

Classifying Pedestrian Maneuver Types Using the Advanced Language Model

Subasish Das¹ , Amir Hossein Oliaee² , Minh Le³ ,
Michael P. Pratt⁴, and Jason Wu³ 

Transportation Research Record
1–13

© National Academy of Sciences:
Transportation Research Board 2023
Article reuse guidelines:

sagepub.com/journals-permissions
DOI: 10.1177/03611981231155187

journals.sagepub.com/home/trr



Abstract

Pedestrians are the most vulnerable roadway user. While there is much emphasis on “green transportation,” a troubling fact emerges in the U.S.A.: pedestrian deaths are increasing significantly in comparison to motorist deaths, reaching nearly 6941 in 2020—the highest in over two decades. The Pedestrian and Bicycle Crash Analysis Tool was developed to determine motorists and non-motorists’ actions before a crash to accurately define the sequence of events and precipitating actions leading to traffic crashes between motor vehicles and pedestrians or bicyclists. Police report traffic crash data and crash narrative reports undoubtedly play a major role in decision-making for the safety engineers. Using crash data from three major cities in Texas (2018–2020), this study assessed the data quality of text narratives in police reports of pedestrian crashes. The objective of this study was to develop a framework for applying advanced language models to classify pedestrian maneuver types from unstructured textual content. The results show that although natural language processing models are promising as crash typing tools, narration inconsistency, data imbalance, and small sample sizes are holding back progress in this area. The framework demonstrated high accuracy for the binary classification task, but it was inconsistent for the more complex multiclass task. This framework provides the basis for applying advanced language models such as the bidirectional encoder representations from transformers model in identifying pedestrian maneuver types associated with pedestrian crashes.

Keywords

natural language processing, advanced language modeling, crash typing, text mining, pedestrian and bicyclist safety

The issue of pedestrian safety is of utmost importance in the field of traffic safety. Pedestrians are mostly unshielded from forces in accidents. This motivates the exploration of accidents involving pedestrians to better understand the topic and produce solutions. The task of crash typing is a significant yet time consuming step toward such understanding. This study presents a crash typing framework based on natural language processing (NLP) to streamline this tedious process. The study expands on the compatibility of current police report narration style and data collection practices to provide recommendations for a more effective assessment and deployment of the framework.

In the recent years, non-motorized trips have been increasing and so have the injuries associated with non-motorists. According to the National Highway Traffic Safety Administration (NHTSA) (1), there were 6516 pedestrians killed in U.S traffic crashes in 2020, a 3.9% increase from the 6272 pedestrian fatalities in 2019 (2).

This equates to a pedestrian killed every 81 min and in 17% of all traffic fatalities. As expected, more occurred in urban areas (82%) than rural areas (18%) and 71% of pedestrian fatalities were males. This mandates further investigation on pedestrian crashes.

There has been an increasing focus on pedestrian-related crashes because of these alarming trends. As part of this recent attention, tools such as the Pedestrian and Bicycle Crash Analysis Tool (PBCAT) have been developed to assist with collection of data related to crashes

¹Civil Engineering, Ingram School of Engineering, Texas State University, San Marcos, TX

²College of Architecture, Texas A&M University, College Station, TX

³Texas A&M Transportation Institute, Dallas, TX

⁴Texas A&M Transportation Institute, College Station, TX

Corresponding Author:

Subasish Das, subasish@txstate.edu

between motor vehicles and pedestrians or bicyclists. As stated in PBCAT 3.0 (3), “Crash data are a primary data source for analyzing and understanding these crash risks. However, crash data are often not as complete or descriptive for crashes involving non-motorists as for crashes that involve only motorists.” The purpose of this study was to gain a better understanding of the use of police-reported crash report content, particularly from crash narratives, in identifying PBCAT Version 3.0 crash typing. This research introduces the application of bidirectional encoder representations from transformers (BERT), an advanced language model, in classifying pedestrian maneuver types.

After the introduction, the rest of this paper is organized as follows. The second section summarizes related work and the third section presents the data preparation work and descriptive statistics. In the fourth section, the PBCAT Version 3.0 crash typing approach and the concepts of NLP-related machine learning algorithms are briefly explained. The results and findings are reported in the fifth section. The study is concluded in the sixth section with remarks on future research.

Literature Review

The subject of pedestrian safety has been the subject of rigorous studies, resulting in a better understanding of factors contributing to accidents. This body of research has laid the foundation for emerging challenges, such as e-scooters, that often blur the line between pedestrians and vehicle operators. To make a meaningful contribution, some of the relevant literature is reviewed to address the current gaps in research and practice that merit attention in the course of this research.

