TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–10

DOI: 10.1177/03611981231157393
journals.sagepub.com/home/trr
SAGE

Applying Few-Shot Learning in Classifying Pedestrian Crash Typing

Yanmo Weng[1], Subasish Das[2], and Stephanie German Paal[1]

Abstract

Pedestrian deaths account for 23% of all road traffic fatalities worldwide. After declining for three decades, pedestrian fatalities in the United States have been increasing with 6,941 fatalities in 2020, the highest number for more than two decades, impeding progress toward a zero-deaths transportation system. The Pedestrian and Bicycle Crash Analysis Tool (PBCAT) was developed to describe the pre-crash actions of the parties involved to better define the sequence of events and precipitating actions that lead to crashes involving motor vehicles and pedestrians or cyclists. Undoubtedly, police crash data influence decision-making processes in the transportation agencies. Using crash data from three major cities in Texas (during the period from 2018 to 2020), this study assessed the data quality of police-reported crash narratives on pedestrian-involved traffic crashes. As the pedestrian crash typing involves many categories, conventional machine-learning algorithms will not be sufficient in solving the classification problem from narrative texts. This study used few-shot learning (FSL), an advanced machine learning, to solve this issue. Using the pre-knowledge obtained from five different crash types and a few labeled data points of three unseen new crash types, the proposed model achieved roughly 40% overall accuracy. Also, four different configurations of crash types were formed and tested which indicates that the model is robust.

Keywords

safety, pedestrian and bicyclist safety, sustainability and resilience, transportation and society, community resources and impacts

In recent years, the number of fatalities and serious injuries among vulnerable road users, particularly pedestrians, has risen dramatically. Pedestrian mobility is obviously critical in the transportation system. In 2020, 6,941 pedestrians died on U.S. roadways as a result of traffic accidents, the highest number of pedestrian fatalities in the last two decades (*1*). The rising number of pedestrian fatalities necessitates improved pedestrian safety analysis methods.

Machine learning has been extremely successful in data-intensive applications, but it is frequently hampered by a small data set. Few-shot learning (FSL) has recently been proposed as a solution to this problem. FSL can quickly generalize to new tasks containing only a few samples with supervised information using prior knowledge (*2*). This study aims to further the understanding of the use of police-reported crash report content in identifying PBCAT Version 3.0 crash typing. This study collected crash narratives from Texas and randomly selected 500 narrative reports to manually classify the pedestrian crash typing. As the manual gold-standard data are limited, FSL seems to be the most feasible machine-learning algorithm for improving model performance. This study offers a methodological framework to quantify the classification accuracies using FSL.

The rest of this paper is organized as follows. We summarize related work in the literature review before presenting the data preparation work and descriptive statistics. We then briefly explain the PBCAT Version 3.0 crash-typing approach and concepts of FSL. After reporting the results and findings, we conclude with remarks on future research.

[1]Zachry Department of Civil and Environmental Engineering, Texas A&M University, College Station, TX
[2]Ingram School of Engineering, Texas State University, San Marcos, TX

Corresponding Author:
Subasish Das, subasish@txstate.edu

Literature Review

There is a substantial body of literature on different safety and mobility issues associated with pedestrians. The current literature is limited to studies on non-motorist crash typing.

The Federal Highway Administration (FHWA) and the National Highway Traffic Safety Administration (NHTSA) launched the Pedestrian and Bicycle Crash Analysis Tool (PBCAT) in 2000 to aid in the analysis of the details of these crashes so that problems and solutions can be easily determined (*3*). PBCAT was created to describe the pre-crash actions of the parties involved to better define the sequence of events and precipitating actions that led to collisions between motor vehicles and pedestrians or cyclists. The most recent version of PBCAT, PBCAT Version 3.0 (*4*), has been updated and modified to assist road safety professionals in improving crash data about pedestrian and cyclist crashes to better understand and address associated safety risks. The NHTSA recently published an associated coding manual (*5*).

There have been few studies involving PBCAT-related analysis. A recent study coded pedestrian and cyclist crashes in North Central Texas using the PBCAT (version 2.0) methodology (*6*). This study created a pilot version of an online application to visually present the results and communicate the North Central Texas safety needs. Using the Tennessee Integrated Traffic Analysis Network (TITAN) database (version 2.0), Shah et al. (*7*) examined 52 e-scooter and 79 bicycle police-reported crashes in Nashville, Tennessee, from April 2018 to April 2020. For e-scooter and bicycle crashes, the authors found statistically significant differences in spatial and temporal distribution, demographics, lighting conditions, and crash distance from home. This investigation of police crash reports added to the existing literature by providing a comprehensive picture of e-scooter safety. Das et al. (*8*) created a framework for classifying crash types from unstructured textual content using machine-learning models. The research team collected pedestrian crash-typing data from two Texas locations for this study. The best classifier was discovered to be the XGBoost model. Lopez et al. (*9*) assessed the data quality of text narratives in police reports about bicycle accidents. Police reports were compared with the PBCAT (version 2.0) to determine how much information was captured and when reports were more likely to capture more information. The findings revealed that police reports only captured most information in one section of the standardized form (crash typing), with an average total of more than 75% missing. The findings also revealed that, while longer reports result in less missing information when compared with the standardized crash form, the average report still misses most of the information that the form would capture.

