word embeddings. CRF is a softmax regression probabilistic model that takes a sequence format of length $n$ as input and outputs a sequence labeling of length $n$ using a potential function. This potential function converts various types of sequence data into high-dimensional boolean sparse vectors to help logistic regression as sequence input data. CRF considers the surrounding context by using preceding and following words and their part-of-speech directly to perform POS tagging effectively. With the CRF method, sequence labeling approaches can implement the semantic features of documents effectively without the need for a generation step to extract candidate phrases.

Unlike binary classification, sequence labeling considers the entire document to assign the keyphrase label, making it possible to capture long-term semantic dependencies within the document. Since the first CRF-based keyword extraction model was proposed in 2008, [31], CRF has been used to overcome the limitations of keyphrase extraction, incorporating multiple textual features such as tf-idf term, orthographic information, and parse-tree information [32]. The BiLSTM-CRF model is one of the earliest neural network models proposed for keyphrase extraction using a sequence labeling approach, incorporating fixed word embeddings from scholarly documents. This model can capture long-distance semantic information and dependencies from both the input and label sequences [33].

### C. FINE-TUNING METHODS

Fine-tuning refers to the process of training a pre-trained model on a specific downstream task using task-specific data. Fine-tuning allows pre-trained models to be adapted to a specific task, resulting in improved performance and efficiency. We adapt two pre-trained models, BERT and ELECTRA, to a specific task by training additional layers tailored to that task. In this method, the model is first trained on a large amount of unlabeled data, and then fine-tuned with labeled data for the target task to adjust all of the model's parameters.

The primary purpose of fine-tuning is to adapt a pre-trained language model, which has learned general language understanding from a large corpus of text, to a specific NLP task. This adaptation involves adjusting the parameters of the model to make it proficient at the task at hand. Fine-tuning involves updating the weights (parameters) using a smaller dataset that is specific to the target task. During this process, the model learns task-specific patterns for the NLP task at hand. Fine-tuning is used to transfer the knowledge and skills gained by the pre-trained model to the specific task, leveraging the model's pre-existing language understanding.

Fine-tuning can have a significant impact on experimental results. It often leads to improved performance on the target task compared to using the pre-trained model without fine-tuning. Benefits include improved accuracy, faster convergence, and the ability to handle task-specific nuances. The extent of the improvement depends on factors such as the quality and size of the fine-tuning data set, and the similarity of the task to the pre-training data. Without fine-tuning, the pre-trained model is used as a feature extractor or representation generator. However, it may not fully exploit the capabilities of the pre-trained model, especially for tasks requiring complex language understanding or handling rare and specialized vocabulary.

Deep learning models that use pre-trained representations have achieved promising results in various natural language processing (NLP) tasks, as shown in studies such as Peters et al. [34], Radford et al. [35], and Devlin et al. [36]. ULMFiT, proposed by Howard and Ruder, demonstrated that pre-trained models can improve the performance of NLP models, especially when training data is limited. ULMFiT [37] also introduced an effective fine-tuning technique that can be applied to any NLP task. Contextual word embedding models such as ELMo, OpenAI GPT, and BERT consider the context in sentences during the embedding process, unlike previously fixed word embedding representations. These contextualized word embedding models utilized pre-trained models that were in an unsupervised way with a large amount of unlabeled corpus. When using pre-trained representations for new tasks, there are two main strategies: feature-based and fine-tuning methods. The feature-based approach involves using the pre-trained representations as an extra feature when performing the specific task. On the other hand, the fine-tuning approach involves adding minimal task-specific parameters and incrementally adjusting the pre-trained parameters during the target task.

ELMo [34] and OpenAI GPT [35] are two pre-trained contextualized word embedding models that have been widely used in NLP tasks. ELMo utilizes a feature-based method, where pre-trained representations are used as additional features on top of the task-specific model structure. In contrast, OpenAI GPT adopts a fine-tuning method, where minimum task-specific parameters are added to the pre-trained parameters, and both types of parameters are fine-tuned adaptively for the target task. However, ELMo and OpenAI GPT suffer from the disadvantage of being based on unidirectional models, where the current token can only consider previously appeared tokens. ELMo tried to overcome this by using two unidirectional models, but it only shallowly concatenates their hidden states, which limits its effectiveness in capturing bidirectional dependencies.

Bidirectional Encoder Representations from Transformers(BERT) [36] is a pre-trained model that uses bidirectional context representations and learns through a masked language modeling (MLM) task, in which about 15% of input sentence tokens are masked and matched. However, effectiveness of BERT is limited by its use of a small subset of the data (only 15% of tokens are masked), and the model requires a large amount of training data to be effective due to the small number of predictions per sentence.

Efficiently learning an encoder that classifies token replacement accurately(ELECTRA) [38] introduced a new pre-training task called Replaced Token Detection (RTD). Unlike BERT, which only masks about 15% of input tokens to make predictions, ELECTRA applies RTD to all input tokens, making it more efficient and achieving better performance on downstream tasks. ELECTRA processes binary classification in which a generator replaces some tokens in real input sentences with plausible fake tokens, and the discriminator guesses whether each token is an original or replaced token generated by the generator. ELECTRA can learn much faster than BERT and perform better on downstream tasks. In this paper, we used pre-trained BERT and ELECTRA models on a Korean corpus.

## II. METHOD

### A. PROPOSED MODEL

We approach the task of extracting keyphrases from accident reports as a sequence labeling task. The semantic understanding of the nlp models is crucial for the quality of the extracted keyphrases. We use the pre-trained nlp models, BERT and ELECTRA, with a fully-connected(FC) layer and BiLSTM-CRF layer. Pre-trained models go through a process of fine-tuning to be optimized for specific tasks. Fine-tuning involves adding the target task-specific layers last. The layer weights are then retrained to optimize for the target task. BERT and ELECTRA are transfer learning models that learn language representations by performing contextualized word embeddings through pre-training with a large unlabeled corpus. They are used for various natural language processing tasks. Contextualized word embedding models such as BERT and ELECTRA are trained to capture complex linguistic and semantic relationships within sentences and paragraphs. The pre-trained model embeds the words in the sequence into a fixed dimension with a numerical representation. The sequence labeling process can be explained with a set of $x = \{x_1, x_2, \ldots, x_n\}$ as tokenized input sequence where xt represents t-th token, a set of $w = \{w_1, w_2, \ldots, w_n\}$ embedding vector corresponding to each input text x, and a set of label $y = \{y_1, y_2, \ldots, y_n\}$ as output label sequence where yt represents corresponding label of t-th token. The pre-trained models map each tokenized word $x_t$ in the input sequence to a numerical vector $w_t$. The shape of the output label $y$ is determined by the keyphrase extraction method. With a FC layer, each token in a sequence is assigned a label, which has three elements, where $s_{start}$ and $s_{end}$ indicate the start and end indices of a keyphrase in tokenized sentences, and the last element $s_{present}$ indicates the presence or absence of a fallen object in the sentence. With a BiLSTM-CRF layer, each token is assigned two binary values of 1 and 0. This binary array marks keyphrase tokens in the sequence as '1' and all other tokens as '0'. We add a FC layer or a BiLSTM-CRF layer to each NLP model to fine-tune it for the keyphrase extraction task. The BiLSTM-CRF layer accounts for label dependencies across the sequence by encoding coherent relationships within a sequence.