In 2000, the Federal Highway Administration (FHWA) and NHTSA started a software product, known as PBCAT, which can assist in analyzing details of non-motorist crashes so that solutions and countermeasures can be easily determined (4). PBCAT was developed to describe the pre-crash actions of the parties involved to define event sequences and precipitating actions that lead to traffic crashes between vehicles and pedestrians or bicyclists. PBCAT Version 3.0, the most recent version of PBCAT (3), has been updated and modified, and this product aims to help traffic safety professionals in attaining better knowledge of the causation patterns so that interventions can be designed appropriately. The tool includes pre-drawn diagrams of various situations and dropdown menus to obtain a deeper understanding of what police officers collect in the field about pedestrian/bicycle crashes (5). NHTSA has recently released the coding manual on crash typing (6). However, categorization tools such as PBCAT cannot compensate for crash narrative

insufficiency. This has merited an exploration into whether current narrations are effective descriptors, whether language models can be used to perform this task, and what steps could be taken to improve their comprehension. It is important to note that crash narrative reports are mostly handwritten or typed in a pdf file. Many state Departments of Transportation (DOTs) do not digitalize this information in a searchable database or structured database, such as Access or a spreadsheet. The digitization work often requires removing personal identifiable information (PII).

The study of PBCAT-related analysis has increasingly drawn the attention of the transportation safety community. A recent study used pedestrian and bicyclist crash data from North Central Texas by coding the crashes according to PBCAT (version 2.0) methodology (7). This study developed a pilot web application tool that can illustrate the results to communicate safety needs in North Central Texas. Shah et al. (8) examined 52 e-scooter and 79 bicycle police-reported crashes during 2018–2020 from the Tennessee Integrated Traffic Analysis Network (TITAN) database using PBCAT (version 2.0). This study found statistically significant differences in spatiotemporal distribution, lighting, distance from home to crash location, and demographics in bicycle and e-scooter crash data. The investigation into police crash narrative reports provided a comprehensive picture of e-scooter safety that enhanced the state-of-the-art literature. The data quality of text narratives in police reports on bicycle accidents was evaluated by Lopez et al. (5). Police reports were compared to PBCAT (version 2) to determine the breadth and depth of the narrative textual contents. The findings revealed that the police reports only captured most of the information in one section of the standardized form (crash typing), with an average total missingness of more than 75%. The findings also revealed that while longer reports result in less missing information when compared to the standardized crash form, the average report still has a lot of missing information. Schneider and Stefanich (9) introduced the location-movement classification method (LMCM) to classify pedestrian and bicycle crashes, and demonstrated that the LMCM provides useful information that PBCAT does not capture. Both typologies were applied to 296 pedestrian and 229 bicycle crashes in Wisconsin reported between 2011 and 2013. Chavis et al. (10) analyzed three years of pedestrian and bicycle crashes (2012–2014) in Washington, D.C. The classification was based on PBCAT (version 2.0) and LMCM crash typologies. The NHTSA crash groups were identified, and the top three groups were examined in detail with applicable PEDSAFE and BIKESAFE countermeasures. The inadequacies of police crash report forms that were used were also discussed.

Table 1. Pedestrian Injuries by City

City	2018–2020					Three-year average	
	KABCO	KAB	2020 Pop.	KABCO/100k Pop.	KAB/100k Pop.	KABCO/100k Pop.	KAB/100k Pop.
Austin	1109	778	961,855	115.30	80.89	38.43	26.96
Dallas	1736	1139	1,304,379	133.09	87.32	44.36	29.11
El Paso	579	343	678,815	85.30	50.53	28.43	16.84
Fort Worth	689	434	918,915	74.98	47.23	24.99	15.74
Houston	2875	1595	2,304,580	124.75	69.21	41.58	23.07
San Antonio	1847	1171	1,434,625	128.74	81.62	42.91	27.21
Total	8835	5460	7,603,169	116.20	71.81	38.73	23.94

Note: K = fatal; A = serious injury; B = moderate injury; C = minor injury; O = no injury or PDO; Pop. = population.

Pedestrian- and bicyclist-involved crashes are typically not reported in detail in state or national crash databases (6). Crash narratives in police reports undoubtedly play a major role to provide PBCAT end users with the data they need to gain deeper insights into factors contributing to crash occurrence. However, it is labor-intensive to manually identify the crash types coded in PBCAT from these huge textual crash narrative datasets. There is an urgent need to utilize text mining analytics to discover crash types from these unstructured textual content in the police reports. Many text mining applications, such as thematic analysis, content analysis, supervised modeling, unsupervised modeling, and NLP, can be used to extract insights from crash narrative textual data (11). For example, Kwayu et al. (12) used crash narratives to conduct semantic analysis and classification of drivers' risky behavior at signalized intersections. The proposed algorithm correctly identified "disregard traffic control" and "fail to yield" hazardous actions from crash narratives. Furthermore, the developed textual-based algorithm demonstrated promise in detecting potential errors made by police officers while coding hazardous actions in crash reports (12). Das et al. (13) formed a framework for using machine learning models to classify crash types from unstructured textual content. For this study, the research team gathered pedestrian crash typing data from two Texas locations. The XGBoost model was discovered to be the best classifier. Kwayu et al. (12) also utilized crash narratives complemented by crash metadata to discern the prevalence and co-occurrence of themes that contribute to crash incidents by using text mining analytics and network topology (14).