Schneider and Stefanich (*10*) introduced the location-movement classification method (LMCM) for classifying pedestrian and bicycle crashes, demonstrating that the LMCM provides useful information that PBCAT does not. Both typologies were applied to 296 pedestrian and 229 bicycle crashes in Wisconsin reported between 2011 and 2013. Chavis et al. (*11*) examined three years of pedestrian and bicycle crashes in Washington, D.C. (2012–2014). The crash typologies PBCAT (version 2.0) and LMCM were used in the classification. The NHTSA crash groups were identified, and the top three groups were thoroughly examined using PEDSAFE and BIKESAFE countermeasures. The shortcomings of the police crash report forms used were also discussed. Kwayu et al. (*12*) evaluated the pedestrian crossing-related crashes based on the report narratives. The Chi-squared test, ordered probit model, and random forest (RF) model were applied to analyze the crash. The crash narratives data were provided by the Michigan State Police, Office of Highway Safety and Planning (OHSP). The predictors of the RF model were identified by the crash narratives. Detailed crash analysis was provided, and the results showed that the crash narratives provide useful knowledge to classify the injury severity. Wali et al. (*13*) proposed a two-stage hybrid approach for analyzing the injury severity of pedestrian and cyclist crashes. Advanced text analysis techniques were used to extract useful data from trespasser crash narratives. This study used the 10-year (2006–2015) crash narratives obtained from Federal Railroad Administration's website as the key input for the text mining procedures. In total, 6,470 pedestrians or cyclists involved in non-crossing trespassing crashes were studied.

The studies mentioned have several limitations. First, the result findings should be interpreted based on several assumptions given the limitations of the data. Also, the range of the response variables (e.g., crash types, injury severity) in the training and testing set should be the same. In other words, the studies mentioned might not be able to analyze an unseen crash type or severity level. Usually, a good prediction for an unseen crash type needs a large amount of data for that unseen crash type. However, expanding the data set with a large amount of data in a new class might be impossible. The uniqueness of this study is that a good prediction performance for an unseen new crash type can be achieved without adding a large amount of data. The only thing needed in the prediction is a few labeled examples of that new class.

Data Preparation

This study assessed the collected data's suitability for the classification task. This study collected 2018–2020 pedestrian crash data from the Texas Department of

Table 1. Descriptive Statistics for Full Data Set

Variable	Category	Percent	Variable	Category	Percent
Year	2018	34.6	Collision type	One vehicle going through	64.88
	2019	37.46		One vehicle turning left	20.08
	2020	27.94		One vehicle turning right	8.44
Crash severity (KABCO scale)	K	10.04		Other	6.6
	A	18.73	City	Austin	23.73
	B	38.9		Dallas	36.99
	C	27.94		San Antonio	39.28
	PDO	4.21	Light condition	Day	48.56
	Unknown	0.18		Dark, lighted	35.64
Contributing factor	None specified	48.29		Dark, not lighted	12.2
	Driver failed to yield to pedestrian	15.51		Dark, unknown lighting	1.31
	Driver inattention	12.79		Dawn	0.92
	Speed-related	3.65		Dusk	1.17
	Pedestrian failed to yield to driver	1.62		Other	0.14
	Faulty evasive action	1.33		Unknown	0.07
	Other	16.82	Relation to intersection	At intersection	7.77
Number of pedestrians	1	95.27		Intersection-related	37.96
	2	4.07		Driveway access	4.89
	$\geqslant$ 3	0.65		Non-intersection	49.39

Note: The KABCO Scale: K = fatal; A= incapacitating injury; B = non-incapacitating injury; C = possible injury; O/PDO = property damage only.

Transportation's (DOT) crash record information system (CRIS) for three major Texas cities: Austin, Dallas, and San Antonio. A total of 4,442 crash data were collected with crash narrative information. Table 1 provides descriptive statistics for the full data set. Table 1 shows that 2019 had a higher percentage of pedestrian crashes compared with other years. As expected for pedestrian–vehicle crashes, very few of the crashes are no injury. In almost two thirds of the crashes, there was one straight-proceeding vehicle involved. Roughly half of the crashes were at or near intersections or driveways. The distribution of crashes across the light condition categories roughly matches the distribution of light conditions in a typical day.

Contributing factors were not coded in the corresponding unit (vehicle or pedestrian) records in roughly half of the crashes. Driver failure to yield to pedestrian (15.51% of crashes), driver inattention (12.79%), or speed-related factors were the most common contributing factors in crashes with coded contributing factors (3.65%). "Speed-related" refers to a combination of three database code values: failed to control speed; speeding-unsafe (under limit); and speeding-unsafe (over limit). More than 95% of the crashes involved only one pedestrian, 4% involved two pedestrians, and less than 1% involved three or more pedestrians. With regard to location, 7.77% of the crashes occurred at intersections, 37.96% occurred on the roadway associated with the intersection, 4.89% occurred in the driveway, and 49.39% occurred at a non-intersection.