### B. PRE-TRAINED LANGUAGE MODEL

In this section, we introduce our proposed model, which utilizes a combination of a language model and keyphrase extraction method, and provide detailed descriptions of their implementations. Language models are designed to assign probabilities to tokens in a sequence, and our goal is to generate a sequence of words that fits the target task for a given sentence. We utilized two types of pretrained language models, BERT and ELECTRA, and adapted them to the target task by adding either a FC layer or a BiLSTM-CRF layer. We used KoBERT [39] and KoELECTRA [40], which were specifically trained on the Korean corpus, as our target data was in Korean.

#### 1) BERT

BERT is a transformer-based model that utilizes only encoders and self-attention to grasp the context of the entire sentence with a bidirectional model. The input of the BERT model is comprised of token, segment, and position embeddings generated through WordPiece embedding. Token embeddings begin with a Special Classification token (CLS) and distinguish between sentences using a Special Separator token (SEP). Segment embeddings are used to differentiate between sentences via the SEP tokens. Position embeddings capture positional information through learning. BERT performs pre-training using two unsupervised prediction tasks, Masked Language Model (MLM) and Next Sentence Prediction (NSP). MLM randomly masks part of the input sentence token and learns to predict the original word of the masked token, while NSP predicts the relationship between two sentences in the corpus. NSP, or Next Sentence Prediction, is a technique that considers the relationship between two sentences and determines whether they are related through fine-tuning tasks such as natural language inference and question answering. This is achieved by concatenating two sentences from a corpus and performing binary NSP to determine whether the second sentence immediately follows the first sentence in the original corpus. This allows the model to identify context and learn the relationships between the sentences. In order to adapt the pretrained BERT model to the target task, we added either a FC layer or an BiLSTM-CRF layer. Our approach involves training these two models and performing a fine-tuning process using new data for the keyphrase extraction task(see Fig.1).)

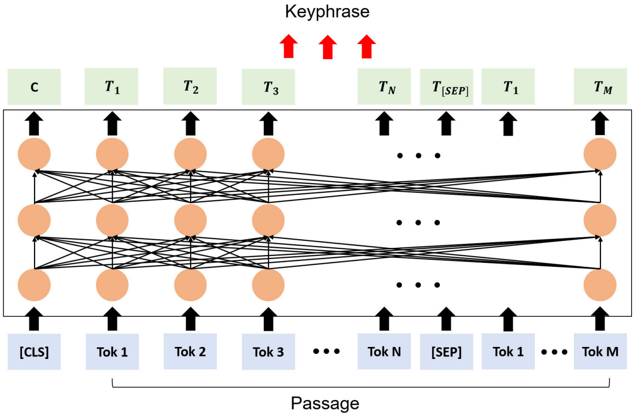


**FIGURE 1.** An overall structure of BERT.

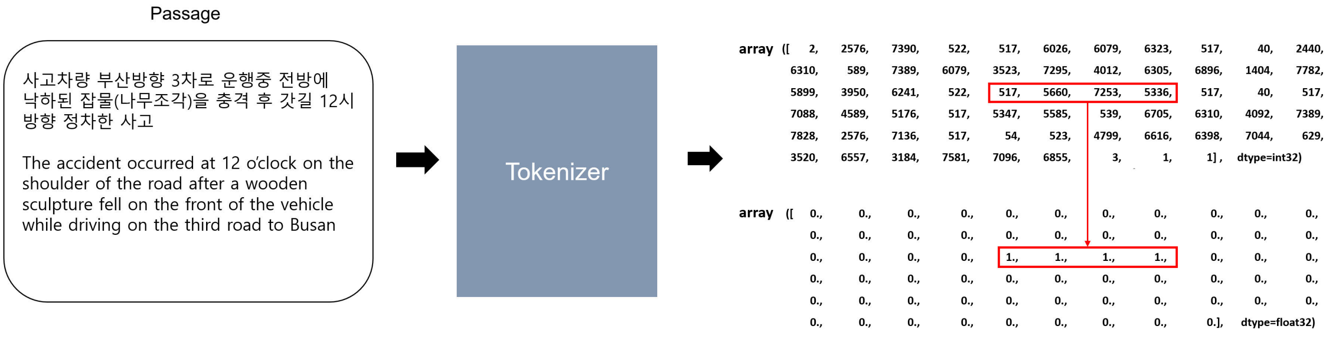
#### 2) ELECTRA

The pre-training methods of MLM, such as BERT, have limitations. During the training process, BERT only uses 15% of the masked tokens among all input tokens. MLM learns by masking a small portion (15% in the case of BERT) of the original input tokens and reconstructing the masked tokens. As a result, data efficiency is reduced. MLM methods require significant computational power and corpora to effectively acquire knowledge. ELECTRA, on the other hand, learns using the entire dataset and has been proven to be more computationally efficient with its replaced token detection (RTD) method. RTD performs binary classification by replacing a few tokens with other tokens and determining whether they are original or replaced. In terms of initial learning speed and performance, RTD outperforms the MLM of BERT. ELECTRA has two neural networks, a generator and a discriminator, where the generator replaces masked tokens with generated samples and the discriminator determines whether a token is original or a generated sample. The generator learns to maximize the likelihood of masked tokens, and its optimal size is between 1/4 to 1/2 of the discriminator's size. The ELECTRA-Base model outperformed the BERT-Large and BERT-Base models. We fine-tuned the pre-trained ELECTRA-Base model for the keyphrase extraction task using a method similar to that of BERT.

Fig.2 illustrates the tokenization process of text data and labels for input to the language model. The tokenizer defined by each model is used to tokenize text data, and the tokenized text data vary depending on the corpus size and the tokenization scheme used during the model training.

### C. KEYPHRASE EXTRACTION

In this section, we describe two approaches to keyphrase extraction: the FC layer and the BiLSTM-CRF layer. We approach keyphrase extraction as a sequence labeling



사고차량 부산방향 3차로 운행중 전방에 낙하된 잡물(나무조각)을 충격 후 갓길 12시 방향 정차한 사고

The accident occurred at 12 o'clock on the shoulder of the road after a wooden sculpture fell on the front of the vehicle while driving on the third road to Busan

```
array  ([    2,  2576,  7390,   522,   517,  6026,  6079,  6323,   517,    40,  2440,
          6310,   589,  7389,  6079,  3523,  7295,  4012,  6305,  6896,  1404,  7782,
          5899,  3950,  6241,   522,   517,  5660,  7253,  5336,   517,    40,   517,
          7088,  4589,  5176,   517,  5347,  5585,   539,  6705,  6310,  4092,  7389,
          7828,  2576,  7136,   517,    54,   523,  4799,  6616,  6398,  7044,   629,
          3520,  6557,  3184,  7581,  7096,  6855,     3,     1,     1] , dtype=int32)

array  ([  0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    1.,    1.,    1.,    1.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
           0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.],  dtype=float32)
```