Studies on the text mining of crash narratives in police reports (from a PBCAT-related perspective) are still limited. This study developed a unique framework for applying advanced language models to classify pedestrian maneuver types from unstructured textual content. This framework can be applied to other transportation safety-related classifications from textual contents.

Methodology

Data Preparation

This study collected 2018–2020 pedestrian crash data from the Texas Department of Transportation's (TxDOT's) crash record information system (CRIS) for six major Texas cities: Austin, Dallas, Fort Worth, Houston, San Antonio, and El Paso. This study selected the three major cities with the highest KAB/100k population (K = fatal, A = serious injury, B = moderate injury), Austin, Dallas, and San Antonio, as shown in Table 1, and extracted the police recorded crash narratives.

This study assessed the collected data's suitability for the classification task. A total of 4442 crash data were collected with crash narrative information. Table 2 provides descriptive statistics for the full dataset. The year 2020 shows a slight reduction in crashes compared to 2018 and 2019, perhaps because of COVID-related restrictions. Almost all the crashes involved one pedestrian. As expected for pedestrian–vehicle crashes, very few of the crashes are property damage only (PDO). 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.

Roughly half of the crashes did not have contributing factors coded in their corresponding unit (vehicle or pedestrian) records. Among the crashes that did have coded contributing factors, the most common factors were driver failed to yield to pedestrian (15.51% of crashes), driver inattention (12.79%), or speed-related (3.65%). "Speed-related" is a combination of the following three code values in the database: failed to control speed, speeding—unsafe (under limit), and speeding—(overlimit).

PBCAT Crash Typing

The main goal of this research has been to develop an advanced NLP framework to address the crash typing

Table 2. Descriptive Statistics for the Full Dataset

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	City	Other	6.6
	A	18.73		Austin	23.73
	B	38.9		Dallas	36.99
	C	27.94	Light condition	San Antonio	39.28
	PDO	4.21		Day	48.56
Contributing factor	Unknown	0.18		Dark, lighted	35.64
	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	Relation to intersection	Unknown	0.07
	Other	16.82		At intersection or near at intersection	45.73
				Driveway access	4.89
Number of pedestrians	1	95.27		Non-intersection	49.39
	2	4.07			
	≥ 3	0.65			

Note: K = fatal; A = serious injury; B = moderate injury; C = minor injury; O = no injury or PDO; PDO = property damage only.

issues associated with pedestrian crashes. The framework was applied to a binary classification task (intentionality) and a multiclass classification task (pedestrian maneuver/action). The classification tasks were selected based on four factors. Firstly, the availability of data. Secondly, the logical possibility of direct inference of the crash type based on narratives without accessing external information. Thirdly, the presence of samples of all types in the dataset. Fourthly, the complexity of the task and its potential for demonstrating the advanced language model's applicability. The capability of the model in performing such a task could help identify the main challenges for developing a more general-purpose model in the field of transportation safety.

Toward this goal, a transfer learning approach was undertaken using the pre-trained BERT model and its ability to create state-of-the-art models without the need for substantial task-specific architecture modifications (15). The Transformers open-source library (16) was used to develop the crash severity classification model. After data preparation and initial preprocessing, the tokenized narrative data, segment embeddings (a learned numerical representation of text in form of a sequence of numbers), and position embeddings were encoded using the Transformer tokenizer's encode function (Step 2: Figure 1). This encoding function produces a compatible input for the pre-trained BERT model. The uncased BERT_{BASE} was then extended by adding a linear transformation layer corresponding to the one-hot-encoded

representation of non-motorist maneuver classes (CL = crossing path from motorist's left; CR = crossing path from motorist's right; CU = crossing path, unknown direction; PO = parallel path opposite direction; ST = stationary; PU = parallel path unknown direction; PS = parallel path same direction) and intentionality classes ("intentional" and "unintentional") (Step 3: Figure 1). The model hyper-parameters were then modified based on the result obtained from training and testing data subsets (Step 3: Figure 1). The model was then trained, tested, and validated for the two specified tasks (Step 4: Figure 1). The final performance metric-based evaluation of the model was performed using the test and validation subsets (Step 5: Figure 1).