To perform the analysis, a list of 500 crash narrative reports was chosen. The researchers used PBCAT 3.0 protocols to manually classify pedestrian crash typing by meticulously reviewing the crash narratives and associated collision diagrams. Pedestrian crash typing is based on two major components, according to PBCAT 3.0: (1) a driver maneuver; and (2) a pedestrian maneuver. A pedestrian crash typed "S-CR," for example, indicates that the driver is going straight (S) and the pedestrian is crossing the path from the motorist's right (CR). Because "collision type" in structured crash data can determine driver or motorist maneuvers, this study focuses on classifying "pedestrian maneuver types" from the crash narrative report using FSL. The final supervised data set is divided into 11 pedestrian maneuver classes, with the number of datapoints in each class listed in Table 2 and sorted from highest to lowest count. Because the sample size was too small, crash types with fewer than 15 counts (i.e., "FC," "MU," and "OU") were excluded from the data set. As a result, the final data set contains eight classes (i.e., "CL," "CR," "CU," "PO," "PS," "PU," "ST," and "UN") and 481 data samples. Aside from the 481 labeled data samples, the data set also includes an unlabeled data set with a total of 4,155 data samples. Currently, only the labeled data set is used in the model training and testing process; however, the unlabeled data

Table 2. Pedestrian Crash Type

Pedestrian maneuver type	Count
CR (crossing path from motorist's right)	134
CL (crossing path from motorist's left)	102
PS (parallel path same direction)	63
ST (stationary)	57
PO (parallel path opposite direction)	49
CU (crossing path, unknown direction)	37
PU (parallel path unknown direction)	22
UN (unknown)	17
OU (other/unusual)	14
MU (moving in unknown path/direction)	8
FC (non-motorist fall or crash)	5

set may be used in the future to validate the trained model.

Methodology

Few-Shot Learning

This study is considered an application of deep learning techniques in the pedestrian crash type classification based on the reported narratives. Commonly, collected data sets have several crash types which only have a very limited number of data samples. However, expanding the data set is time-consuming and expensive. Also, the conventional machine-learning algorithms (e.g., random forest, support vector machine, artificial neural network, etc.) assume that the feature space of the training and testing set should be the same. In other words, these algorithms are not be able to predict out-of-the-distribution data samples. In addition, these algorithms require a large amount of data for training, and sometimes the data set size is not large enough to obtain a good prediction. The FSL approach can be an ideal machine-learning approach to solving the classification problem with only a very limited number of data available. First, the formal definition of FSL can be described as a type of machine-learning problem where the training data only contain a limited number of examples for an available task (*2*). Several ML methods can achieve the FSL objective (*14–16*). Meta-learning is regarded as a general framework for FSL among various methods (*17*). The key purpose of meta-learning is to obtain good initial parameters. In this research, the model-agnostic meta-learning (MAML) algorithm (*18*) is used to achieve FSL. MAML is one of the most well-known and widely used types of meta-learning model. The MAML algorithms first obtain the pre-knowledge from a relatively large database and then learn the unseen new classes of crash type using only a few gradients' decent update progresses (*18*). Specifically, the MAML aims to train a learner to learn a new unseen class with only a small number of data provided. MAML trains the learner to improve the overall prediction ability across different problems rather than training the model to solve a specific problem. The conventional supervised learning algorithm splits the data set into a training set and a testing set. The feature space of the training set and testing set is the same. The model parameter is updated based on minimizing the loss of the training set. However, in the MAML algorithm, the minimum unit for training is the task. Each task contains training samples and testing samples and is further named as a support set and a query set. The number of classes (***N***) in the support set defines the selected task as an ***N***-class classification problem, also called an ***N***-way task. The number of labeled examples in each class (***K***) is called ***K***-shot. Using the above definitions, in MAML a sampled task can be described as an ***N***-way, ***K***-shot task. For instance, assume we have a data set containing five classes (i.e., $C_1,\ C_2,\ C_3,\ C_4,\ and\ C_5$) and each class has multiple datapoints. If a two-way, two-shot problem is needed to be solved, two out of the mentioned five classes are randomly selected and each class should contain two labeled sampled examples.

In the MAML algorithm, a random initial model parameter ϕ is first generated and the tasks are randomly sampled following the ***N***-way, ***K***-shot formula. First, the initial model parameter is fed into a learner (e.g., a multilayer perceptron [MLP] model), then the learner is trained based on the data from the support set. The learner then makes a prediction using the query set as the input for the trained learner. The loss comes from the query set examples used for the backpropagation process. The model parameter is finally updated based on optimizing the loss. The updated loss serves as the initial model parameter for a new randomly sampled task. When all the tasks are sampled and model parameters are updated, the iteration will end and move to the next iteration. The detailed training process for each task is shown in Figure 1. Unlike the conventional machine-learning training process, in each training iteration of the MAML algorithm we need to compute two different gradient updates. The first update occurs while training the learner based on the support set. The second update occurs after computing the loss of the query set. This helps the learner to enhance the model to learn the generalized information and avoid the overfitting issue.

The mentioned MAML algorithm aims to train a learner to learn the pre-knowledge from the training set and rapidly learn the new classes within a few gradient decent steps. The MAML algorithm is a model-agnostic algorithm, and usually a deep learning neural network will be selected as the learner. For example, an MLP is used as the learner in Figure 1. For a natural language processing problem, the convolutional neural network (CNN)