**FIGURE 2.** The process of text tokenization.

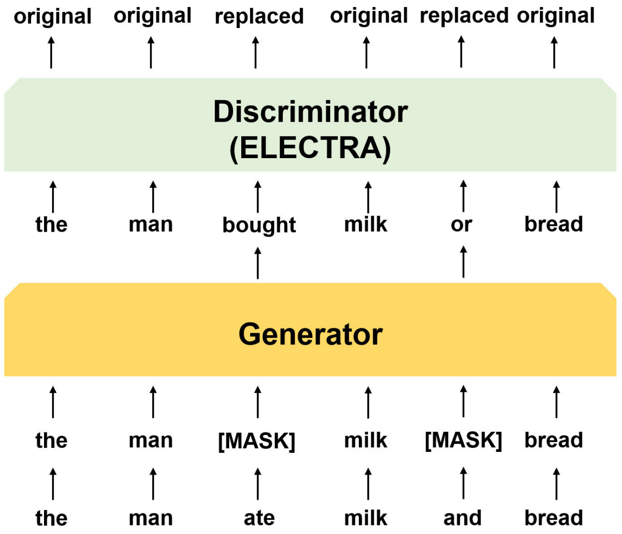


**FIGURE 3.** An overview of the main architecture of ELECTRA.

task. Sequence labeling refers to the classification of sequential data. It takes as input categorical sequence data $x$ of length $n$ and finds the best label sequence $y$ of the same length. Let $x = \{x_1, x_2, \ldots, x_n\}$ be the input or observation sequence and $y = \{y_1, y_2, \ldots, y_n\}$ be the output or label sequence. Sequence labeling is a type of logistic regression where the input is a sequence rather than a single vector. In this paper, we use discriminant models for sequence labeling. Discriminant models are one of the statistical models used in machine learning for classification and prediction tasks.

Maximum-entropy Markov model(MEMM) [41] is a discriminative model that calculates the conditional probability of a label. MEMM performs n classifications on a sequence of length n sequentially. MEMM produces locally biased predictions because the current prediction depends on current observations and previous labels. CRF avoids label bias by normalizing over the entire sequence when performing sequence labeling. CRF considers bidirectional context in their output labels. CRF ensures that label predictions are made within the context of the entire sequence. CRF secures that for all possible label sequences for a given input sequence, the sum of the probabilities assigned to each possible label sequence is at most 1. This is called global normalization, and global normalization considers the probability distribution of all possible sequence labels.

Keyphrase label representation depends on approah to extract the keyphrases. To represent the keyphrase, both methods tokenize and embed the sequence. when performing keyphrase extraction, the input sequence is tokenized to a maximum length of 64. Each token in the sequence then becomes an embedding vector of fixed size with a numerical value. The sequences were tokenized, which also adjusted the length of the sequences to 64. For sequences shorter than that, padding was used to make them fit. The sequence is then embedded by assigning a vocabulay index so that each token with a categorical value has a numerical value, where the vocabulary index is determined by the embedding dimension.

With a FC layer, the model receives each token in the sentence as a vector, and the output layer of the model predicts whether that token is a keyphrase through an activation function. The output layer typically consists of one neuron per label or class in a sequence labeling task. Each neuron produces a score or probability distribution for the label of a given token in the input sequence. For sequences up to 64 in length, the model checks the model output to see if each token is an actual token in the keyphrase. In this case, instead of extracting the vocabulary index of each token classified as a keyphrase, we represent it as an index of its position in the 64-length array of tokenized sentences. The label of using a FC layer is represented as an array of length 3, $s = \{s_{start}, s_{end}, s_{present}\}$, the shape of this array is $(hiddensizemaxlength, 3)$. Element $s_{start}$ in label s indicates the position of the token in the sequence where the keyphrase begins, Element $s_{end}$ indicates the position of the token in the sequence where the keyphrase ends and the last element $s_{present}$ shows the presence or absence of a keyphrase in the sequence. If a sentence mentions a fallen object explicitly, the final element in the array is set to "1". However, if a sentence does not mention or is unclear about a fallen object, the final element in the array is set to "0".

With a BiLSTM-CRF layer, the model can take into account dependencies between predicted keyphrases. The BiLSTM layer reflects the bidirectional context of the input words and the CRF layer reflects the bidirectional context of the output labels. Models with a fc layer, the label is determined after the softmax activation function, but in models with a CRF layer, the result of the activation function is passed to the input of the CRF layer. The features that pass through the BiLSTM cell become the input to the CRF Layer. Sequence labeling is then performed by considering all possible sequences of output labels as they pass through the CRF, the entire label is considered when determining the label of a token through the CRF layer to learn the constraints. The LSTM-CRF layer predicts the highest-scoring sequence over the possible label sequences and outputs the final label by marking tokens that correspond to that keyphrase as 1 and those that do not as 0.

The label of using a BiLSTM-CRF layer is represented by a binary array that sets tokens that are part of the keyphrase to "1" and the rest to "0". If the phrase makes no mention of the keyphrase, falling object, or is unclear, all elements in the array are set to "0".

CRF is a sequential labeling model designed to address label bias, which can convert data in any format into a high-dimensional Boolean vector using potential functions. The potential function is responsible for converting the values input as a sequence into a vector, including categorical values. CRF and MEMM perform similarly, but the main difference lies in the normalization of scores for a given input.

CRF can capture both local and global label dependencies, while MEMM only computes local dependencies based on previous labels. MEMM normalizes locally for each token in a sequence. This leads to the problem of label bias for locally normalized values, since the labeling of the words in the sequence is only optimized at one point in time, rather than being optimal given the words in the sequence. MEMM only considers competition between outgoing path values from one state and not from all other transitions in the model, which is why normalization is locally adapted. MEMM's prediction is overly influenced by the most likely prior labels, and the label bias problem is a consequence of MEMM not considering the probability of the entire sequence when labeling it. In summary, MEMM only considers the previous word and its label to determine the label of the current word. CRF, on the other hand, performs a global normalization, considering all possible transitions and scores for a given input, which helps to overcome the labeling bias. CRF ensures that the sum of the probabilities for all possible label sequences is 1, so that the label probabilities are consistent thanks to the consideration of the entire context, and therefore less prone to label bias. This means that CRF is better at modeling complex, long-range relationships between labels in a sequence. The sequence labeling process $P(y_{1:n}|x_{1:n})$ using CRF is defined as follows:

