International Journal of Transportation Science and Technology 12 (2023) 1038–1051

Contents lists available at ScienceDirect

International Journal of Transportation Science and Technology

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

Application of text mining techniques to identify actual wrong-way driving (WWD) crashes in police reports

Parisa Hosseini [a], Seyedalireza Khoshsirat [b], Mohammad Jalayer [a,*], Subasish Das [c], Huaguo Zhou [d]

[a] Department of Civil and Environmental Engineering, Center for Research and Education in Advanced Transportation Engineering Systems (CREATEs), Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA
[b] Department of Computer & Information Sciences, University of Delaware, 18 Amstel Ave, Newark, DE 19716, USA
[c] Civil Engineering Program, Ingram School of Engineering, Texas State University, RFM 5202, 601 University Drive, San Marcos, TX 78666, USA
[d] 224 Harbert Center, Department of Civil and Environmental Engineering, Auburn University, Auburn, AL 36849, USA

ARTICLE INFO

Article history:
Received 20 September 2021
Received in revised form 15 November 2022
Accepted 22 December 2022
Available online 28 December 2022

Keywords:
Wrong-Way Driving Crashes
Crash Report Narratives
Text Mining
Text Classification

ABSTRACT

Wrong-way driving (WWD) has been a long-lasting issue for transportation agencies and law enforcement, since it causes pivotal threats to road users. Notwithstanding being rare, crashes occurring due to WWD are more severe than other types of crashes. In order to analyze WWD crashes, there is a need to obtain WWD incidents or crash data. However, it is time-consuming to identify actual WWD crashes from potential WWD crashes in large crash databases. It often involves large man-hours to review hardcopy of crash narratives in the police reports. Otherwise, it may cause an overestimation or underestimation of WWD crash frequencies. To fill this gap, the present study, as the first-of-its-kind, aims at identifying actual WWD crashes from potential WWD crashes in police reports by using machine learning methods. Recently, Bidirectional Encoder Representations from Transformers (BERT) models have shown promising results in natural language processing. In this study, we implemented the BERT model as well as five conventional classification algorithms, including Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Single Layer Perceptron (SLP) to classify crash report narratives as actual WWD and non-WWD crashes. Cross-validation and different performance metrics were used to evaluate the performance of each classification algorithm. Results indicated that the BERT model outperformed in identifying actual WWD crashes in comparison with other algorithms with an accuracy of 81.59%. The BERT classification algorithm can be implemented to reduce the time needed to identify actual WWD crashes from crash report narratives.

© 2022 Tongji University and Tongji University Press. Publishing Services by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

In recent decades, wrong-way driving (WWD) has been perceived as troublesome for the safety of road users. Crashes as a result of WWD are mainly recognized for their severity rather than their frequency. Although being rare in terms of fre-

Peer review under responsibility of Tongji University and Tongji University Press.
* Corresponding author.
E-mail address: jalayer@rowan.edu (M. Jalayer).

https://doi.org/10.1016/j.ijtst.2022.12.002
2046-0430/© 2022 Tongji University and Tongji University Press. Publishing Services by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

quency in comparison with other crashes, WWD crashes typically result in a higher likelihood of severe injuries and fatalities. According to Federal Highway Administration (FHWA), 300 to 400 people are being killed as a result of WWD crashes per year in the United States (FHWA, 2021). Moreover, based on National Highway Traffic Safety Administration's (NHTSA's) Fatality Analysis Reporting System (FARS) (NHTSA, 2019), a total of 131 902 fatal motor vehicle crashes were registered in the United States from 2014 to 2017. During the same period, by applying filter criteria for driver-related factors (50 = "Driv ing Wrong Way on One-Way Trafficway" and 51 = "Driving on Wrong Side of Road") on the FARS database, it was observed that 1 323 of those fatal crashes were recorded as WWD crashes. Hence, crash data can provide us with useful information about the traffic conditions which are unsafe for road users. However, one of the main issues in WWD data collection is that there is no direct resource for obtaining the WWD crashes. In order to overcome this issue, a two-step approach can be employed on general crash datasets. In the first step, potential WWD crashes can be extracted from total crashes by using filter criteria on relevant variables of the dataset. In the second step, extracted potential WWD crashes obtained from the first step are verified by obtaining the police crash reports and reviewing the crash narrative. Nevertheless, going through each hard copy of crash narratives is a time-consuming task. Therefore, there is a need to develop a methodology to reduce the time of manually verifying potential WWD crashes. Addressing this issue by applying text classification and machine learning techniques is the main objective of the present research study.

Due to the recent advancements in data science as well as the existence of enormous amounts of text data, text mining has been widely used and addressed in many research fields in recent years. Text mining is the process of extracting useful information from unstructured texts (Das et al., 2016). Unstructured texts are unorganized text data without any predefined format. Hence, some specific preprocessing steps must be taken to prepare the text data usable for computers for further analysis (Dadgar et al., 2016; Rana et al., 2014). Classifiers are one of the main portions of text mining systems. Text classification is perceived as one of the fundamental problems in Natural Language Processing (NLP) with the aim of categorizing a sequence of text into predefined categories. This process can be achieved by incorporating machine learning (ML) techniques. There are many algorithms available for text classification purposes. However, selecting the best classifier requires a comprehensive understanding of each algorithm. In recent years, considerable efforts have been made to propose pre-trained models as an alternative and beneficial approach for text classification as well as other NLP tasks. By incorporating pre-trained models, the training step for a new model will take less time. One of the most recent pre-trained language models is Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), which has accomplished outstanding results in many NLP tasks (Sun et al., 2019).

The main contribution of this study is to employ text classification and machine learning techniques for identifying actual WWD crashes from the potential WWD crashes in police reports. To achieve this goal, a total of 421 crash reports for the states of Alabama and Illinois were obtained and analyzed. Thereafter, by using machine learning methods, crash report narratives were classified as actual WWD and non-WWD crashes. BERT model, as one of the emerging language models, as well as five conventional classifiers including Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Single Layer Perceptron (SLP) were selected in this study. The performance of BERT and the conventional classifiers in identifying actual WWD crashes from non-WWD crashes was evaluated, and the obtained results were compared.

2. Literature review

During recent decades, document classification by employing text mining and machine learning techniques has received a great deal of attention. Many studies have been devoted to investigating the application of these techniques in different fields, such as topic modeling (Das et al., 2016), sentiment analysis (Lin and He, 2009), news retrieving (Hong, 2012), event tracking (Lee, 2012), and content exploration (Duan and Zeng, 2013; Martinez-Romo and Araujo, 2013; Waters and Jamal, 2011).