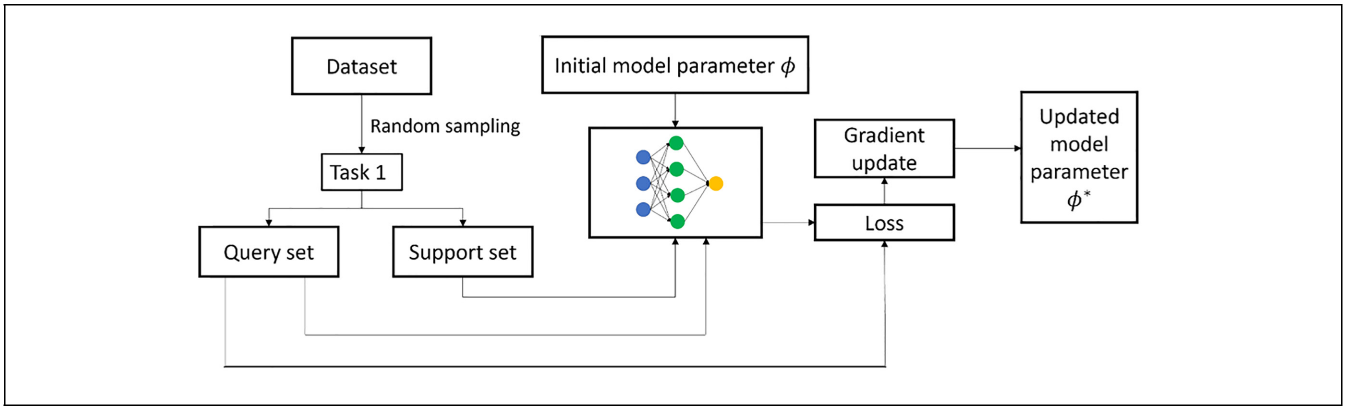

Figure 1. Illustration of Model-Agnostic Meta-Learning (MAML) algorithm.

can be used as the learner. Also, a lot of advanced and reliable models have been used in the NLP field. In this research, the bidirectional encoder representations from transformers (BERT) model is used as the basic learner for the MAML algorithm (*18*). BERT is a transformer architecture (*19*) designed for learning language representations. It can be widely used in NLP tasks (e.g., text summarization, text classification, etc.). BERT is evolved from the transformer architecture and pre-trained models for NLP tasks using encoders as a sub-structure (*20*). Transformer is a sequence transduction model based on self-attention mechanisms (*21*). The original Transformer model used an encoder-decoder architecture, where the encoder was used to read the sequence input and the decoder generated the task prediction. BERT was developed for language understanding, thus only the encoder with self-attention is necessary (*21*). The attention mechanism enables the Transformer encoder to process the entire input sequence at once. Therefore, the BERT model can take into account both sequential directions of the input text to capture more meaningful contextual information. Figure 2 illustrates the overall architecture of the BERT classification network. The input text is split into a sequence of tokens (Tok) and then processed in the BERT model (*22*). [CLS] is a special classification token inserted at the beginning of every input sequence. The BERT output vectors of the special token [CLS] and the input token (Tok) are denoted as C and T, respectively. The output feature vectors corresponding to each token are used as the sequence representation for the classification task (*21*). BERT model was pre-trained on texts from the BooksCorpus data sets with 800 million words and the English Wikipedia with 2,500 million words (*21*). In general, the initial model parameter for the learner in MAML at the first iteration is randomly selected. Since the BERT is a pre-trained model and the pre-trained weight can be directly used here as the initial model parameter, it helps the model to quickly learn the tasks in the training set.

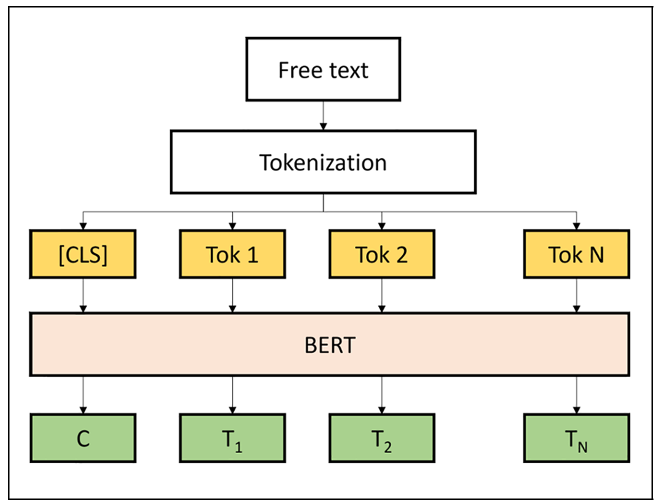

Figure 2. The bidirectional encoder representations from transformers (BERT) classification network architecture.

Data Preprocessing

In the used data set, the narrative of each sample is usually a several-sentence long summary. The text length of the narrative is smaller than the usual text length used in the input for the BERT model. To keep as much useful information as possible, the stop-words were first removed before encoding the text. The commonly used stop-words were first downloaded from the NLTK packages (*23*). Besides the commonly used stop-words, several additional stop-words were added after roughly reviewing all the narratives of the samples in the data set. The added stop-words include: “unit,” “northbound,” “southbound,” “eastbound,” “westbound,” “bound,”

Table 3. Training and Testing Trial Configurations

Trial number	Crash types in the training set	Crash types in the testing set
Trial #1	CL, CR, PU, ST, UN	PS, PO, CU
Trial #2	CU, CR, PS, PU, PO	ST, CL, UN
Trial #3	CL, CR, PS, PU, PO	ST, UN, CU
Trial #4	CL, CR, PS, ST, UN	PO, PU, CU

Note: CL = crossing path from motorist's left; CR = crossing path from motorist's right; PU = parallel path unknown direction; ST = stationary; UN = unknown; PS = parallel path same direction; PO = parallel path opposite direction; CU = crossing path, unknown direction.