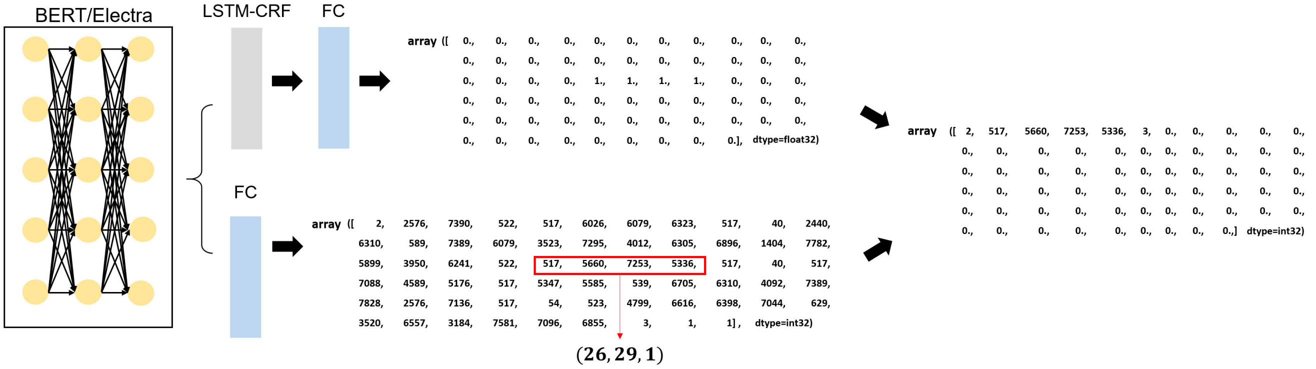
$$P(y|x) = \frac{exp(\sum_{j=1}^{m}\sum_{i=1}^{n}\lambda_j f_j(x, i, y_i, y_{i-1}))}{\sum_y exp(\sum_{j=1}^{m}\sum_{i=1}^{n}\lambda_j f_j(x, i, y_i, y_{i-1}))} \quad (1)$$

The framework for keyphrase extraction involves using two types of NLP models and two methods, with a focus on CRF. CRF performs a single classification operation on each sequence using a coefficient vector $\lambda$ and potential function $f$. To output results of the same length as the input sequence, a BiLSTM-CRF layer is added to pretrained NLP models. The two methods take the same tokenized text data as inputs, but differ in how they are labeled, resulting in different output dimensions. Finally, the output layer uses an input token and vocabulary index to express an output keyphrase.

### D. DATA

In this section, we present a quantitative analysis based on 3,971 reports of domestic highway crash accidents collected by the Ministry of Transportation of the Korea Expressway Corporation (KEC) from 2015 to 2019. The Ministry of Transportation has been constructing a database of fallen objects since May of 2019, and the results of fallen objects collected through this database were also included in the 2019 research data. The accident description include the falling object that caused the accident, the location of the accident, and the degree of vehicle damage, and the falling object that caused the accident was manually identified by label. The maximum length of the accident sequence used in the experiment is 64, and the minimum length is 15. The pre-trained BERT used in this study uses sequence data with a maximum length of 512 as input. Considering this, all sequences were preprocessed into sequences with a maximum length of 64, considering the available learning resources and efficiency. If the maximum length of a sequence was less than 64, the length was adjusted to 64 using padding.

To address the performance limitations of NLP models when processing Korean data, we employed NLP models that were trained on Korean corpora, namely KoBERT and KoELECTRA. These models were chosen for their ability to capture the irregular characteristics of the Korean language. KoBERT was trained on a large corpus of approximately five million Korean sentences collected from Wikipedia and news articles, while KoELECTRA was trained on a corpus of written Korean texts, including Wikipedia articles, news articles, and book plots. Both models have a vocabulary size of 35,000 words.To adapt the NLP models for the task at hand, we fine-tuned them using accident reports from Korean highways between 2015 and 2019. The reports contained 108 variables related to accidents, including textual descriptions, car types, speeds, and time zones. We focused on the text descriptions and categorized them based on whether the fallen object was identified directly or vaguely. Manual labeling was performed, and the corresponding words or phrases were recorded as labels for identified fallen



```
array ([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
         0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
         0.,  0.,  0.,  0.,  1.,  1.,  1.,  1.,  0.,  0.,  0.,
         0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
         0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
         0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.], dtype=float32)
```

```
array ([    2, 2576, 7390,  522,  517, 6026, 6079, 6323,  517,   40, 2440,
         6310,  589, 7389, 6079, 3523, 7295, 4012, 6305, 6896, 1404, 7782,
         5899, 3950, 6241,  522,  517, 5660, 7253, 5336,  517,   40,  517,
         7088, 4589, 5176,  517, 5347, 5585,  539, 6705, 6310, 4092, 7389,
         7828, 2576, 7136,  517,   54,  523, 4799, 6616, 6398, 7044,  629,
         3520, 6557, 3184, 7581, 7096, 6855,    3,    1,    1] , dtype=int32)
```

```
array  ([ 2,  517,  5660,  7253,  5336,  3,  0.,  0.,  0.,  0.,  0.,
          0.,  0.,    0.,    0.,    0.,  0.,  0.,  0.,  0.,  0.,  0.,
          0.,  0.,    0.,    0.,    0.,  0.,  0.,  0.,  0.,  0.,  0.,
          0.,  0.,    0.,    0.,    0.,  0.,  0.,  0.,  0.,  0.,  0.,
          0.,  0.,    0.,    0.,    0.,  0.,  0.,  0.,  0.,  0.,  0.,
          0.,  0.,    0.,    0.,    0.,  0.,  0.,  0.,  0.,] dtype=int32)
```

**FIGURE 4.** An overview of the keyphrase extraction process. Labels vary by keyprhase extraction method. Keyphrase extraction with LSTM-CRF produces output labels as a binary array of the same length as a tokenized sentence of length 64. The binary array is marked as '1' for the presence of tokens of the falling object, otherwise as '0'. The label of keyphrase extraction with the FC layers is simpler. The label is represented as an array of length 3. Each of the first and the second element of an array is a start and end index of a falling object that appears in the tokenized sentence array of length 64. The last element of an array provides information about the presence of a falling object in the sentence. The last element of the array provides information about the presence of falling objects in the sentence. The last part is denoted with a '1' indicating the presence of a falling object and a '0' otherwise.