Intended Versus Unintended Pedestrians

Pedestrians are generally defined as any person on foot, walking, running, jogging, hiking, sitting, or lying down that is involved in a motor vehicle crash. However, there seems to be a need to clarify the definition to exclude persons coded as "pedestrians" but associated with an on-scene vehicle or working at the crash scene (e.g., construction workers or emergency responders) that were not at the scene as an intended pedestrian (i.e., to cross or walk along the roadway). This distinction becomes more critical if the increasing "pedestrian" crash trend is the basis of future policies, programs, and funding. Researchers have developed a process to determine

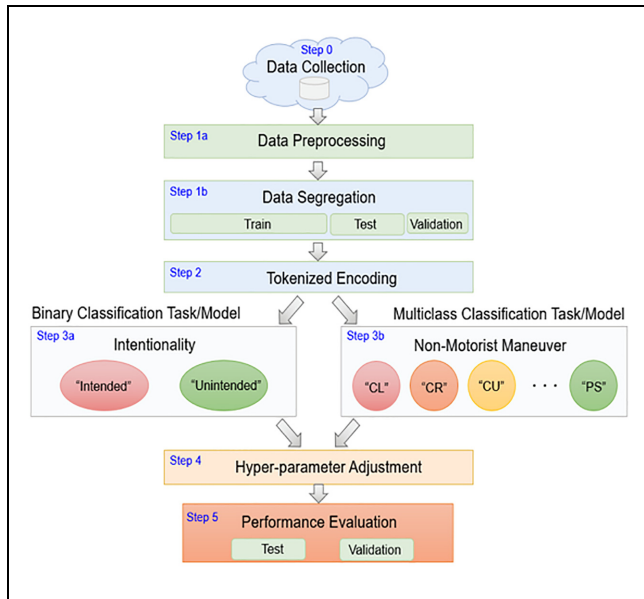


Figure 1. Pedestrian and Bicycle Crash Analysis Tool crash typing framework.

Note: CL = crossing path from motorist's left; CR = crossing path from motorist's right; CU = crossing path, unknown direction; PS = parallel path same direction.

whether the non-motorist person was an “intended” or “unintended” pedestrian. This was done by determining the crash circumstance (i.e., why the pedestrian was at the crash scene). Note that the “intension” measure considered only pedestrian’s point of view (not the driver’s point of view). Generally, the pedestrian was deemed “unintended” if they were associated with a vehicle, as shown in Table 3. Otherwise, they were considered as an “intended” pedestrian because they were likely at the scene intending to walk or cross the roadway.

Narrative Data Preprocessing

This study collected the selected number of crashes from TxDOT, and TxDOT provided the crash narratives by removing all PII from the narratives. As manual crash typing requires significant effort, this study randomly selected 500 crash reports for classifying the crash typing manually. Later, four crash reports were discarded because of inadequate narrative information. This study started the process with crash typing (following the PBCAT 3.0 crash typology) and later considered only pedestrian maneuver types as the response, as the driver maneuver can be determined from the “collision type” column in the structured crash database. Although PBCAT 3.0 has set a standardized crash type classification protocol and significantly improved the integration of different data sources, streamlining the process, it inherently relies on human operators for text mining to

Table 3. Pedestrian Intentionality Matrix

Circumstance	Pedestrian intention
Changing seat positions	Unintended if associated with a vehicle; intended if not
Commuting/moving from (one place to another)	
Crossing roadway, crossing street	Unintended
Fleeing police	
Jumping from bridge	
Jumping from car	
Retrieving items from road	
Standing in traffic	
Standing on median, shoulder, or off the road	
Suicide	
Taking pictures	
Unconscious	
Walking along the sidewalk	Unintended
Walking or lying down in traffic	
Walking or lying down on median, shoulder, or off the road	
Previous crash	
Stalled or stopped vehicle	
Working	Unintended
Sitting at bus stop	

perform its tasks. The applicable NLP models capable of automating the process of text mining and classification are often adversely affected by the narration format or stylistic choices of the narrators. Early testing indicated that the basic narrative structure was helpful for the model’s inference capability, most likely because of the relatively small sample size. As a result, the following limited preprocessing steps were undertaken on the crash narrative sentences with the goal of not disturbing the underlying structure.

- **Replacement of specialized characters:** less common control characters and more familiar characters, such as apostrophes and punctuation marks, were replaced with whitespaces.
- **Removal of numbers:** the crash narratives included numerical characters that often represent information such as case numbers or dates. However, not all numbers were removed. Single-digit numbers were preserved because they were most often used to distinguish subjects of the narratives.
- **Removal of extra whitespaces and newlines:** most often, chains of whitespaces were formed because of the previously mentioned preprocessing

Table 4. Training Dataset by City and Access Type

Access	Austin	Dallas	San Antonio	Total	Percent of total
Controlled	7	27	11	45	9
Non-controlled	107	156	192	455	91
Total	114	183	203	500	100
Percent of total	23	37	41	NA	NA

Note: NA = not available.