"travel," "street," "st," "st.," "road," and "eat." After removing the stop-words, the BERT model was applied to encode the text using the BertTokenizer from the package called Transformers. The encoded string text further serves as the input for the MAML algorithm.

This study applied several data preprocessing methods before training the model. As mentioned previously, the tasks used for training the MAML algorithm for a classification problem should follow the ***N***-way, ***K***-shot formula. Also, there is no need to guarantee that the feature spaces of the training and testing set are the same. In other words, the testing set can contain unseen classes compared with the training set. In this research, we investigated a three-way, five-shot problem. Specifically, their classes were randomly selected from the prepared data set and the remaining five classes served as the training set. When randomly sampling the task, three out of six classes are randomly selected, and each class contains five labeled samples in the support set and three unlabeled samples in the query set. The N and K selection is limited by the data set. Specifically, there are eight crash types in the data set and the MAML algorithm needs firstly to train a learner using the majority part of the data set. Therefore, the training set is designed to contain five crash types and the remaining three crash types serve as the unseen new crash types in the testing set. Also, for each class, five labeled samples are already sufficiently used for fine-tuning the learner. Therefore, a three-way, five-shot task is used in this research. To further evaluate whether the MAML algorithm can be rapidly adopted into different new classes, the researchers randomly set up four trials. In each trial, the selected three crash types in the testing set were different. The detailed configurations of the trials are listed in the following Table 3.

Experimental Settings

The MAML algorithm updates the model weights first in the training stage and further updates the weight during the adoption of the new classes in the testing set. For both weight update stages, the Adam optimization algorithm (*24*) is used as the optimization algorithm for moving the loss toward the global minimum. This study used the cross-entropy loss function to train the model. This function is calculated as follows:

$$\ell(x,y) = L = \{l_1, \ \ldots, \ l_N\}^T, \ \ l_n = \ -w_{y_n} \ log\frac{exp(x_{n,y_n})}{\sum_{c=1}^{C} exp(x_{n,c})} \tag{1}$$

where
x is the input,
y is the target,
w is the weight,
C is the number of classes, and
N spans the minibatch dimension.

The detailed experimental setting for the MAML training and fine-tuning progress was introduced as follows. First, a data loader was implemented to sample the tasks into the three-way, five-shot formula. The size of the query set equals 3. The detailed BERT pre-trained model used in this research is the BERT base uncased model. As mentioned previously, the learning rates of the optimization algorithm for both gradient update steps were equal to 5e-5. During each iteration of training in the training set, five tasks were randomly sampled by the data loader. When the learner was well trained based on the samples from the training set, the learner was further trained based on a very small portion of the testing set. By doing so, the learner can classify the unseen classes accurately. The data loader was also applied to the testing set and only one task on the testing set was sampled following the three-way, five-shot formula. In other words, 15 examples were sampled from the testing set. There is no need to know any additional information about the testing set. The remaining data can be used as the testing set to test the model performance. The precision, recall, and F1 scores were applied to evaluate the prediction performance. Precision is an indicator for measuring the percentage of correct positive predictions among the total positive predictions. The recall was used to measure the percentage of correct positive predictions among the total actual positive case in the data set. The F1 score can be calculated based on the following equation (2). The F1 score is considered the harmonic mean of precision and recall. The higher the F1 score, the better prediction performance it can make.

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

Results and Findings

In each mentioned trial, the MAML model was trained based on the training set. When the model was well trained based on the training set, the updated model

Table 4. Prediction Results for Each Trial

Trial number	Percentage of the training set	Overall accuracy (%)	Target Class	Precision	Recall	F1 score
Trial no. 1	10.1	37.7	PS	0.391	0.600	0.474
			PO	0.400	0.267	0.320
			CU	0.333	0.267	0.296
Trial no. 2	8.5	44.4	ST	0.750	0.200	0.316
			CL	0.333	0.200	0.250
			UN	0.438	0.933	0.596
Trial no. 3	13.5	40	ST	0.294	0.333	0.313
			UN	0.500	0.467	0.482
			CU	0.429	0.400	0.414
Trial no. 4	13.8	37.7	PO	0.444	0.267	0.333
			PU	0.360	0.600	0.450
			CU	0.364	0.267	0.308

Note: CL = crossing path from motorist's left; PU = parallel path unknown direction; ST = stationary; UN = unknown; PS = parallel path same direction; PO = parallel path opposite direction; CU = crossing path, unknown direction.