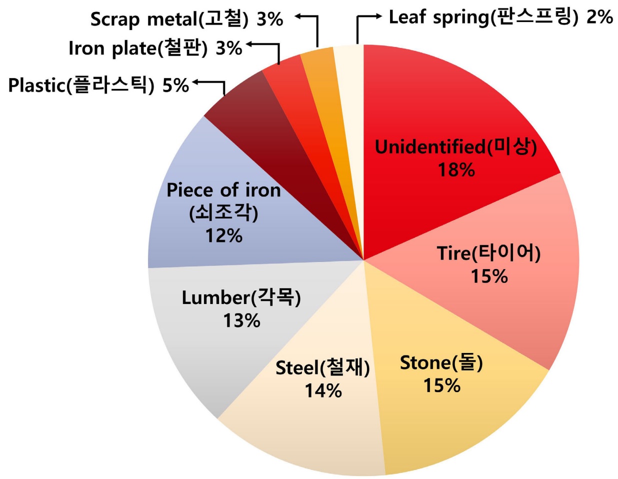


**FIGURE 5.** Chart of the frequency of object causing crashing accidents.

objects, while "unidentified" was used for the remaining cases. In total, 3,971 accidents caused by fallen objects on Korean highways were recorded, with 921, 905, 809, 883, and 453 cases in 2015, 2016, 2017, 2018, and 2019 (until June), respectively. In Fig. 5, the frequencies of fallen objects that appeared on highways more than 10 times are presented. The data shows that there were 492 cases of accidents caused by unidentified falling objects, and 407 cases caused by tires. Additionally, stone, lumber, and steel were recorded in more than 300 cases. The majority of identified fallen objects were heavy vehicle parts or loads dropped from other vehicles during movement, with gravel scattered on the road also being a common cause of accidents. However, in general, the identity of a fallen object often goes unconfirmed due to destruction during the collision or difficulty identifying it at night. Table 1 lists the frequencies of accidents based on vehicle type, with mid-sized cars having the highest number of accidents (1,423), followed by full-sized cars (750) and compact cars (625). Car accidents are more common in passenger cars than in freight vehicles. Fig. 6 presents the speed at which accidents occurred for each vehicle type. The data shows that most vehicles fell within the range of 80 to 110 km/h, which is the maximum speed limit of Korean highways. The average speed by vehicle type is at least 93 km/h for most vehicle types, including mid-sized and compact vehicles. Finally, zero speed refers to the occurrence of an accident when a vehicle is stationary.

**TABLE 1.** Frequency of accidents by vehicle type.

| Vehicle type | Frequency(%) |
|---|---|
| Midsize car | 1,423(0.358) |
| Sport Utility Vehicle(SUV) | 564(0.142) |
| Compact car | 625(0.157) |
| Full-sized car | 750(0.189) |
| Sub-Compact car | 210(0.052) |
| General freight vehicle | 145(0.036) |
| Truck | 23(0.005) |
| Full-sized bus | 13(0.003) |
| Tank Truck | 7(0.002) |
| Unidentified | 211(0.053) |

### E. EXPERIMENTS

In order to fine-tune the pre-trained nlp for the keyphrase extraction task, experiments were conducted on a dataset of 3,971 text samples, divided into 2,779 for the training set and 1,192 for the test set. This partitioning was randomised, with 70% of the total dataset allocated to training and 30% to validation. In addition, we used cross-validation to robustly assess model performance and protect against overfitting. Our approach utilized a combination of two NLP models, namely BERT-Base and ELECTRA-Base, along with two keyphrase extraction methods, namely a FC layer and a BiLSTM-CRF layer. The tokenization results

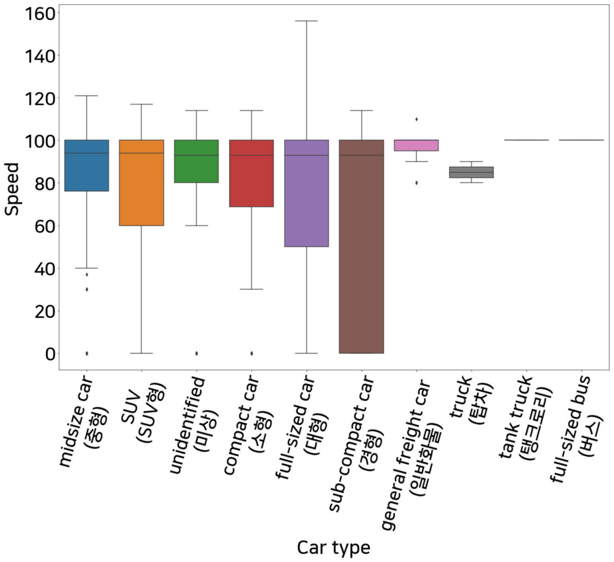


**FIGURE 6.** **Boxplot of vehicle speeds at the time of accidents.**

obtained from the BERT and ELECTRA models differed due to the different sizes of the corpora used to train them. We set the maximum tokenized sequence length to 64 for both models during the experiments. The experiments were carried out by modifying the keyphrase-extraction method for both BERT and ELECTRA models. The first experiment involved adding a FC layer as a keyphrase-extraction method, with the final layer dimensions set at 768 for the pre-trained models and a maximum tokenized sentence length of 64. The labels used in the FC layer for keyphrase extraction had a dimension of three, resulting in a final FC layer with dimensions of $(768 \times 64, 3)$. The output of the model was in the form of a three-dimensional array with the shape $s = (s_{start}, s_{end}, s_{present})$. The first element of the output array represented the starting index of the keyphrase in the tokenized sequence, while the second position represented the ending index of the keyphrase tokens. The third position indicated whether a fallen object was present in the keyphrase. If a fallen object was identified in the keyphrase, the existing label was set to 1, and if a fallen object was not identified or missing, the existing label was set to 0. In cases where the keyphrase did not contain information about a fallen object, the output label was $(0, 0, 0)$. In this study, we only focused on extracting more self-explanatory keyphrases. If the cause of an incident could not be determined from the input data, it was treated as "cause unknown" and no keyphrases were extracted. The label representation for "cause unknown" is (0,0,0) or a 64-length array containing only 0.

In the second experiment, the output labels were generated using a BiLSTM-CRF layer added to the pre-trained models. The output label is a binary array with a length of 64, which is the same as the tokenized sequence. The binary array contains elements of 1 and 0, where an element of the keyphrase in a sequence is marked as 1 and all other elements are marked as 0. It is worth noting that the same tokenized data was used for both experiments.

To summarize, training was performed on a server equipped with an NVIDIA GeForce RTX 2080ti with 11GB of video memory. The batch size was 16 sequences and AdamW optimizer with a learning rate of $5e^{-5}$ was used for experiments. We experimented with several epochs for learning efficiency, with the final number of epochs set to 50 and 20 for the fc layer and BiLSTM-CRF layer, respectively. Transfer learning was applied, and the preloading weights were based on the training results on the SKT-learn dataset. The first experiment used MSELoss as the loss function and an FC layer for keyphrase extraction, while the second experiment used negative log-likelihood as the loss function and a BiLSTM-CRF layer for keyphrase extraction. Both experiments also applied a dropout rate of 0.5. Cross-validation was performed to evaluate model performance on different subsets of the data and check for robustness. We also used various metrics to evaluate the performance of the proposed model, and provided the results for each metric with confidence intervals.

## III. RESULT

The objective of this study was to accurately extract keyphrases from accident reports, where a keyphrase was defined as a fallen object that caused an accident. We evaluated the proposed approach with various metrics, such as accuracy, area under the receiver operator characteristic curve(AUC), F1-score, simple matching coefficient(SMC) [42] and slot error rate(SER) [43]. Among these metrics, AUC, f-score, SMC and SER are the ones that consider false positives and false negatives. We only considered an output vector as correct when it matched the true label exactly.