Table 5. Training Dataset by Crash Type

Veh/Ped	CR	CL	CU	PS	PO	PU	MU	ST	OU	UN	FC	Total
Straight	99	87	31	25	5	3	6	30	8	13	na	307
Right	26	6	2	9	5	4	na	3	na	na	na	55
Left	4	5	1	29	38	14	na	4	1	2	na	98
Parked	na	na	1	na	na	na	na	na	2	na	na	3
Entering	1	na	na	na	na	na	na	2	na	na	na	3
Backing	1	1	na	na	na	na	1	6	2	1	na	12
Other	1	1	na	na	na	na	na	5	1	na	na	8
Unknown	na	na	2	na	na	na	na	2	na	1	na	5
Non-collision	na	na	na	na	na	na	na	na	na	na	5	5
Total	132	100	37	63	48	21	7	52	14	17	5	496

Note: CR = crossing path from motorist's right; CL = crossing path from motorist's left; CU = crossing path unknown direction; PS = parallel path same direction; PO = parallel path opposite direction; PU = parallel path unknown direction; MU = moving in unknown path/direction; ST = stationary; OU = other/unusual; UN = unknown; FC = non-motorist fall or crash; veh/ped = vehicle/pedestrian; na= not applicable.

measures or the original narrator's stylistic choices. Narrators mostly used newline characters for stylistic purposes.

- **Lowercasing of the characters:** the BERT model that serves as the backbone of the proposed classifier has been pre-trained on lowercase text, so characters had to be lowercased before use.
- **Tokenization:** for the input data to be compatible with the pre-trained BERT model, this study tokenized the preprocessed narrative data using the BERT tokenizer that was originally used to pre-train the model. Because the tokenized representation of the crash narratives often exceeded the input limit of the model (512 tokens), the initial 510 tokens were used. A [CLS] was then added to the beginning of the token sequences to mark the classification task, and [SEP] was added to the end, marking the end of each token sequence.
- **Data splitting:** the dataset was randomly split into three subsets for training (70%), testing (20%), and validation (10%) purposes. The training dataset was used to adjust model weights, while the development dataset was used to assess model performance, adjust the hyper-parameters, and perform early stopping. The validation dataset was used to validate the model's performance.

Researchers developed a training dataset from the 4442 crashes, of which 500 crashes were randomly selected from the three cities and roadway access type, as shown in Table 4.

The 500 crash narratives were carefully reviewed and binned into the appropriate crash type per PBCAT's 3.0 user guide. Table 5 summarizes the crash types found in the test dataset. Note that four crashes could not be crash typed because they did not involve a pedestrian or the pedestrian involved was not injured/hit.

BERT Model

The BERT model (15) is a conceptually simple yet powerful language representation model. The original model was developed in the two variants of BERT_{BASE}, consisting of 12 transformer blocks and 768 attention heads, and BERT_{LARGE}, consisting of 24 transformer blocks and 1024 self-attention heads. The model's transformer block architecture (Figure 2) was originally introduced by Vaswani et al. (17), allowing a fusion of the left- and right-hand context. The transformer blocks are, as demonstrated in Figure 2, the building blocks of the BERT and many other language models. Considering that attention heads are neural network segments that prominently featured in modern NLP models, their

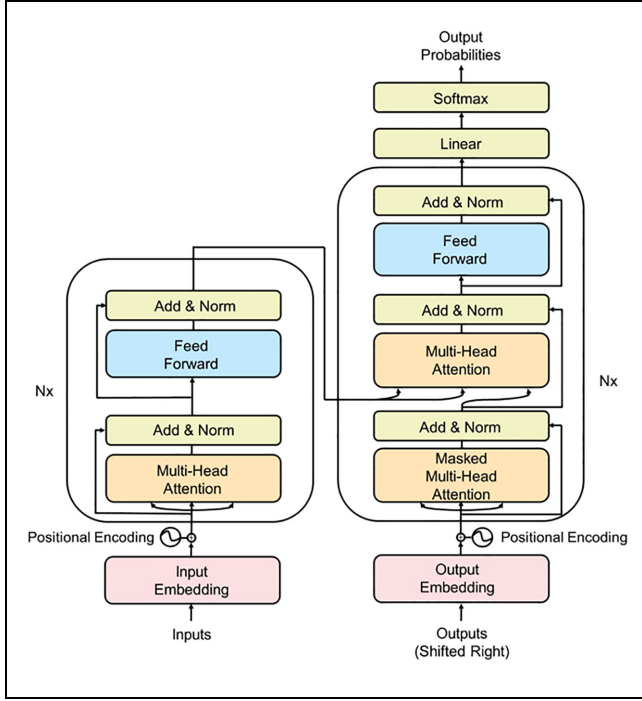


Figure 2. Transformer architecture.

definition and function as a part of the BERT language model is outside the scope of this research. However, the number of transformer blocks and self-attention are provided as a general descriptor of the model's structure. The BERT_{BASE} that has been used in this research has been pre-trained using the following unsupervised tasks.