parameters were saved as the final model parameter for the first training stage. The MAML model was then trained based on the single three-way, five-shot task that came from the testing set. Specifically, the saved final model parameter for the first training stage was first loaded as the initial model parameter for fine-tuning on the testing set. Then, a single three-way, five-shot task was sampled based on the designed data loader. After several gradient steps, the entire training progress was finished. To decrease the uncertainty caused by the random sampling, we repeated the progress of fine-tuning the testing set five times with different sampled three-way, five-shot tasks in each trial. The overall accuracy of the trials was computed as the average of five times the randomly sampled tasks results. Table 4 shows the overall average accuracy of each trial. Additional evaluation matrices (i.e., precision, recall, F1 score) are illustrated in Table 4 as well. This study focused on the multi-class classification problem. Therefore, in a specific trial, the evaluation metrics should be applied to one target class and should regard the remaining two classes as other classes. To further visualize the predicted label results, Figure 3 shows the confusion matrix for each trial. As mentioned previously, the fine-tuning progress for each trial is repeated five times with different sampled tasks. In each fine-tuning progress, a single three-way, five-shot task sampling is formed. Therefore, for each trial, there are 15 labeled data samples used for fine-tuning the model. The remaining unseen data samples are applied for testing the model performance. For instance, in trial No. 1, there are 149 data samples in three unseen crash types (i.e., PS, PO, CU). The model is first trained on the data sample from the five crash types (i.e., CL, CR, PU, ST, UN). Then, 15 out of 149 (i.e., 10.1%) data samples are used in the second training stage, the remaining 134 data samples are used for testing. The key focus of the proposed model is to use only a very limited number of data samples in the unseen crash types to predict many unseen data samples. Therefore, only using a small number of unseen data samples in the training stage is important. Table 4 shows the percentage of data samples in the training set for each trial.

Observed from Table. 3 and Figure 3, the proposed MAML model has the best prediction performance on trial no. 2. In trial no. 2, the MAML model predicts the "UN" class most accurately and the F1 score indicates that the prediction performance on "ST" and "CL" is not good. The potential reason for the high F1 score in "UN" is that the total number of samples in the "UN" class was very small. As a result, when sampling a three-way, five-shot task five times, some datapoints might be sampled multiple times, and the weights of the trained MAML model move toward the direction of minimum loss for the "UN" class. Similarly, the "PU" class only has a limited number of datapoints, and the true positive value shown in Figure 3*d* indicates that the prediction performance for "PU" is good. For the remaining trials (nos 1, 3, and 4), the prediction performances were similar, and the overall accuracy was close to 40%. In general, the performance is acceptable since only 15 examples of unseen classes were used for training. The potential factors preventing the performance from increasing may include the class unbalances issue and the small data set size of the training set. Specifically, the unbalanced classes issue might cause the model to not be considered a robust model. Additionally, the very small size of the training set means there is very limited pre-knowledge that can be used to train the learner. As a result, the prediction performance might not be good.

Although the overall accuracy of the proposed model is not high, the current outcome is still valuable since it is able to predict the unseen crash types with only a very