Accuracy measures the degree to which the extracted keyphrase matches the actual relevant keyphrase. AUC is a single number measure that evaluates the predictive ability of a model, in this case for keyphrase extraction. It quantifies overall performance, with higher values indicating better discrimination between relevant and non-relevant keyphrases. The AUC of 1.0 indicates perfect discrimination, while 0.5 represents random chance. F1-score is a metric that combines precision and recall into a single measure. F1-score is particularly useful for unbalanced data sets and provide an overall measure of model performance, with higher values indicating better performance. The F-score ranges from 0 to 1, with values closer to 1 indicating better performance. F1-score is the harmonic mean of precision and recall. Precision measures the proportion of correctly identified keyphrases out of all predicted keyphrases, while recall measures the proportion of correctly identified keyphrases out of all actual keyphrases. Precision and recall are calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP), as defined in Eq.2. TP indicates the number of cases where the extracted keyphrase matches the label, while FN occurs when the extracted keyphrase does not match the label. FP corresponds to the cases where the model incorrectly identifies that the extracted keyphrase does not match the label. F1-score takes into account false positives and false negatives when both types of error have different costs.

SER can be derived from the F1-score, where 1 - F1 represents the error rate (E), indicating the proportion of cases in which the model fails to accurately identify the keyphrase. When the F1-score is expressed as the weighted harmonic mean of recall and precision, the most popular $\alpha$ value is 0.5 [44]. When alpha is between 0 and 1, the weights of FN and FP are reduced in the value of the F1-score, which reduces the total error rate of the model. The F1-score is defined in Eq.3.

$$\text{Precision(P)} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall(R)} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)$$
$$\text{F1-score} = \frac{\alpha}{P} + \frac{1-\alpha}{R}$$
$$= \frac{PR}{(1-\alpha)P + \alpha R}, \quad 0 \leq \alpha \leq 1.$$
$$\text{F1-score} = 2 \times \frac{P \times R}{P + R}, \quad \alpha = 0.5. \quad (3)$$

SMC is a statistical measure for quantifying the similarity between binary data samples. This metric measures the correlated similarity between the extracted keyphrase and the actual relevant keyphrase. It is calculated by considering both common elements (true positives) and differences (false positives and false negatives) between the extracted keyphrase and the correct response phrase. It is defined as the ratio of the number of matching keyphrase to the total number of keyphrase present. SMC measures the similarity and diversity between the output and label, without requiring the definition of true negatives(TN). SER is good for slot-filling tasks such as keyphrase extraction. It evaluates the overall slot-filling error rate by considering false positives (incorrect keyphrases), false negatives (missing keyphrases) and true positives (correct keyphrases). SER introduces new parameters to measure errors and addresses the problem of de-weighting. It quantifies the cost of errors generated by the system and considers the word error rate, commonly used in speech recognition. Notably, under certain conditions, SER can be greater than 1. We defined SMC and SER in Equation Eq.4.

$$\text{SMC} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}$$
$$\text{SER} = \frac{\text{S} + \text{D} + \text{I}}{\text{N}} \quad (4)$$

In this study, we used various metrics to evaluate the performance of the keyphrase extraction model. The evaluation metrics included **N**, **S**, **D**, and **I**, where **N** represented the number of slots in the label, **S** represented the number of substitutions, **D** represented the number of incorrectly scored keyphrase extractions, and **I** represented the number of correctly scored keyphrase extractions. In our experiment, both the predictions of the model and actual relevant keyphrase labels had 64 slots, which matched the maximum length of the text dataset. While **I** and **D** can be interpreted as FP and FN, respectively, these substitutions were not directly relevant to our study. The CRF methodology we used only considered binary classification of 0s and 1s for each token in a text description, and term substitutes were undefined. However, SER can be represented by the F1-score. We found that the error $1 - F$ was approximately 30% lower than the SER metric, indicating that SER is approximately 1.5 times the error rate [43].

$$1 - \text{F} \cong 0.7 \times \text{SER}$$
$$\text{SER} \cong 1.5 \times 1 - \text{F} \quad (5)$$

All of the experiments in this study were cross-validated, and we used bootstrapping to compute confidence intervals for all of the metrics in this paper. Bootstrapping generates multiple resampled datasets by randomly drawing data points from the original data set with replacement to make inferences about the population from which the original sample. We used bootstrapping to compute confidence intervals by randomly drawing 500 samples from test dataset. The confidence interval for each value in the metric is shown in parentheses. The results are presented in Table 2.

We observed that ELECTRA with a BiLSTM-CRF layer achieved the highest accuracy, AUC, F1-score, SMC, and SER values of 0.943, 0.991, 0.963, 0.929 and 0.075 respectively. We found that the model with the BiLSTM-CRF layer outperformed the model with the simple FC layer when we compared the best model with other models. Overall, our proposed approach using ELECTRA with BiLSTM-CRF demonstrated superior performance in terms of accuracy, AUC, F1-score, SMC and SER, indicating an improvement in the validity of keyphrase extraction. The key to BiLSTM-CRF's keyphrase extraction is to assign a label to each token in the input sequence, indicating whether it belongs to a keyphrase. The BiLSTM-CRF layer takes into account all relationships between words in the sequence.

In this study, we utilized pre-trained BERT and ELECTRA models to analyze Korean accident report data for keyphrase extraction, with the aid of BiLSTM-CRF. It was observed that BiLSTM-CRF played a crucial role in capturing word relationships within text sequences for this task. However, it is important to note that there were variations in the amount of textual data used for pre-training these two models, as well as their vocabulary sizes. Specifically, ELECTRA had a vocabulary size of 35,000 while BERT had a vocabulary size of 8,002, owing to its larger pretraining corpus. As a result, the overall metrics were better for ELECTRA regardless of the keyphrase extraction method, highlighting the importance of proper training data and model architecture for optimal performance.

**TABLE 2.** Experimental result with Confidence Intervals [95% CI].

| Model | KP method | Accuracy | AUC | F1-score | SMC | SER |
|---|---|---|---|---|---|---|
| BERT | FC | 0.204[0.179, 0.230] | 0.655[0.64, 0.673] | 0.392[0.357, 0.425] | 0.204[0.182, 0.229] | 2.100[1.566, 2.689] |
| | BiLSTM-CRF | 0.931[0.914, 0.946] | 0.981[0.974, 0.988] | 0.963[0.953, 0.972] | 0.929[0.910, 0.945] | 0.074[0.056, 0.095] |
| ELECTRA | FC | 0.663[0.632, 0.693] | 0.745[0.719, 0.77] | 0.652[0.61, 0.696] | 0.376[0.335, 0.419] | 1.042[0.893, 1.223] |
| | BiLSTM-CRF | 0.943[0.927, 0.957] | 0.991[0.985, 0.995] | 0.963[0.951, 0.972] | 0.929[0.907, 0.946] | 0.075[0.056, 0.101] |

## IV. DISCUSSION

In this study, we present a method for analyzing textual accident reports to determine the cause of accidents caused by falling objects. Our approach utilizes NLP models to automatically extract keyphrases from accident reports, aiding in the recognition and elimination of different causes of accidents. The main contributions of our study are twofold. First, we apply NLP-based models combined with keyphrase-extraction methods to accident reports. Second, we utilize multiple fine-tuning models to analyze and uncover the main contributors to accidents in accident reports, providing valuable insights for managing and preventing such accidents. Moreover, we demonstrate that combining BiLSTM-CRF methodologies with fine-tuned models leads to enhanced performance in keyphrase extraction. Although this study used a relatively small dataset, cross-validation was used to confirm the generalis ability of the model and transfer learning was used to check the scalability of the results using a pre-trained NLP model.