Colas and Brazdil (Colas and Brazdil, 2006) performed a comparative study on *20newsgroups* dataset by applying SVM, KNN, and NB as text classifiers. They showed that KNN performs the best among other classifiers under the condition that the dataset is suitably preprocessed. This algorithm also showed a well scale-up with the number of documents. Yu and Xu (2008) conducted a study to classify spam emails by implementing four commonly used classification algorithms, including SVM, relevance vector machine (RVM), NB, and Neural network (NN). They found that SVM and RVM are the most suitable classifiers for classifying spam emails. In a study conducted by Wang and Chiang (2011), multi-class documents were classified by employing SVM with membership function. For comparison purposes, other methods, including NB, Bp-MLL, Multi-Label Mixture, and Jaccard Kernel, were used to classify Reuters datasets. It was presented that the classifier suggested in this study for multi-class documents has a better performance compared to other approaches. In order to improve text classification, Hassan et al. (2011) developed SVM and NB classifiers using Wikitology as the knowledge repositories. The results indicated that the NB improved by +28.78% in terms of macro-average f-measure, while SVM improved by +6.36% by incorporating 121 datasets. Based on obtained results, they suggested that NB would be a better selection under the condition that any external enriching is applied. Fernández-Delgado et al. (2014) investigated the performance of different families of classifiers by applying 179 classification algorithms. They concluded that RF classifiers performed the best with an accuracy of 94.1%, followed by SVM with Gaussian kernel and an accuracy of 92.3%. Shahare and Giri (2015), in 2015, employed two classification algorithms, including SVM and Artificial Neural Network (ANN), for categorizing breast cancer records. The authors found that SVM is superior to ANN in classification, with an accuracy of 93%. In 2016, Das et al. (2016) investigated a total of

15 357 TRB published papers by applying text mining techniques. They collected some particular attributes of papers, including publication year, author names and affiliations, title, abstract, and review committee information. Developing an SVM algorithm with various training sets and testing sets ratios for classifying Reuters datasets was the main focus of a study conducted by Fatima et al. (2017). They concluded that the accuracy of the classifier could be improved by increasing the number of training samples. Kowsari et al. (2017), in an attempt to overcome the multi class datasets, developed a Hierarchical Deep Learning for Text classification (HDLTex) approach. They showed that higher accuracies could be achieved by combining deep neural networks (DNN) or convolutional neural networks (CNN) at a lower level and recurrent neural networks (RNN) at a higher level. Another study, carried out in 2019, was advocated to investigate the effect of word restriction on text classification by generating different word domains from the hotel review dataset. The authors implemented four different classification algorithms (SVM, NB, RF, and DT) and evaluated the results using accuracy, classification time, and used memory. They suggested that NB is the best classifier in terms of memory usage and classification time, while SVM performs well in terms of accuracy (Campos et al., 2019).

The BERT model was initially proposed as an emerging language model in a study conducted by Devlin et al. (2019). The authors introduced their framework in two steps as follows: pre-training the BERT model using unlabeled data and fine-tuning the model parameters utilizing labeled data. Based on the results obtained from BERT fine-tuning, it was shown that this model could improve the GLUE score by 7.7% point, SQuAD v1.1 question answering Test F1 by 1.5 points, SQuAD v2.0 Test F1 by 5.1 points, and the MiltiNLI accuracy by 4.6%. By investigating various fine-tuning methods, Sun et al. (2019) suggested a general approach for fine-tuning the BERT model for text classification purposes. Eventually, they applied the suggested approach to eight commonly used text classification datasets. In 2019, Adhikari et al. (2019) provided the first application of the BERT model on text classification. Firstly, they fine-tuned the BERT model by classifying documents using four widely-used datasets. Secondly, they resolved the high computational expense by distilling the BERT model into a simpler model with 30x fewer parameters and 40x faster inference times. In one of the most recent studies conducted in 2020, the superiority of the BERT model application against other classical NLP approaches for text classification was proved by developing four different NLP scenarios. It was demonstrated that the BERT model outperformed the conventional NLP approaches in terms of accuracy (González-Carvajal and Garrido-Merchán, 2020).

In recent years, a number of studies investigated crash report narratives for different purposes, such as identifying a particular type of crash or extracting useful information using different techniques. In 2019, Trueblood et al. (2019) proposed a semi-automated tool in Microsoft Excel for identifying agricultural crashes in police report narratives. They used six years of crash reports from 2010 to 2015 for Louisiana for further analysis. The results showed that the proposed tool could decrease the search space from 6.7% to 59.4% for the narratives which needed to be manually reviewed. More recently, Zhang et al. (2020) conducted research to identify secondary crashes in crash narratives using text mining techniques. They indicated that among several implemented classification models, the logistic regression model performs the best in terms of accuracy. Arteaga et al. (2020) proposed a new method to investigate traffic crash narratives with the purpose of determining contributing factors related to high injury severity levels. Authors combined text mining techniques with their proposed approach, Global Cross-Validation Local Interpretable Model-Agnostic Explanations (GCV-LIME), using heavy vehicle crashes that occurred between 2007 and 2017 in Australia. Based on the obtained results, the authors found strong relations among some expressions, including ‘‘pedestrian,’’ ‘‘cab,’’ ‘‘side-collided,’’ ‘‘collided-head-on,’’ and ‘‘motorcycle.’’ In 2021, Wali et al. (2021) determined the thematic concept in crash narratives using a two-staged hybrid approach. In this study, text mining was initially applied to extract useful information out of narratives. Then, new variables were derived using advanced statistical model formulation in text mining. The authors used the non-motorists non-crossing trespassing injury data from 2006 to 2015. The results indicated that trespassers who talk on the phone, wear headphones, or have confirmed experience of suicide attempts are more likely to be involved in fatal injuries.

Based on the previous studies, classifiers such as NB, SVM, NN, and RF are some of the common conventional algorithms used for text classification purposes. More recently, BERT has also been employed in text mining classification with promising results. The summary of studies on text classification by employing text mining and machine learning techniques is tabulated in Table 1.

3. Methodology

3.1. Document classification process

Classification is the most critical step in text mining analysis, which can be defined as categorizing documents into some predefined classes (Kumbhar and Mali, 2016). Text classification consists of many steps. Fig. 1 illustrates an overview of the classification process employed in this study.

The first step in the text classification process is to collect documents. Documents can be collected in different formats, such as PDF, web format, and HTML. This step is followed by document preprocessing. Preprocessing is performed on collected documents to convert them into an understandable format for the machine. Document preprocessing includes converting the document to word format, removing punctuations, tokenizing, stemming, and removing stop words. Vector representation of the document, also known as indexing, is the third step aiming to convert a document text to a document vector. Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) and Word2Vec (Goldberg and Levy, 2014)

Table 1
Summary of studies on text classification.

#	Authors (Year)	Data	Methods
1	Colas and Brazdil (2006)	20 000 20newgroup dataset	SVM, KNN, and NB
2	Yu and Xu (2008)	6000 emails from *SpamAssassin* and 5000 emails from *Babletext* datasets	NB, NN, SVM, and RVM
3	Wang and Chiang (2011)	Reuter's datasets	SVM, NB, Multi-Label Mixture, Jaccard Kernel, and Bp-MLL
4	Hassan et al. (2011)	20 000 20newgroup dataset	SVM and NB
5	Fernández-Delgado et al. (2014)	121 University of California at Irvine (UCI) dataset	179 Classifiers such as RF, SVM, and NN
6	Shahare and Giri (2015)	683 Breast Cancer records from UCI dataset	SVM and NB
7	Fatima et al. (2017)	Reuter's datasets	SVM
8	Kowsari et al. (2017)	46 985 published papers from the Web of Science	Hierarchical Deep Learning for Text classification (HDLTex)
9	Campos et al. (2019)	515 000 Hotel Reviews Data	SVM, NB, RF, and DT
10	Sun et al. (2019)	IMDB, Yelp, TREC, AG's News, and DBPedia	BERT
11	Adhikari et al. (2019)	Reuters-21578, arXiv Academic Paper dataset (AAPD), IMDB, and Yelp 2014 Datasets	BERT
12	Trueblood et al. (2019)	Louisiana crash reports	Logistic Regression, RF, NB, and SVM
13	González-Carvajal and Garrido-Merchán (2020)	IMDB, RealOrNot tweets Portuguese news, and Chinese hotel reviews	BERT and NLP approaches (such as Voting Classifier, Multinomial NB, Ridge Classifier, and Passive-Aggressive Classifier)