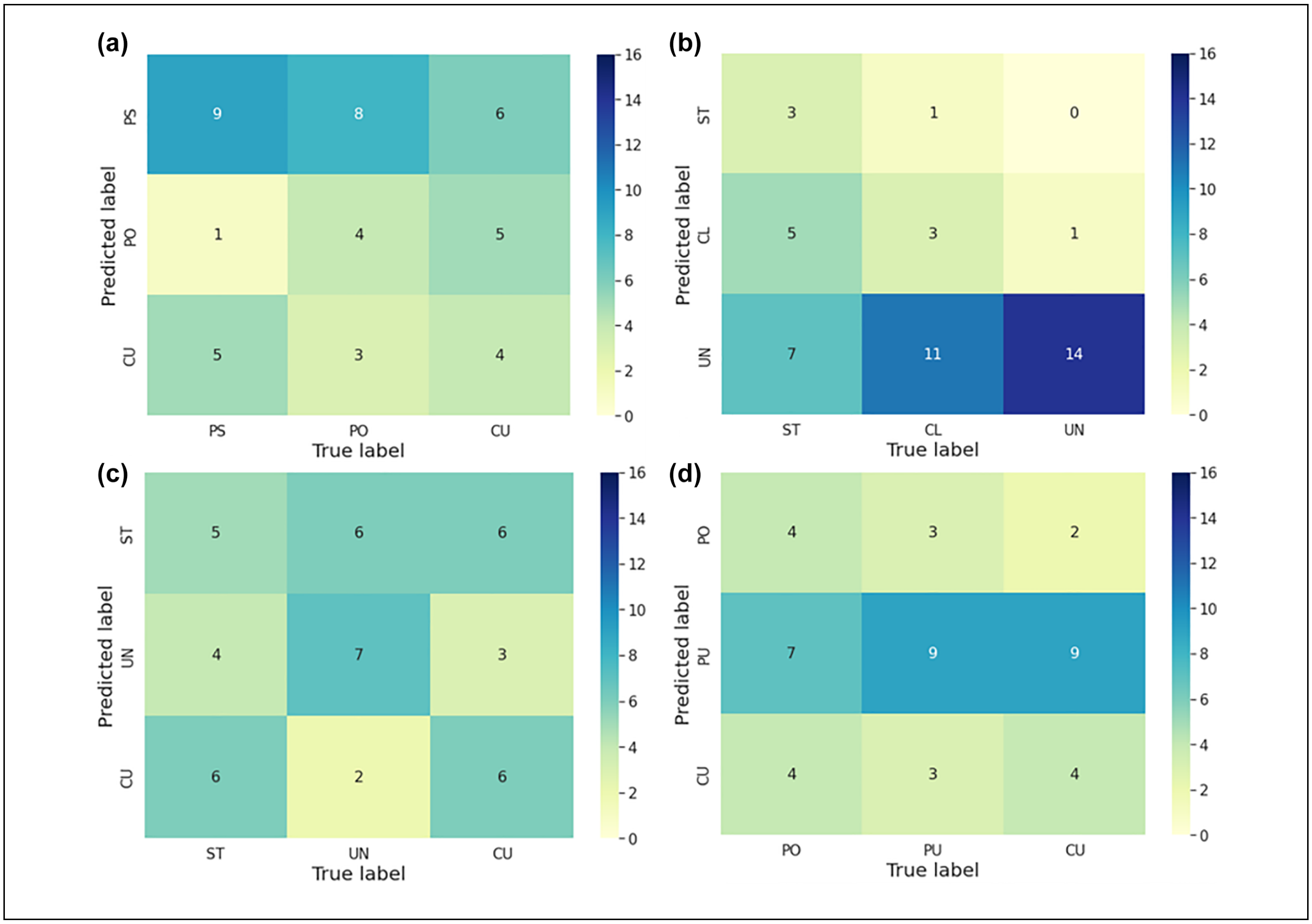

Figure 3. Confusion matrix for each trial: (*a*) trial no. 1; (*b*) trial no. 2; (*c*) trial no. 3; and (*d*) trial no. 4.
Note: CL = crossing path from motorist's left; PU = parallel path unknown direction; UN = unknown; PO = parallel path opposite direction.

limited number of data samples involved in the training process. Moreover, conventional machine learning cannot accomplish such a task. Specifically, the organization of training and testing set in FSL is different from conventional machine learning such as support vector machines and artificial neural networks. In conventional machine learning, the feature space of the training and testing set should be the same which means the training set should include these eight crash types and the testing set should also includes these eight crash types. However, the training set of the FSL only includes five crash types, and the testing set includes the remaining three crash types. Therefore, given the difference in training and testing set configuration, it is hard to use the conventional machine-learning algorithms as baseline models. The proposed FSL algorithm (i.e., MAML) is the first of its kind in this field. The result of this research might, therefore, be able to serve as the baseline model for some more advanced FSL algorithms. The superiority of FSL can already be revealed from its ability to predict the unseen crash types by only using a very limited number of data samples.

Conclusion

Recently, advanced language models have been adopted in several studies to solve transportation science problems (*25–28*). This research aims to use a small data set to obtain a good prediction for a crash type classification problem using an advanced language model. An MAML algorithm is proposed and integrated with the BERT model for prediction purposes. The research team collected crash narrative data from three major cities in Texas and manually classified pedestrian crash typing by meticulously reviewing the crash narratives and examining the crash reconstruction diagrams. The data preprocessing was first completed to benefit the training efficiency and accuracy. The BERT model served as the learner in the MAML model and was used for encoding the text information. A three-way, five-shot crash type classification problem was formed for both the training and testing procedure. To evaluate the robustness of the proposed MAML model, four trials were formed with different configurations of training and testing sets. The testing results showed that the four configured trials had a similar prediction performance. The highest overall accuracy, 44.4%, came from trial no. 2. The lowest overall accuracy, 37.7%, came from trials no. 1 and no. 4. Overall, with only a limited number of data used in the training procedure, the accuracy of the testing set can be considered a good result. For a small sample problem, the proposed FSL can quickly update the model parameters based on the obtained pre-knowledge and further rapidly obtain a prediction result. Compared with the FSL algorithm, the traditional machine-learning algorithm might not be able to obtain the result for a small sample problem accurately and rapidly. The resulting framework provides a reproducible workflow for conducting crash narrative classification using FSL. The potential application of the proposed model will be classifying the crash narratives when only a small data set is available. Specifically, the proposed FSL is able to make a prediction using a very limited number of data samples of unseen crash narratives. In the real world, it is hard to always obtain a large enough data set to train a machine-learning model. At this point, the proposed FSL performs well.

Future methods for enhancing the performance of the MAML algorithm may include experimenting with other text-encoding algorithms and expanding the training data set size. In the current study, BERT was used to encode the text. However, there are many different algorithms and approaches that can be used to encode the text and serve as the learner for the MAML algorithm. Word2vec and one-hot encoding methods, for instance, can encode text and serve as input variables for a CNN. In this instance, the traditional neural network will serve as the MAML algorithm's learner. MAML techniques have the advantage of requiring less information about the samples in an unknown new class. However, they still require a substantial amount of data for training the model, acquiring generalized prior knowledge, and predicting categorization across several domains. In the future, it will be beneficial to increase the size of the training data set by adding more examples. Additionally, adding human knowledge to constrain the loss function used in FSL might be helpful to increase accuracy. Also, more current and advanced FSL algorithms might have better accuracy compared with the current result.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: S. Das, Y. Weng; data collection: S. Das, Y. Weng; analysis and interpretation of results: S. Das, Y. Weng; draft manuscript preparation: S. Das, Y. Weng, S Paal. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Yanmo Weng https://orcid.org/0000-0002-7436-1816
Subasish Das https://orcid.org/0000-0002-1671-2753
Stephanie German Paal https://orcid.org/0000-0002-0141-6679