The quality of the extracted keyphrases is affected by the quality of the pre-trained model, as it depends on the ability to capture complex semantic relationships to gain contextual information within sentences and paragraphs of the incident report. This means that a pre-trained model may produce biased keyphrases or miss important aspects of a thought, depending on the data it is pre-trained on. Therefore, it is important to consider the model quality and model bias to improve the accuracy and reliability of the keyphrase extraction process. The performance of the proposed method can be improved by using pre-trained NLP models in domains specific to incident reports, or by using NLP models with more efficient pre-training methods.

Although, the proposed approach has the following limitations. Since this study focused on accidents caused by falling objects, the extracted keyphrases were mainly nouns. However, we found that even less frequent falling objects can be extracted well through experiments. Therefore, it can be said that this approach can effectively adapt to new words that did not appear at all in the learning process by identifying the relationship between certain parts of speech and certain words. Similarly, we expect it to be effective for extracting keyphrases containing different parts of speech that did not appear in the learning process.

Also, If the size of the pre-trained NLP model is small, there may be difficulties in extracting words or other parts of speech that are not learned in the pre-training. This can be solved by training an NLP model on a larger corpus and learning different types of sentences. In addition, the study was limited to accident reports written in Korean, and therefore the models used were limited to those trained on Korean corpora.

## V. CONCLUSION

In conclusion, we present an novel approach for keyphrase extraction from incident reports, emphasizing its practical relavance for accident analysis. Unlike traditional methods that mainly focus on document summarization, our study provides a unique approach that customizes keyphrase extraction specifically for identifying falling objects as the primary cause of accidents.

We evaluated the performance of two prominent models, BERT and ELECTRA, integrating FC and BiLSTM-CRF layers for keyphrase extraction. Notably, our study demonstrates the superiority of the ELECTRA-based model using BiLSTM-CRF over alternative models. It achieves accuracy of 0.943, an AUC of 0.991, and a low SER of 0.075. The F1-score and SMC closely mirrored the BERT-based model using BiLSTM-CRF, with no statistically significant differences within the 95% confidence interval.

Our findings underscore the potential of fine-tuning pre-trained natural language processing models for post-hoc analysis of traffic accidents, facilitating the swift identification of key factors. While automated keyphrase extraction serves as an efficient means to quickly identify crucial elements within incident reports, it's essential to recognize its inherent limitations. This approach has innate constraints, potentially overlooking certain contextual nuances and fail to provide a comprehensive incident description. To address these limitations, human expertise remains irreplaceable. Human experts can scrutinize reports, consider broader contexts, and offer a more comprehensive understanding of incidents. Therefore, we advocate a balanced approach that integrates keyword extraction as an initial step in incident analysis, enabling the swift identification of crucial elements, followed by thorough review by human analysts.

The primary practical application of this study is centered on accident prevention. By extracting keywords from the textual descriptions in accident reports can reveal valuable insights into the factors that contribute to accidents. Our fine-tuning approach shows great potential for various applications in the fields of road safety and accident prevention. This information can be leveraged to devise strategies for simplifying compensation procedures, accelerating the identification of accident causes, proactively improving safety measures in high-risk areas, and implementing preemptive measures in locations with persistent traffic congestion to avert accidents and enhance traffic flow. To exemplify the practical benefits, consider a scenario where keywords extracted from incident reports consistently indicate issues such as ‘stones’ in a particular area. This data can facilitate road safety improvements and accident prevention by giving priority to inspections to repair road damage in that section.

In summary, our study proposes a flexible strategy with the potential to enhance road safety and prevent accidents. The combination of automated keyword extraction and human expertise offers a holistic approach to incident analysis, contributing to enhanced road safety and accident prevention.

## ACKNOWLEDGMENT

*(Seung-Seok Lee and So-Mi Cha contributed equally to this work.)*

## REFERENCES

[1] B. C. Tefft, “The prevalence of motor vehicle crashes involving road debris, United States, 2011–2014,” *Age*, vol. 20, no. 7, pp. 1–10, 2016.

[2] California Office of Traffic Safety. (2019). *Highway to Help: Secure your Load Makes Roads Safer With the Steadfast Support of Its Partners in Safety*. Accessed: Jul. 1, 2021. [Online]. Available: https://www.ots.ca.gov/2019/05/23/highway-to-help-secure-your-load-makes-roads-safer-with-the-steadfast-support-of-its-partners-in-safety

[3] F. L. Mannering, V. Shankar, and C. R. Bhat, “Unobserved heterogeneity and the statistical analysis of highway accident data,” *Analytic Methods Accident Res.*, vol. 11, pp. 1–16, Sep. 2016.

[4] H. S. Chase, L. R. Mitrani, G. G. Lu, and D. J. Fulgieri, “Early recognition of multiple sclerosis using natural language processing of the electronic health record,” *BMC Med. Informat. Decis. Making*, vol. 17, no. 1, pp. 1–8, Dec. 2017.

[5] P. Hughes, D. Shipp, M. Figueres-Esteban, and C. van Gulijk, “From free-text to structured safety management: Introduction of a semi-automated classification method of railway hazard reports to elements on a bow-tie diagram,” *Saf. Sci.*, vol. 110, pp. 11–19, Dec. 2018.

[6] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal, “Natural language processing for aviation safety reports: From classification to interactive analysis,” *Comput. Ind.*, vol. 78, pp. 80–95, May 2016.

[7] W. Jin, R. K. Srihari, H. H. Ho, and X. Wu, “Improving knowledge discovery in document collections through combining text retrieval and link analysis techniques,” in *Proc. 7th IEEE Int. Conf. Data Mining (ICDM)*, Oct. 2007, pp. 193–202.

[8] J. Cao, K. Zeng, H. Wang, J. Cheng, F. Qiao, D. Wen, and Y. Gao, “Web-based traffic sentiment analysis: Methods and applications,” *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 844–853, Apr. 2014.

[9] Y. Zhao, T.-H. Xu, and W. Hai-Feng, “Text mining based fault diagnosis of vehicle on-board equipment for high speed railway,” in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 900–905.

[10] R. Nayak, N. Piyatrapoomi, and J. Weligamage, “Application of text mining in analysing road crashes for road asset management,” in *Engineering Asset Lifecycle Management*. Cham, Switzerland: Springer, 2010, pp. 49–58.

[11] K. S. Hasan and V. Ng, “Automatic keyphrase extraction: A survey of the state of the art,” in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1262–1273.

[12] T. Tomokiyo and M. Hurst, “A language model approach to keyphrase extraction,” in *Proc. ACL Workshop Multiword Expression Anal., Acquisition Treatment*, 2003, pp. 33–40.