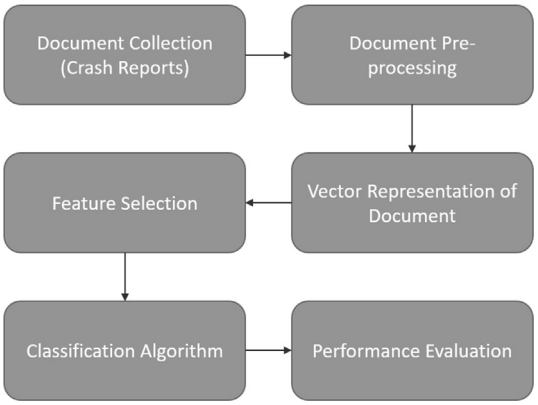

Fig. 1. Overview of the text classification process and its different steps.

are two example algorithms for obtaining vector representations for words. Feature selection, as the fourth step, is the process of selecting a discriminative subset of features. Term Frequency (TF) (Salton and Buckley, 1998) and Term Frequency-Inverse Document Frequency (TF-IDF) are some of the commonly used feature selection methods. The next step is to classify the documents into predefined classes by incorporating a classification algorithm. There are several broadly used algorithms for document categorization, some of which will be outlined in the following subsection. The final step is to evaluate the performance of the proposed classifiers. There are a number of commonly used evaluation metrics, which can be calculated in accordance with a "confusion matrix" (Kowsari et al., 2019). In this study, we considered accuracy, precision, and F-1 score as the metrics to evaluate the performance of the proposed classifiers. The definition of these evaluation metrics is provided as follows (Abikoye et al., 2018):

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \tag{1}$$

$$Precision = TP/(TP + FN) * 100 \tag{2}$$

$$F - 1score = (2 * Recall * Precision)/(Recall + Precision) * 100 \tag{3}$$

where

TP: Documents are positive and correctly assigned to the positive category.
TN: Documents are negative and correctly assigned to the negative category.
FP: Documents are negative, but wrongly assigned to the positive category.
FN: Documents are positive but wrongly assigned to the negative category.

We also collected the classification time and memory of each classifier for result comparison purposes. Classification time can be defined as the time between the start of the classification and the end. Moreover, memory used refers to the memory

used in the classification process. The time and memory usage reported in this study are for one sample in inference mode. It should be noted that the classification time also includes the preprocessing time of the input sample.

3.2. Classification algorithms

Several different classifiers can be utilized for document classification purposes. In the following subsection, five conventional classification algorithms, as well as one of the most recent pre-trained language models, will be discussed.

3.2.1. Naïve Bayes (NB)

The NB algorithm is known as one of the simple classifiers extensively used in the document classification field. This classifier is formulated based on the Bayes theorem. Independency of a feature from other features, under the condition that class variable is given, is a preliminary hypothesis in this method. Assuming that d denotes document, n denotes the number of documents labeled into m classes, and m is a subset of $B = \{b_1, b_2, b_3, \cdots, b_k\}$, the predicted class denoting with b is also a subset of B. Hence, the NB classifier can be formulated as (Kumbhar and Mali, 2016):

$$P(b|d) = \frac{P(d|b)P(b)}{P(d)} \tag{4}$$

$$\begin{aligned} B_{MAP} &= arg\max_{b\in B} P(d|b)P(b) \\ &= arg\max_{b\in B} P(x_1, x_2, \cdots, x_n|b)P(b) \end{aligned} \tag{5}$$

3.2.2. Support Vector Machine (SVM)

This method is a supervised machine learning approach initially introduced by Vapnik and Chervonenkis in 1964 (Vapnik and Chervonenkis, 1964). SVM generally was developed for classifications with two outputs. Nevertheless, some research studies applied this method for multi-class classifications (Kowsari et al., 2019). The main idea of this method is to seek a hyperplane separating two classes having the maximum distance ξ, also known as margin. Support vectors refer to documents locating with ξ from the hyperplane. SVM is independent of the dimensionality of the dataset. This method can be utilized for datasets with high dimensions. A study conducted by Joachims (Joachims, 1998) introduced SVM as a promising algorithm for document classification (Allahyari et al., 2017).

3.2.3. Decision Tree (DT)

This method benefits from trees for predicting the class. The classification process starts from the root node in the tree. Each tree can be a labeled leaf node or a structure with a test node that is connected to some subtrees. In each step of classification, if the node is tested, the outcome is obtained, and the classification continues through the appropriate subtree. Finally, when the process gets to the leaf, the leaf label is considered as the predicted class for that process (Campos et al., 2019).

3.2.4. Random forest

The history of random forest (RF), also known as the decision forest technique, can be traced back to the 90s. This method was developed by Ho (1995) in 1995 for the first time and expanded by Breiman (2001) in 2001. RF establishes several individual random decision trees performing as an ensemble. The generalization error of this method directly relates to the generated trees and their associations (Onan et al., 2016). The larger the number of generated trees, the more the error goes to a converged limit (So et al., 2019).

3.2.5. Neural Network (NN)

The NN algorithm is a natural inspiration for biological neural networks imitating the human brain learning process. A neural network contains layers of interconnected nodes. Each node is a perceptron and is similar to multiple linear regression.

3.2.5.1. Single Layer Perceptron (SLP). An SLP algorithm is a feed-forward network based on a threshold transfer function. This classifier uses documents as input nodes and assigns feature weights to its input. The single-layer perceptron does not have a priori knowledge, so the initial weights are assigned randomly. The decision for the final classification is made based on the final values of output nodes. The SLP follows backpropagation in the classification process. Backpropagation means, in case of any misclassification, the error will be propagated back through the neural network, trying to identify each node's portion in causing the error. Then, weights will be updated and modified with the goal of minimizing the error. Being flexible to be applied to complex problems and having the ability to deal with noisy datasets are some of the advantages of this classifier (Kumbhar and Mali, 2016).

3.2.5.2. BERTxxx. BERT, as one of the recent pre-trained language models, has been widely used in several natural language understanding tasks and accomplished amazing results. Having the basis of a multi-layer bidirectional Transformer, BERT is a neural network language model that has been trained on plain text with the goal of predicting the next sentence or a masked word in a sentence (Zheng and Yang, 2019). This model consists of an encoder with 12 Transformer blocks, 768 hid-

den units, and 12 self-attention heads (Sun et al., 2019). An overview of the BERT model structure for text classification is illustrated in Fig. 2. As shown in this figure, [CLS] and [SEP] represent the beginning and end of a sentence in the input. The lexicon encoder assigns a vector of length 768 to each word in the sentence. Then, the Position Embeddings are added to these vectors to help the network perceive the position of each word. The middlebox in the figure represents the transformer encoder which contains the multi-headed self-attention mechanism. This mechanism outputs an encoded state corresponding to each input. C, as one of the outputs shown in the figure, attains the information of all words in the text; hence it is the representation of the whole text. As the final layer in the BERT model, a softmax classifier is used in order to estimate the probability of the label y by using the following equation:

$$p(y_i|C,\theta) = softmax(CV^T) \tag{6}$$

where $y = \{y_1, \cdots, y_l\}$ ($or\ y = \{0,1\}$) represents the two classes of actual WWD and non-WWD, θ is the set of all the network parameters, and $V \in R^{c\times l}$ represents our task-specific trainable parameter matrix such that c is the length of the output vector C, and l is the number of the classes, in our case 2.

In this study, a pre-trained "bert-base-uncased" model was used for classifying the crash narratives. A "bert-base-uncased" model has fewer parameters compared to a "base-large-uncased" model. Moreover, this model is uncased, meaning that the model does not make a difference between lowercase and uppercase letters.