- **Masked language modeling (MLM):** in MLM, also known as the Cloze task (18), a random percentage of the input tokens are masked, and model is trained to predict the masked word. In the case of BERT, 15% of all word piece tokens in each sequence are masked.
- **Next sentence prediction (NSP):** the NSP task trains the model to understand the relationships between two sentences. This relationship, which is not directly captured by language modeling, is instilled in the model by pre-training the model for binarized NSP. In the BERT implementation of this task, the model transfers all parameters to initialize end-task model parameters.

Results and Findings

Experimental Settings

The Adam optimization algorithm (19) was used in the training process to iteratively update the network weights for the classifier portion of the network. This study used

the cross-entropy loss function to train the model. This function is calculated as follows:

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^T,$$

$$l_n = -w_{y_n} \log \frac{\exp(x_{n, y_n})}{\sum_{c=1}^C \exp(x_{n, c})}$$

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 following hyper-parameters were used in the experiments:

- dropout rate of the classification layer: 0.;
- learning rate: 0.00002;
- epochs: 20;
- batch size: 16;
- warmup steps: 10%.

Word Embeddings

Text classification models, much like other mathematical models, perform a task based on numerical input. The process of converting raw text to a numerical vector representation, also known as encoding, varies based on the model's architecture and the intended task. However, some of the most notable text encoding methods include Word2Vec (20), Global Vectors (GloVe) (21), and FastText (22). Although a primitive encoding serves the simple purpose of text-to-vector conversion, the more recent methods improve the model's inference capability by creating token embeddings instead of whole word tokens and, most importantly, performing the conversion in a context-aware manner. The BERT model's input is made of an elementwise sum of three embeddings, which are sequences of numbers. The first sequence is a sequence of Wordpiece (23) tokens, which are a representation of the input text. The second sequence is the segment embeddings, which indicate which sentence each token belongs to. In this context of this research, a sentence is an "arbitrary span of contiguous text" (15) and the values in this sequence are zero because of the type of task (classification). The third sequence is position embeddings, which signify the position of each Wordpiece token. The resulting input of the BERT model is a sequence of 512 numbers. Much like the input, the BERT model's output is a vectorized representation of the text that is referred to as BERT embeddings. The BERT_{BASE} that was used in this research produces a sequence of 768 embeddings.

Performance Metrics

Performance metrics quantify the performance of models. The following classification metrics were used to

assess the model's overall viability as a classifier and its performance (24).

- **Accuracy (acc):** the prediction accuracy of a model in the context of classification is the ratio of correct predictions over the total number of examined instances. The accuracy is given as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

- **Precision (p):** precision measures positive patterns that are correctly predicted over the total positive prediction patterns. Precision is calculated for each class separately, so macro average precision is weighted averaged precision that is calculated as a holistic measure of the model's precision. These metrics are given as follows:

$$Macro\ Averaged\ Precision = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + FP_i}}{l} \quad (2)$$

$$Weighted\ Averaged\ Precision = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + FP_i} \times n_i}{l} \quad (3)$$

- **Recall (r):** recall is a measure of positive patterns over the total correct predictions. Much like precision, recall is calculated for each class, and thus averaging is required for multiclass model assessment. This metric, its macro average, and weighted average are given as follows:

$$Macro\ Averaged\ Recall = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + TN_i}}{l} \quad (4)$$

$$Weighted\ Averaged\ Recall = \frac{\sum_{i=1}^l \frac{TP_i}{TP_i + TN_i} \times n_i}{l} \quad (5)$$

- **F-measure (F1):** the F -measure is the harmonic mean between recall and precision values. The macro-averaged F -measure and weighted averaged recall are calculated using macro-averaged and weighted averaged recall and precision, respectively:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

Here, the samples classified by the model as positive that are negative are denoted as FP , and samples that the model classified as negative but are positive are denoted as FN . The samples that the model correctly predicted as

positive and negative are denoted as TP and TN , respectively. The l represents the number of classes.

- **Area under the receiver operating characteristics curve (AUROC):** the AUROC is a well-known measure for classification problems. The receiver operating characteristics (ROC) curve displays the relationship between the true positive rate (TPR) and false positive rate (FPR) of a model. The ROC is independent of classification thresholds, and the AUROC reflects the overall ranking performance of the classifier (24, 25). In this research, the one versus rest (OvR) method was used to produce the ROC curves, reducing the classification output into binary classification metrics for further evaluation:

$$TPR = \frac{TP}{TP + FN} \quad (7)$$

$$FPR = \frac{FP}{FP + TN} \quad (8)$$

$$AUC = \frac{S_p - n_p(n_n + 1)/2}{n_p n_n} \quad (9)$$

Here, S_p is the sum of all positive samples ranked, while n_p and n_n denote the number of positive and negative samples, respectively.