References

1. NHTSA. *Traffic Safety Fact Sheets: Pedestrian (2020 Data)*. Report No. DOT HS 813 310. Washington, D.C., 2021.
2. Wang, Y., Q. Yao, J. T. Kwok, and L. M. Ni. Generalizing From a Few Examples: A Survey on Few-Shot Learning. *ACM Computing Surveys*, Vol. 53, 2020, pp. 1–34.
3. Federal Highway Administration. Develop Pedestrian and Bicycle Crash Analysis Tool. *Public Roads*, Vol. 63, No. 5, 2000, p. 54.
4. Thomas, L., D. Levitt, M. Vann, K. Blank, K. Nordback, and A. West. *PBCAT–Pedestrian and Bicycle Crash Analysis Tool Version 3.0*. Federal Highway Administration, Washington, D.C., 2021.
5. NHTSA. *2020 FARS/CRSS Pedestrian Bicyclist Crash Typing Manual: A Guide for Coders Using the FARS/CRSS Ped/Bike Typing Tool*. Washington, D.C., 2022, p. 86.
6. Vavrova, M., C. Chang, S. Kumar, D. Ruiz, S. Gonzalez, and M. D. Benitez. *North Texas Bicycle and Pedestrian Crash Analysis: Final Report*. Texas, 2021, p. 213.
7. Shah, N. R., S. Aryal, Y. Wen, and C. R. Cherry. Comparison of Motor Vehicle-Involved e-Scooter and Bicycle Crashes Using Standardized Crash Typology. *Journal of Safety Research*, Vol. 77, 2021, pp. 217–228. DOI: 10.1016/j.jsr.2021.03.005.
8. Das, S., M. Le, and B. Dai. Application of Machine Learning Tools in Classifying Pedestrian Crash Types: A Case Study. *Transportation Safety and Environment*, Vol. 2, No. 2, 2020, pp. 106–119. DOI: 10.1093/tse/tdaa010.
9. Lopez, D., L. C. Malloy, and K. Arcoleo. Police Narrative Reports: Do They Provide End-Users with the Data They Need to Help Prevent Bicycle Crashes? *Accident Analysis & Prevention*, Vol. 164, 2022, P. 106475. DOI: 10.1016/j.aap.2021.106475.
10. Schneider, R. J., and J. Stefanich. Application of the Location–Movement Classification Method for Pedestrian and Bicycle Crash Typing. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2601, 2016, pp. 72–83.

11. Chavis, C., Y.-J. Lee, and S. Dadvar. *Analysis of Bicycle and Pedestrian Crash Causes and Interventions*. District Department of Transportation, 2018, p. 368.
12. Kwayu, K. M., V. Kwigizile, and J.-S. Oh. Evaluation of Pedestrian Crossing-Related Crashes at Undesignated Midblock Locations Using Structured Crash Data and Report Narratives. *Journal of Transportation Safety & Security*, Vol. 14, 2022, pp. 1–23.
13. Wali, B., A. J. Khattak, and N. Ahmad. Injury Severity Analysis of Pedestrian and Bicyclist Trespassing Crashes at Non-Crossings: A Hybrid Predictive Text Analytics and Heterogeneity-Based Statistical Modeling Approach. *Accident Analysis & Prevention*, Vol. 150, 2021, p. 105835.
14. Snell, J., K. Swersky, and R. S. Zemel. 2017. *Prototypical Networks For Few-Shot Learning*. arXiv preprint arXiv:1703.05175.
15. Sung, F., Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.
16. Vinyals, O., C. Blundell, T. Lillicrap, and D. Wierstra. Matching Networks for One Shot Learning. *Advances in Neural Information Processing Systems*, Vol. 29, 2016, pp. 3630–3638.
17. Chen, Y., Z. Liu, H. Xu, T. Darrell, and X. Wang. 2021. Meta-baseline: Exploring Simple Meta-Learning for Few-Shot Learning. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9062–9071.
18. Finn, C., P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. International conference on machine learning, PMLR, 2017, pp. 1126–1135.
19. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is All You Need. *Advances in Neural Information Processing Systems*, Vol. 30, 2017, pp. 1–15.
20. Acheampong, F. A., H. Nunoo-mensah, and W. Chen, Transformer Models for Text-Based Emotion Detection: A Review of BERT-Based Approaches. *Artificial Intelligence Review*, Vol. 54, 2021, pp. 5789–5829.
21. Devlin, J., M. W. Chang, K. Lee, and K. Toutanova. *Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805, 2018.
22. Lim, J., I. Sa, H. S. Ahn, N. Gasteiger, S. J. Lee, and B. MacDonald, Subsentence Extraction from Text Using Coverage-Based Deep Learning Language Models. *Sensors*, Vol. 21, No. 8, 2021, p. 2712.
23. Bird, S., E. Loper, and E. Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.
24. Kingma, D. P., and J. Ba. Adam: A method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2014.
25. Das, S., A. H. Oliaee, M. Le, M. Pratt, and J. Wu. Classifying Pedestrian Maneuver Types using Advanced Language Model. *Transportation Research Record: Journal of Transportation Research Board*, 2023.
26. Hosseini, P., S. Khoshsirat, M. Jalayer, S. Das, and H. Zhou. Application of Text Mining Techniques to Identify Actual Wrong-Way Driving (WWD) Crashes in Police Reports. *International Journal of Transportation Science and Technology*, 2022. https://doi.org/10.1016/j.ijtst.2022.12.002
27. Kutela, B., K. Msechu, S. Das, and E. Kidando. (2023). *ChatGPT's Scientific Writings: A Case Study on Traffic Safety (January 19, 2023)*. SSRN: https://ssrn.com/abstract=4329120
28. Oliaee, A. H., S. Das, and J. Liu. Using Bidirectional Encoder Representations from Transformers (BERT) to Classify Traffic Crash Severity Types. *Natural Language Processing*, 2023.