4. Data

In order to achieve this paper's objective, a total of 421 crash reports for Alabama and Illinois were obtained from the Alabama and Illinois statewide crash database. The crash reports consist of different sections, including location and time, driver and vehicle information, victim's/uninjured occupant's information, crash diagram, roadway environment, investigation, truck/bus information, and most importantly, crash narrative. The crash narrative is one of the main sections of the police reports providing critical information on the crash details, vehicle maneuvers, and driver's condition (whether a driver is under the influence of drugs or alcohol). The information in narratives provides a clear understanding of what exactly occurred during a crash, which is helpful for re-constructing crash situations (So et al., 2019). An example of a hard copy of a crash report is shown in Fig. 3. Some of the crash reports were handwritten or scanned versions. A further step was taken to convert handwritten and scanned versions of the reports to typed versions. Thereafter, crash narratives were extracted from crash reports and utilized for further analysis. Each crash narrative, afterward, was manually reviewed and labeled

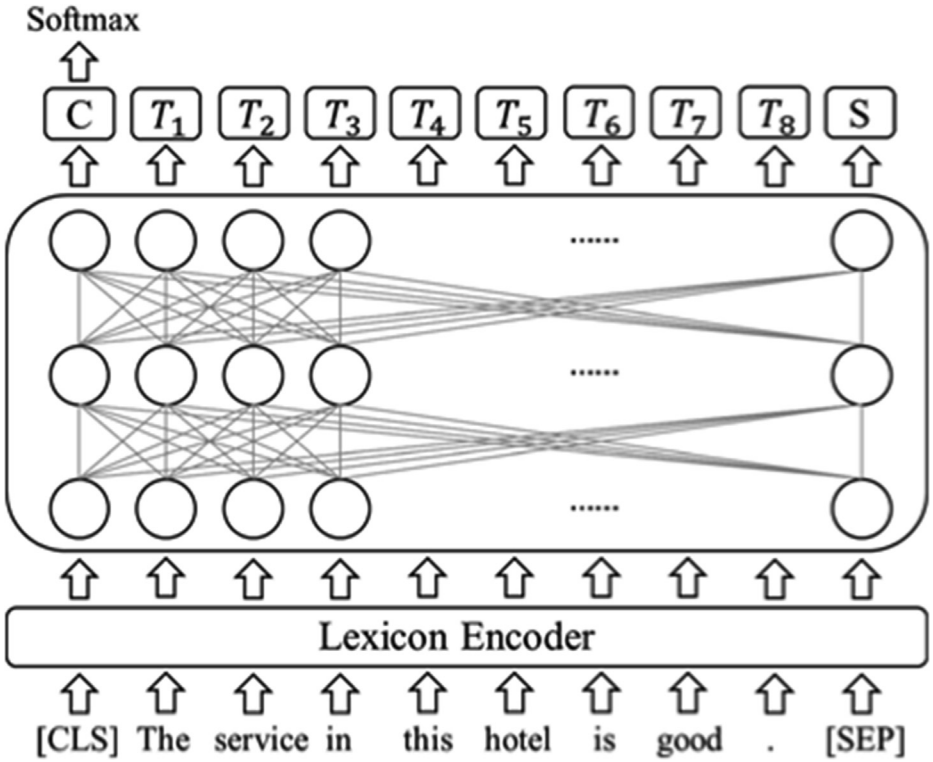

Fig. 2. Overview of BERT model structure (Sun et al., 2019).

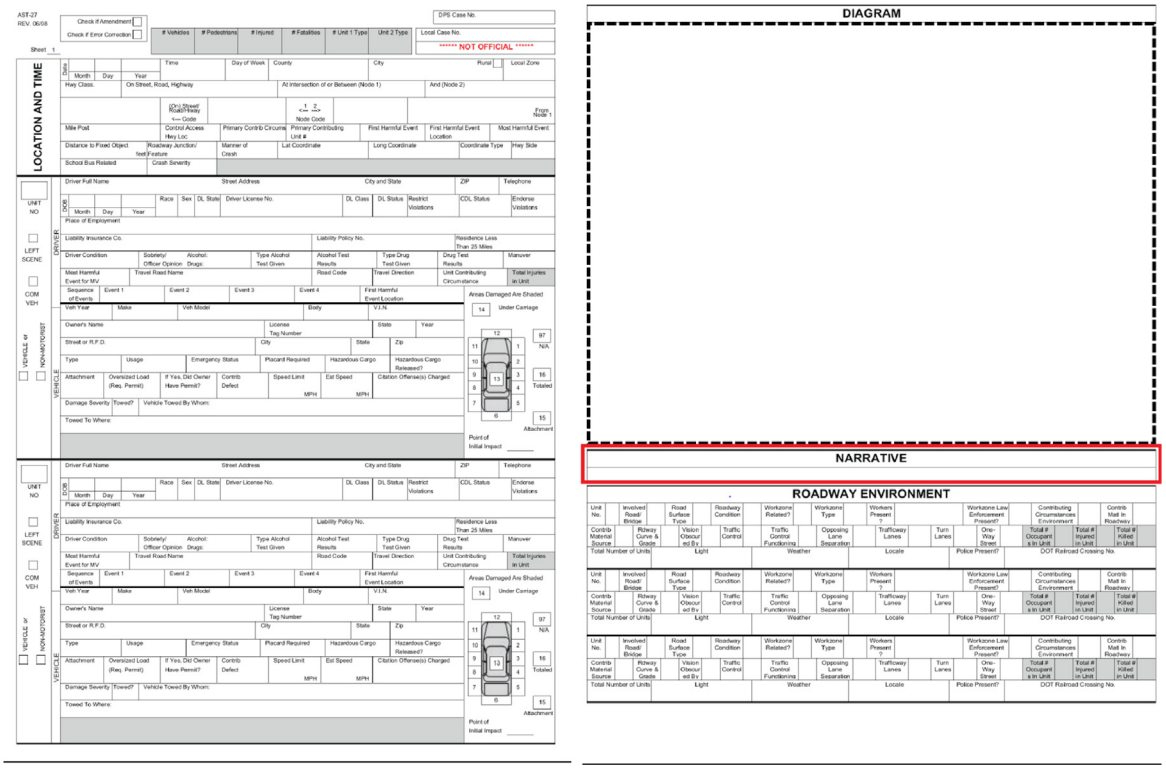

Fig. 3. An example of a crash report (NHTSA, 2020).

as actual WWD crash or non-WWD crash. Out of 421 crash cases, 177 were identified as actual WWD crashes, and the rest were labeled as non-WWD crashes.

5. Implementation details

In this study, all the methods are implemented in Python Version 3.6. The Transformer package (Wolf et al., 2020) is used as the BERT model implementation, and the Scikit-Learn package (Pedregosa et al., 2011) is used for all the other methods. In order to prepare the data in a readable format for the machine, document preprocessing was performed. As the first step in document preprocessing, the text of documents was converted to lowercase. This step will make the text consistent and easier to be compared. Following this, some extra parts, such as numbers, stop words, and punctuations, were removed from the documents. Afterward, the text in the documents was tokenized. Tokenization is the process of breaking the text into meaningful elements, including words and phrases. It is noted to mention that the document preprocessing explained here was only applied to the five conventional classifiers. However, to maximize the use of all the text data for the BERT model, the document preprocessing steps were skipped. An example of a crash report narrative before and after text preprocessing is provided as follows:

Crash Narrative Before Text Preprocessing:

"V1 WAS TRAVELING NORTH ON I459 IN THE SOUTHBOUND MIDDLE LANE OF TRAVEL. ONE SEMI TRUCK TRAVELING SOUTH ON I459 IN THE MIDDLE LANE SWERVED TO THE RIGHT LANE TO AVOID CONTACT WITH V1. ANOTHER NON-CONTACT VEHICLE IN THE LEFT LANE SWERVED TO THE LEFT TO AVOID CONTACT WITH V1. V2 WAS UNABLE TO TAKE EVASIVE ACTIONS. V1 STRUCK V2. V1 CAME TO FINAL REST UPON IMPACT WITH V2 IN THE ROADWAY. V2 LEFT THE ROADWAY TO THE LEFT AND STRUCK THE CONCRETE MEDIAN WALL BEFORE COMING TO FINAL REST. THREE WITNESSES CONFIRMED THAT THE TRAFFIC CRASH HAPPENED AS STATED ABOVE. V2 WAS CARRYING A FOAM SUBSTANCE THAT WAS NOT HAZARDOUS. EMA WAS CALLED DUE TO A FUEL LEAK FROM V2. ALDOT WAS ALSO NOTIFIED. THE DRIVER OF V1 ADMITTED THAT HE HAD 1 ALCOHOLIC DRINK PRIOR TO THE TRAFFIC CRASH."