As previously stated, the main objective of this study was to develop a framework for applying machine learning for typing from unstructured textual content. In this section, the crash typing performance of the proposed framework for the task of binary and multiclass classification is assessed and elaborated on. The early stopping criteria for both the multiclass and binary classification task is the macro-averaged F -measure for the test subset. This was done to reduce the model's emphasis on a single class. The observations suggest that the learning process did not end by achieving the minimum loss. After the minimum loss was reached, a trade-off between desired loss (lower) and the macro-averaged F -measure (higher) was observed (see Figure 3).

The assessment of classification performance metrics requires a combination of metrics to be considered to discriminate between task-specific models. The accuracy for the binary classification task (at the seventh epoch) is 89.9% for the test subset and 86% for the validation subset (Figure 4). The similarity between test and validation with respect to the binary classification model suggests that the model can perform the task to an extent; however, its capability is hindered by the dataset imbalance. The model's emphasis on the majority classes could be observed in the confusion matrix (Figure 5).

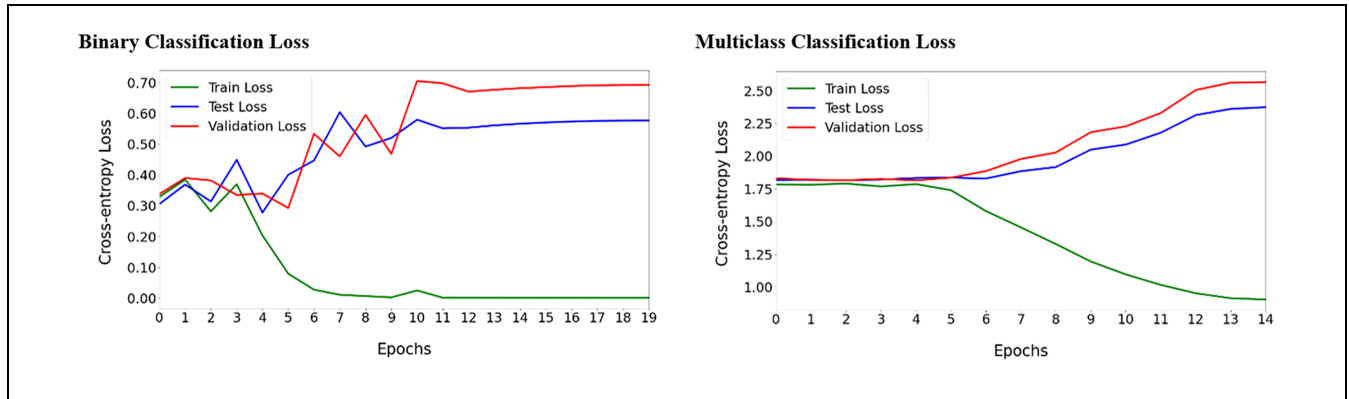


Figure 3. Binary and multiclass model loss.

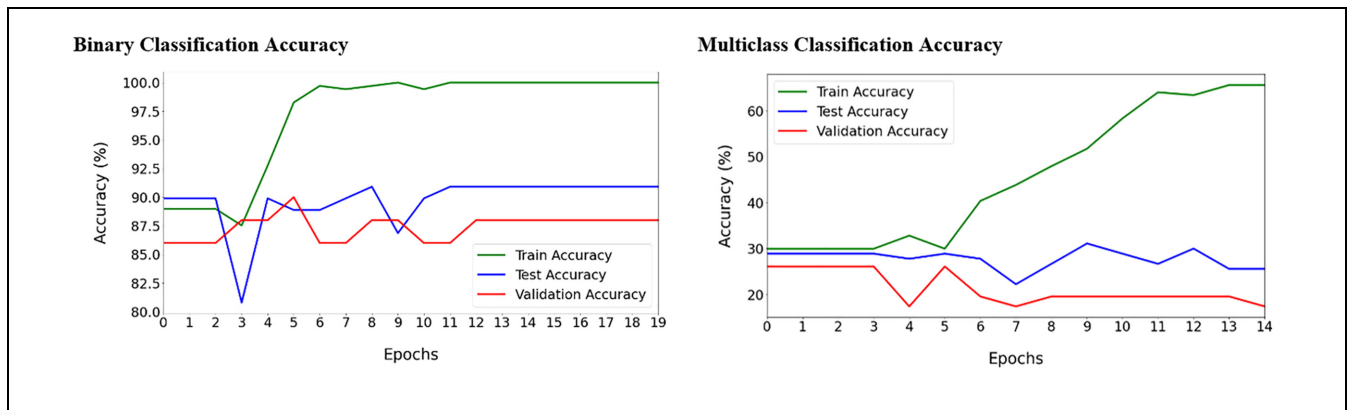


Figure 4. Binary and multiclass model accuracy.