[13] H. P. Luhn, “A statistical approach to mechanized encoding and searching of literary information,” *IBM J. Res. Develop.*, vol. 1, no. 4, pp. 309–317, Oct. 1957.

[14] J. D. Cohen, “Highlights: Language- and domain-independent automatic indexing terms for abstracting,” *J. Amer. Soc. Inf. Sci.*, vol. 46, no. 3, pp. 162–174, Apr. 1995.

[15] G. Salton, C. S. Yang, and C. T. Yu, “A theory of term importance in automatic text analysis,” *J. Amer. Soc. Inf. Sci.*, vol. 26, no. 1, pp. 33–44, Jan. 1975.

[16] Y. Matsuo and M. Ishizuka, “Keyword extraction from a single document using word co-occurrence statistical information,” *Int. J. Artif. Intell. Tools*, vol. 13, no. 01, pp. 157–169, Mar. 2004.

[17] A. Hulth, “Improved automatic keyword extraction given more linguistic knowledge,” in *Proc. Conf. Empirical methods Natural Lang. Process.*, 2003, pp. 216–223.

[18] T. D. Nguyen and M.-Y. Kan, “Keyphrase extraction in scientific publications,” in *Proc. Int. Conf. Asian Digit. Libraries*. Cham, Switzerland: Springer, 2007, pp. 317–326.

[19] P. D. Turney, “Learning algorithms for keyphrase extraction,” *Inf. Retr.*, vol. 2, no. 4, pp. 303–336, May 2000.

[20] S. N. Kim and M.-Y. Kan, “Re-examining automatic keyphrase extraction approaches in scientific articles,” in *Proc. Workshop Multiword Expressions Identificat., Interpretation, Disambiguation Appl.*, 2009, pp. 9–16.

[21] C. Caragea, F. A. Bulgarov, A. Godea, and S. D. Gollapalli, “Citation-enhanced keyphrase extraction from research papers: A supervised approach,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1435–1446.

[22] D. Kelleher and S. Luz, “Automatic hypertext keyphrase detection,” in *Proc. IJCAI*, vol. 5, 2005, pp. 1608–1609.

[23] W.-T. Yih, J. Goodman, and V. R. Carvalho, “Finding advertising keywords on web pages,” in *Proc. 15th Int. Conf. World Wide Web*, May 2006, pp. 213–222.

[24] P. D. Turney, “Coherent keyphrase extraction via web mining,” 2003, *arXiv:cs/0308033*.

[25] P. D. Turney, “Learning to extract keyphrases from text,” 2002, *arXiv:cs/0212013*.

[26] A. Hulth, J. Karlgren, A. Jonsson, H. Boström, and L. Asker, “Automatic keyword extraction using domain knowledge,” in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Cham, Switzerland: Springer, 2001, pp. 472–482.

[27] K. Zhang, H. Xu, J. Tang, and J. Li, “Keyword extraction using support vector machine,” in *Proc. Int. Conf. Web-Age Inf. Manage.* Cham, Switzerland: Springer, 2006, pp. 85–96.

[28] X. Jiang, Y. Hu, and H. Li, “A ranking approach to keyphrase extraction,” in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2009, pp. 756–757.

[29] P. Lopez and L. Romary, “HUMB: Automatic key term extraction from scientific articles in Grobid,” in *Proc. SemEval Workshop*, 2010, p. 4.

[30] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Proc. 18th Int. Conf. Mach. Learn. (ICML)*. San Francisco, CA, USA: Morgan Kaufmann, 2001, pp. 282–289.

[31] C. Zhang, “Automatic keyword extraction from documents using conditional random fields,” *J. Comput. Inf. Syst.*, vol. 4, no. 3, pp. 1169–1180, 2008.

[32] S. D. Gollapalli and X.-L. Li, “Keyphrase extraction using sequential labeling,” 2016, *arXiv:1608.00329*.

[33] R. Alzaidy, C. Caragea, and C. L. Giles, “Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents,” in *Proc. World Wide Web Conf.*, May 2019, pp. 2551–2557.

[34] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, “Deep contextualized word representations,” 2018, *arXiv:1802.05365*.

[35] A. Radford et al., “Improving language understanding by generative pre-training,” OpenAi, San Francisco, CA, USA, 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” 2018, *arXiv:1810.04805*.

[37] J. Howard and S. Ruder, “Universal language model fine-tuning for text classification,” 2018, *arXiv:1801.06146*.

[38] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training text encoders as discriminators rather than generators,” 2020, *arXiv:2003.10555*.

[39] SKTBrain. (2019). *KoBERT: Korean BERT Pre-Trained Cased*. [Online]. Available: https://github.com/SKTBrain/KoBERT

[40] Jangwon Park. (2020). *Koelectra: Pretrained Electra Model for Korean*. [Online]. Available: https://github.com/monologg/KoELECTRA

[41] A. McCallum, D. Freitag, and F. C. N. Pereira, “Maximum entropy Markov models for information extraction and segmentation,” in *Proc. ICML*, vol. 17, 2000, pp. 591–598.

[42] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Amsterdam, The Netherlands: Elsevier, 2012.

[43] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, ''Performance measures for information extraction,'' in *Proc. DARPA Broadcast News Workshop*, Herndon, VA, USA, 1999, pp. 249–252.

[44] N. Chinchor, ''Four scorers and seven years ago: The scoring method for MUC-6,'' in *Proc. 6th Conf. Message Understand.*, Held Columbia, MD, USA, 1995, pp. 1–13.

**SEUNG-SEOK LEE** received the B.S. degree in statistics from Chonnam National University, South Korea, in 2020, where he is currently pursuing the master's and Ph.D. degrees. His research interests include machine learning algorithm, artificial intelligence methodologies, natural language process, and financial time series analysis.

**SO-MI CHA** received the B.S. and M.S. degrees in statistics from Chonnam National University, South Korea, in 2017 and 2019, respectively, where she is currently pursuing the Ph.D. degree. Her research interests include machine learning algorithm, artificial intelligence methodologies, computer-aided diagnosis, and financial time series analysis.

**BONGGYUN KO** received the B.S. and M.S. degrees in mathematical science from the Korea Advanced Institute of Science and Technology, South Korea, in 2013, and the Ph.D. degree in industrial engineering from Seoul National University, South Korea, in 2016. From 2016 to 2018, he was a Senior Professional with the Big Data Analytics Group of Mobile Communications Business in Samsung Electronics. He was a Senior Data Scientist with the Hana Institute of Technology, Hana TI, South Korea, in 2018. He is currently an Associate Professor with the Department of Statistics, Chonnam National University, South Korea. His research interests include machine learning algorithm, financial time series analysis, and risk management.

**JE JIN PARK** received the B.S. degree in civil engineering from Chosun University, South Korea, and the M.S. and Ph.D. degrees in civil engineering from Chonnam National University, South Korea, in 1999 and 2003, respectively. He is currently an Associate Professor with the Department of Civil Engineering, Chonnam National University. His research interests include road engineering, urban disaster prevention engineering, autonomous cooperative driving-based road safety engineering, and road capacity and road maintenance based on the Fourth Industrial Revolution.