Crash Narrative After Text Preprocessing:

"able, action, admitted, alcoholic, aldot, avoid, call, came, carry, come, concrete, confirm, crash, drink, drive, ema, evasive, final, foam, fuel, happen, hazardous, impact, lane, leak, left, median, middle, non, north, notify, prior, re, right, roadway, semi, sou, southbound, state, struck, substance, swerve, tact, traffic, travel, truck, v, vehicle, wall, witness"

Frequency analysis for identifying the most frequently used terms is one of the common steps in text mining. Performing frequency analysis can provide a deeper understanding of the most frequent terms cited by police officers in crash report narratives. Word cloud is one of the useful tools for visualizing frequency analysis results. Fig. 4 illustrates the word cloud indicating the frequency of the shared term among both actual WWD and non-WWD crash narratives. As can be seen in Fig. 3, the terms "unit," "driver," "vehicle," "traveling," and "stated" are the top five frequently used terms in police crash report narratives.

6. Results

Five conventional classifiers, as well as the BERT model, were employed in order to identify actual WWD crashes from potential WWD crashes in crash report narratives. Accuracy, precision, F-1 score, classification time, and memory were selected as the performance metrics for conducting k-fold cross-validation for evaluating the classifiers' performance. In this study, five folds were considered for cross-validation with four repeats. Moreover, the analyses were run on the Intel Xeon 2.30 GHz CPU. Detailed descriptions of considered parameters and the obtained results for each classifier are discussed in the following subsections:

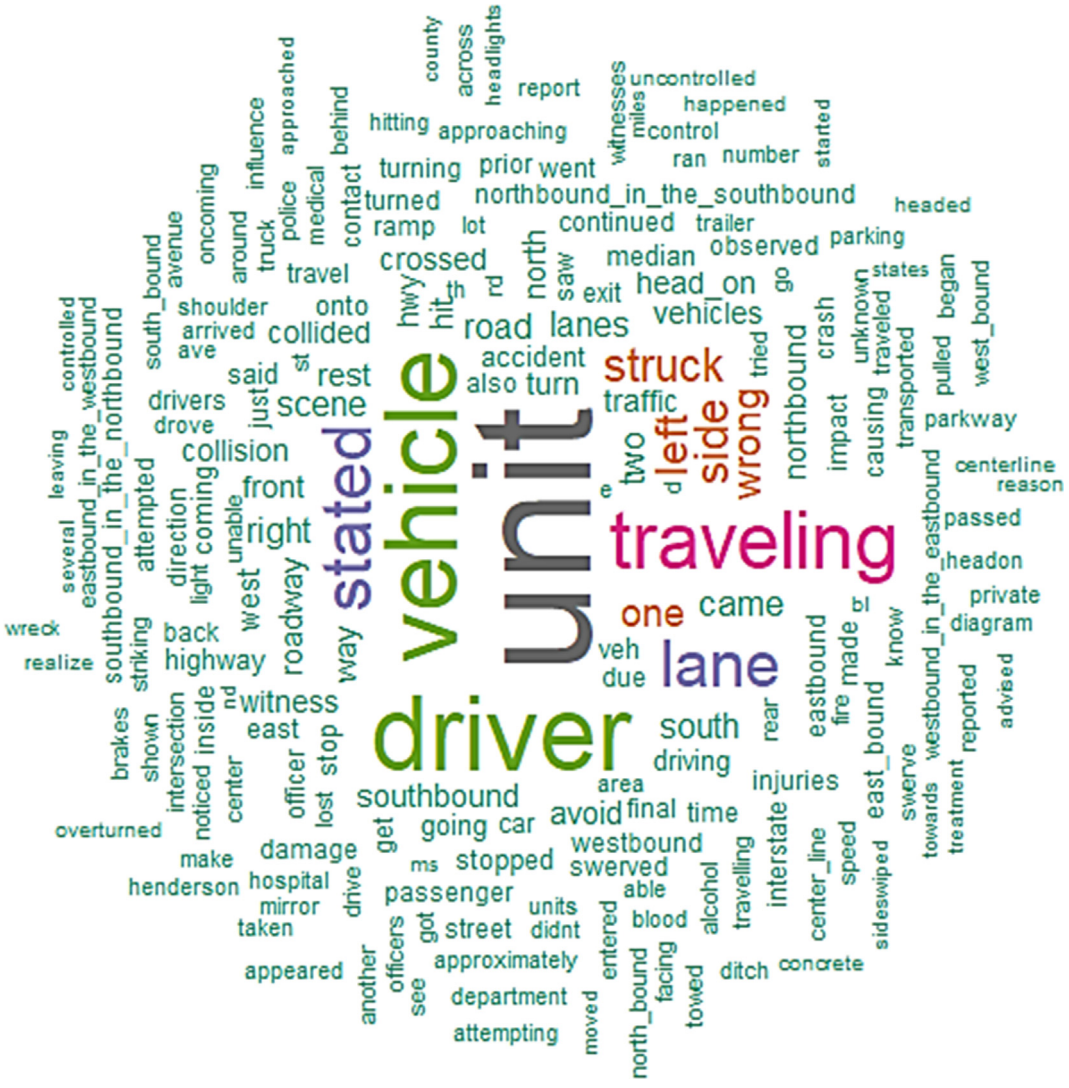

Fig. 4. Word clouds of shared terms cited in police crash report narratives.

6.1. Naïve Bayes (NB)

For classifying the crash report narratives using the NB algorithm, two different variants were tested: Gaussian Naive Bayes and Multinomial Naive Bayes. Table 2 lists the results obtained for the two methods by using five folds for cross-validation with four repeats. The results showed that the Multinomial Naive Bayes performed better, with 73.51% accuracy, 74.58% precision, and 63.79% F-1. The classification time for the two methods was obtained the same (0.26 seconds). However, Multinomial Naive Bayes spent less memory than the Gaussian method.

6.2. Support Vector Machine (SVM)

In order to get better results using SVM, four different kernels were tested: linear, polynomial, Radial Basis Function (RBF), and sigmoid. As depicted in Table 3, the linear kernel could achieve the best accuracy of 76.24%, the precision of 71.26%, and the best F-1 of 71.52%. The linear kernel also had the best performance in terms of classification time and memory.

6.3. Decision Tree (DT)

As the third classifier, DT was employed considering the split criterion. The results obtained from this classifier are provided in Table 4. As shown in this table, the best result was observed for the Gini criterion with an accuracy of 70.02%, a precision of 64.42%, and an F-1 of 64.16%. The classification time and memory were obtained almost the same for both criteria.

6.4. Random forest (RF)

RF was used with different numbers of trees. The results of this classifier are indicated in Table 5. As shown in Table 5, the best result was recorded for 250 trees with an accuracy of 77.79%, a precision of 75.31%, and an F-1 of 72.52%. However, the RF algorithm with 50 trees had the best classification time and memory. Moreover, the Gini impurity was used as the measure of the quality of a split. Fig. 5 compares the effect of the number of features to consider when looking for the best split.

6.5. Single Layer Perceptron (SLP)

For this classifier, different numbers of hidden nodes were considered in order to obtain the optimal model. Table 6 presents the obtained results for this classifier. The best accuracy of 78.92% was recorded by using five hidden nodes. Moreover, SLP with three hidden nodes recorded the best classification time and memory.

6.6. Bidirectional Encoder Representations from Transformers (BERT)

Eventually, for the BERT classifier and after running the analysis many times, the following parameters were used: batch size = 10, number of epochs = 15, and learning rate of 0.001. Table 7 summarizes the obtained results for the BERT classifier by considering the mentioned parameters and using the fivefold cross-validation with four repeats. Finally, the performance metrics were averaged for all four repeats. As provided in Table 7, the average accuracy, precision, and F-1 were obtained as 81.59%, 78.24%, and 77.96%, respectively. The total classification time and memory were obtained as 1.7 seconds and 432 Mb, respectively.

In order to get a better insight into the BERT model results, detailed information on the keywords identified by the model for classifying two sample narratives is provided in Fig. 5. To generate the visualizations in this figure, we used the Captum library, which applies a gradient-based attribution method to interpret model predictions (Ancona et al., 2017). As can be seen in Fig. 5(a), "traveling," "westbound," "eastbound," and "lanes" are the words having a positive impact on the model, which led to classifying the narrative as an actual WWD crash (Label 1). Moreover, as shown in Fig. 5(b), "crossed over" and "center line" are the keywords having a negative impact on the model. The negative impact implies that the crash narrative is a non-WWD crash (Label 0). Based on the definition, vehicles crossing the center line or the median are not identified as actual WWD crashes. These results indicate that the proposed BERT model successfully learned the words representing the actual WWD crashes vs non-WWD crashes in the police crash narrative.

Table 2
Obtained performance metrics for two different variants of NB classifier.

Method	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
Gaussian Naive Bayes	59.03%	51.63%	48.48%	0.26	89
Multinomial Naive Bayes	**73.51%**	**74.58%**	**63.79%**	**0.26**	**86**

Table 3
Performance metrics of SVM for four different kernels.

Kernel	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
Linear	**76.24%**	71.26%	**71.52%**	**0.38**	**81.5**
Polynomial	61.57%	65.50%	29.26%	0.41	82
RBF	66.98%	**76.67%**	44.87%	0.41	81.5
Sigmoid	58.79%	51.32%	47.66%	0.39	82

Table 4
Performance metrics of DT for two different split criteria.

Criterion	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
Gini	**70.02%**	**64.42%**	**64.16%**	0.29	85
Entropy	69.13%	63.00%	63.34%	**0.29**	**84**

Table 5
Performance metrics of RF for different numbers of trees.

Num. of Trees	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
50	75.94%	74.11%	69.51%	**0.42**	**88.7**
100	76.72%	74.57%	70.77%	0.54	89
150	76.71%	73.70%	70.91%	0.66	89.5
200	77.01%	74.39%	71.41%	0.80	89.9
250	**77.79%**	**75.31%**	**72.52%**	0.92	90.2
300	77.49%	74.76%	72.09%	1.05	90.8

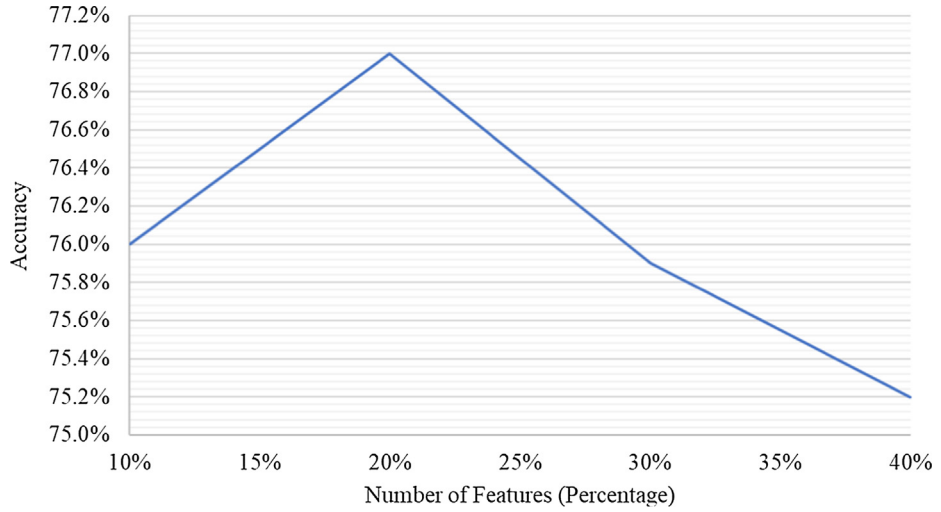

Fig. 5. Accuracy of RF classifier for different numbers of features.

Table 6
Performance metrics of SLP for different numbers of hidden nodes.

Num. of Nodes	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
3	78.56%	**75.82%**	73.71%	**0.46**	**78.8**
4	78.15%	74.77%	73.49%	0.54	78.9
5	**78.92%**	75.76%	**74.62%**	0.63	79.5
6	78.80%	75.58%	74.27%	0.71	79.6
7	78.15%	74.48%	73.63%	0.78	79.8

Table 7
Performance metrics of BERT for different repeats.

Repeat Num.	Accuracy	Precision	F-1	Classification Time (s)	Memory Usage (Mb)
1	79.80%	78.53%	75.32%	1.7	432
2	81.94%	77.48%	78.70%		
3	81.94%	77.83%	78.78%		
4	82.66%	79.12%	79.06%		
Average	**81.59%**	**78.24%**	**77.96%**	_	_

6.7. Algorithms comparison

In order to have a better insight into choosing the best model for classifying crash report narratives, a comparison was made between the best performance obtained for each algorithm. Fig. 6 illustrates the best accuracy for all selected classifiers. According to this figure, it was concluded that BERT had the best performance with 81.59% accuracy among other classifiers. SLP ranked second place after BERT having the best accuracy value of 78.92%. Consequently, according to obtained results, it can be concluded that the BERT model outperformed compared to the other five conventional classification algorithms (NB, SVM, DT, RF, and SLP). Hence, the BERT model can be considered as one of the reliable classifiers for identifying actual WWD crashes from potential WWD crashes in police report narratives.

In Addition, Fig. 7 shows the classification time and memory for all the best classifiers mentioned earlier. According to Fig. 8, NB had the shortest classification time compared to other classifiers. On the other hand, SVM had the lowest memory

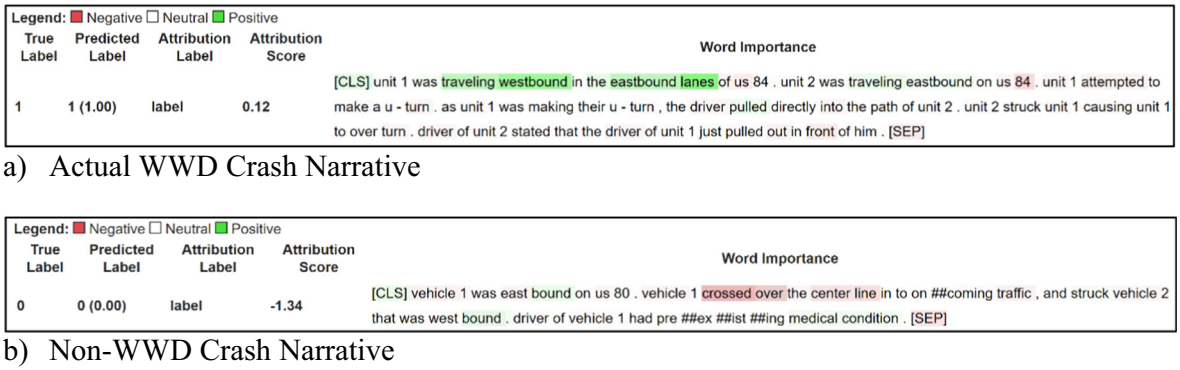

Fig. 6. Word importance visualization for two sample crash narratives.

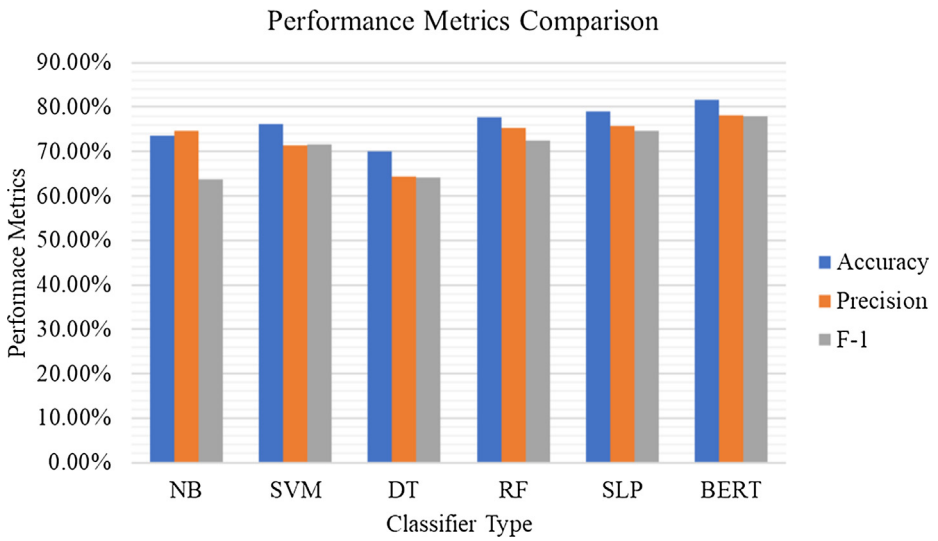

Fig. 7. Accuracy comparison for the selected classifiers.

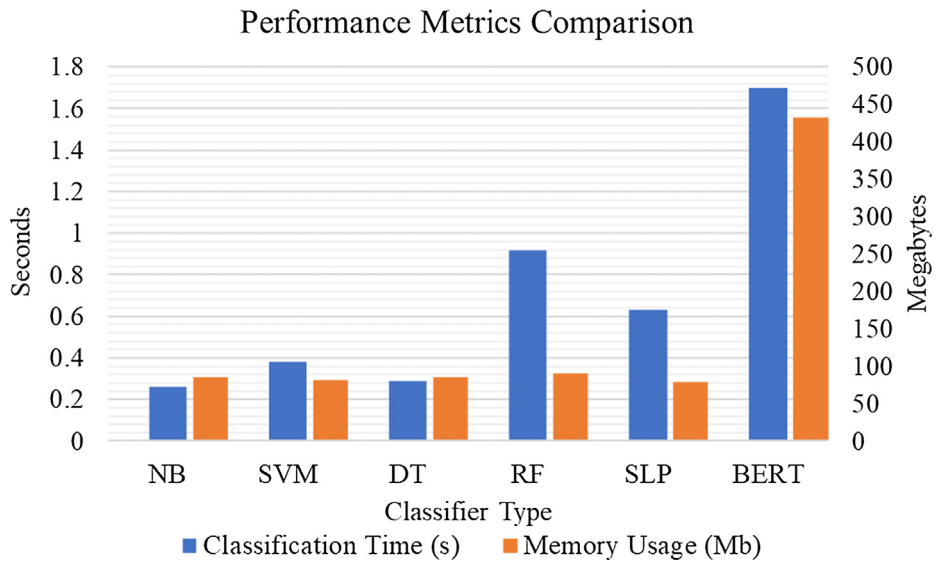

Fig. 8. Classification time and memory usage comparison for selected classifiers for one sample.

usage. It is notable to mention that, on average, it takes around 5 minutes to manually review each crash narrative and classify it. Hence, the proposed methodology can significantly reduce the time needed to identify actual WWD crashes from crash report narratives.

7. Conclusions

With the increase in the number of available textual data as well as the rapid advances in technology and programming software, the application of machine learning techniques for text classification has drawn remarkable popularity in recent decades. The main contribution of this research work was to identify actual WWD crashes from potential WWD crashes in crash report narratives using text mining and machine learning techniques. Generally, there is not any resource or database directly providing WWD crash data. One of the potential solutions is to filter relevant variables of the general crash dataset and verify them by reviewing hard copies of those crashes, which is a very time-consuming task. In order to overcome this issue, we attempted to classify crash report narratives into actual WWD and non-WWD crash categories by incorporating an emerging pre-trained language model as well as five conventional classification algorithms (NB, SVM, DT, RF, and SLP) with 5-fold cross-validation. Accuracy, F-1Score, precision, classification time, and memory were selected as the evaluation metrics. Results showed that the BERT classifier outperformed other conventional classifiers in categorizing WWD crash report narratives, although this classifier requires more classification time and memory usage compared to other methods. The best accuracy for BERT was recorded as 81.59%. SLP was the second classifier, recording an accuracy of 78.92%. This research work provided methodological evidence that the BERT algorithm can be promising in identifying actual WWD crashes from potential WWD crashes in the police report narratives. Moreover, by suggesting a viable text mining technique for automating the classification process, this study provides the opportunity for practitioners and state agencies to eliminate the man-hours required for reviewing hardcopy crash report narratives manually.

The data used in this study was only for Illinois and Alabama. However, the methodology developed can be employed for any textual data with classification purposes, specifically crash report narratives. Moreover, due to the limitations of accessing the crash reports, only 421 cases were obtained. Hence, as a future endeavor, the authors will expand the analysis of crash report narratives classification by covering more states and accessing more data. In addition, the suggested BERT model is limited to a certain sentence length, usually 512 words. All the crash narratives used in this study were less than 512 words. However, as future work, other BERT models capable of dealing with this issue will be applied to the crash narratives with longer sentence lengths.

Conflict of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abikoye, O.C., Omokanye, S.O., Aro, T.O., 2018. Text Classification Using Data Mining Techniques: A Review.

Adhikari, A., Ram, A., Tang, R., Lin, J., 2019. DocBERT: BERT for Document Classification.

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E.D., Gutierrez, J.B., Kochut, K., 2017. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques.

Ancona, M., Ceolini, E., Öztireli, C. and Gross, M., 2017. Towards better understanding of gradient-based attribution methods for deep neural networks. *arXiv preprint arXiv:1711.06104*.

Arteaga, C., Paz, A., Park, J.W., 2020. Injury severity on traffic crashes: A text mining with an interpretable machine-learning approach. Saf. Sci. 132,. https://doi.org/10.1016/j.ssci.2020.104988 104988.

Breiman, L., 2001. Random Forests. Mach. Learn. 45, 5–32.

Campos, D., Silva, R.R., Bernardino, J., 2019. Text mining in hotel reviews: Impact of words restriction in text classification. In: IC3K 2019 – Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management. SciTePress, pp. 442–449. https://doi.org/10.5220/0008346904420449.

Colas, F., Brazdil, P., 2006. On the behavior of SVM and some older algorithms in binary text classification tasks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4188 LNCS, 45–52. https://doi.org/10.1007/11846406_6.

Dadgar, S.M.H., Araghi, M., Farahani, M., 2016. A Novel Text Mining Approach Based on TF-IDF and Support Vector Machine for News Classification. 2nd IEEE International Conference on Engineering and Technology (ICETECH), pp. 16–20.

Das, S., Sun, X., Dutta, A., 2016. Text mining and topic modeling of compendiums of papers from transportation research board annual meetings. Transp. Res. Rec. 2552, 48–56. https://doi.org/10.3141/2552-07.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies – Proceedings of the Conference 1, pp. 4171–4186.

Duan, J., Zeng, J., 2013. Web objectionable text content detection using topic modeling technique. Expert Syst. Appl. 40, 6094–6104. https://doi.org/10.1016/j.eswa.2013.05.032.

Fatima, S., Shugufta, B.S., Tech, F.M., Srinivasu, B., 2017. Text Document categorization using support vector machine. Int. Res. J. Eng. Technol.

Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D., Fernández-Delgado, A., 2014. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? J. Mach. Learn. Res.

FHWA, 2021. Wrong Way Driving [WWW Document]. Federal Highway Administration. URL https://safety.fhwa.dot.gov/intersection/other_topics/wwd/ (accessed 7.15.21).

Goldberg, Y., Levy, O., 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method, pp. 1–5.

González-Carvajal, S., Garrido-Merchán, E.C., 2020. Comparing BERT against traditional machine learning text classification.

Hassan, S., Rafi, M., Shahid Shaikh, M., 2011. Comparing SVM and Naïve Bayes Classifiers for Text Categorization with Wikitology as knowledge enrichment.

Ho, T.K., 1995. Random Decision Forests Tin Kam Ho Perceptron training. Proceedings of 3rd International Conference on Document Analysis and Recognition 1, pp. 278–282.

Hong, S., 2012. Online news on Twitter: Newspapers' social media adoption and their online readership. Inf. Econ. Policy 24, 69–74. https://doi.org/10.1016/j.infoecopol.2012.01.004.

Joachims, T., 1998. Text categorization with support vector machines: Learning with many relevant features. Eur. Conf. Mach. Learn., 137–142

Kowsari, K., Brown, D.E., Heidarysafa, M., Jafari Meimandi, K., Gerber, M.S., Barnes, L.E., 2017. HDLTex: Hierarchical Deep Learning for Text Classification. In: Proceedings – 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017 2017-Decem, pp. 364–371. https://doi.org/10.1109/ICMLA.2017.0-134.

Kowsari, K., Meimandi, K.J., Heidarysafa, M., Mendu, S., Barnes, L., Brown, D., 2019. Text classification algorithms: A survey. Information (Switzerland) 10, 1–68. https://doi.org/10.3390/info10040150.

Kumbhar, P., Mali, M., 2016. A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification. Int. J. Sci. Res.

Lee, C.H., 2012. Unsupervised and supervised learning to evaluate event relatedness based on content mining from social-media streams. Expert Syst. Appl. 39, 13338–13356. https://doi.org/10.1016/j.eswa.2012.05.068.

Lin, C., He, Y., 2009. Joint sentiment/topic model for sentiment analysis. International Conference on Information and Knowledge Management, Proceedings, pp. 375–384. https://doi.org/10.1145/1645953.1646003.

Martinez-Romo, J., Araujo, L., 2013. Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst. Appl. 40, 2992–3000. https://doi.org/10.1016/j.eswa.2012.12.015.

National Highway Traffic Safety Administration (NHTSA), 2019. Fatality Analysis Reporting System (FARS). URL https://www-fars.nhtsa.dot.gov/Main/index.aspx (accessed 7.21.21).

National Highway Traffic Safety Administration (NHTSA), 2020. Alabama Uniform Traffic Crash Report. URL https://one.nhtsa.gov/nhtsa/stateCatalog/states/al/alabama.html (accessed 1.15.21).

Onan, A., Korukoğlu, S., Bulut, H., 2016. Ensemble of keyword extraction methods and classifiers in text classification. Expert Syst. Appl. 57, 232–247. https://doi.org/10.1016/j.eswa.2016.03.045.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Pennington, J., Socher, R., Manning, C.D., 2014. GloVe: Global Vectors for Word Representation Jeffrey, in: In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1532–1543.

Rana, M.I., Khalid, S., Akbar, M.U., 2014. News classification based on their headlines: A review. In: 17th IEEE International Multi Topic Conference: Collaborative and Sustainable Development of Technologies, IEEE INMIC 2014 – Proceedings, pp. 211–216. https://doi.org/10.1109/INMIC.2014.7097339.

Salton, G., Buckley, C., 1998. Term-weighting approaches in automatic text retrieval. Inf. Process. Manag. 24, 513–523.

Shahare, P.D., Giri, R.N., 2015. Comparative Analysis of Artificial Neural Network and Support Vector Machine Classification for Breast Cancer Detection. Int. Res. J. Eng. Technol.

So, J. (Jason), Park, I., Wee, J., Park, S., Yun, I., 2019. Generating Traffic Safety Test Scenarios for Automated Vehicles using a Big Data Technique. KSCE J. Civil Eng. https://doi.org/10.1007/s12205-019-1287-4.

Sun, C., Qiu, X., Xu, Y., Huang, X., 2019. How to Fine-Tune BERT for Text Classification? Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11856 LNAI, pp. 194–206. https://doi.org/10.1007/978-3-030-32381-3_16.

Trueblood, A.B., Pant, A., Kim, J., Kum, H.C., Perez, M., Das, S., Shipp, E.M., 2019. A semi-automated tool for identifying agricultural roadway crashes in crash narratives. Traffic Inj. Prev. 20, 413–418. https://doi.org/10.1080/15389588.2019.1599873.

Vapnik, V.N., Chervonenkis, A.Y., 1964. A class of algorithms for pattern recognition learning. Avtomat. i Telemekh 25, 937–945.

Wali, B., Khattak, A.J., Ahmad, N., 2021. Injury severity analysis of pedestrian and bicyclist trespassing crashes at non-crossings: A hybrid predictive text analytics and heterogeneity-based statistical modeling approach. Accid. Anal. Prev. 150,. https://doi.org/10.1016/j.aap.2020.105835 105835.

Wang, T.Y., Chiang, H.M., 2011. Solving multi-label text categorization problem using support vector machine approach with membership function. Neurocomputing 74, 3682–3689. https://doi.org/10.1016/j.neucom.2011.07.001.

Waters, R.D., Jamal, J.Y., 2011. Tweet, tweet, tweet: A content analysis of nonprofit organizations' Twitter updates. Public Relat. Rev. 37, 321–324. https://doi.org/10.1016/j.pubrev.2011.03.002.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., Platen, P. von, Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. le, Rush, A., 2020. Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) Proceedings of the 2020 EMNLP (Systems Demonstrations). pp. 28–45.

Yu, B., Xu, Z.B., 2008. A comparative study for content-based dynamic spam classification using four machine learning algorithms. Knowl.-Based Systems 21, 355–362. https://doi.org/10.1016/j.knosys.2008.01.001.

Zhang, X., Green, E., Chen, M., Souleyrette, R.R., 2020. Identifying secondary crashes using text mining techniques. J. Transp. Saf. Security 12, 1338–1358. https://doi.org/10.1080/19439962.2019.1597795.

Zheng, S., Yang, M., 2019. A New Method of Improving BERT for Text Classification, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer International Publishing. https://doi.org/10.1007/978-3-030-36204-1_37.