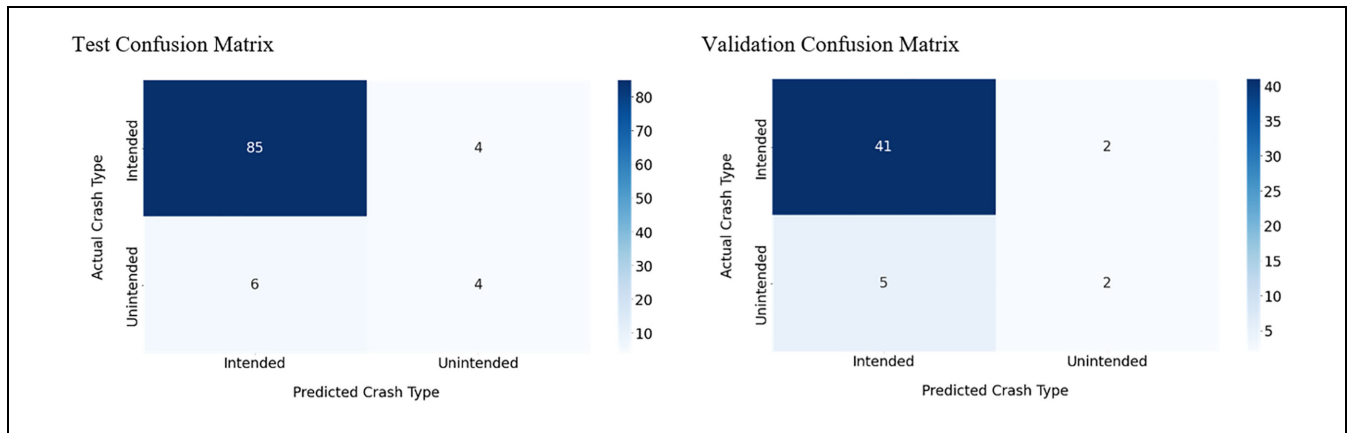


Figure 5. Binary model confusion matrix.

Based on the availability of data, the pedestrian maneuver type, as a multiclass classification problem, was mainly chosen for four reasons. Firstly, there had to be narrative data available for development of the model. Secondly, the class types had to be logically and directly

inferable based on the crash narrative without accessing information outside the narratives. Thirdly, all types had to be present in the dataset. Fourthly, the task had to possess the complexity to benefit from the features of an advanced language model and could not be simply

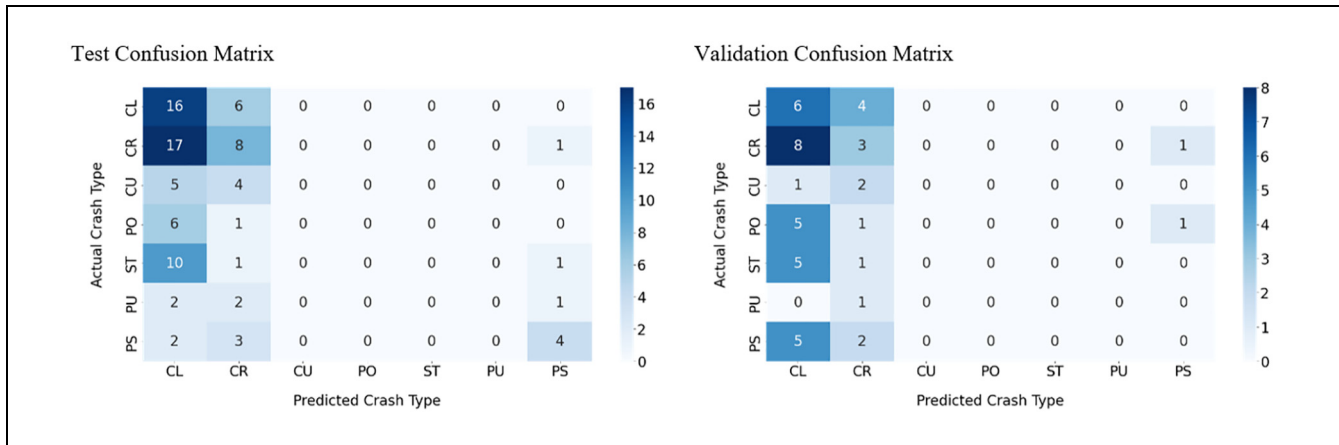


Figure 6. Multiclass model confusion matrix.

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

Table 6. Model Test and Validation Performance for Combined Data

Task	Binary model		Multiclass model	
	Test	Validation	Test	Validation
Support	99	50	90	46
Accuracy	89.90%	86.00%	31.11%	19.57%
Macro precision	71.70%	69.57%	16.68%	5.92%
Macro recall	67.75%	61.96%	21.13%	12.14%
Macro F1	69.44%	64.25%	17.34%	7.58%
Weighted precision	89.02%	83.65%	21.70%	9.94%
Weighted recall	89.90%	86.00%	31.11%	19.57%
Weighted F1	89.39%	84.33%	23.84%	12.54%

performed using more rudimentary methods, such as term frequency. The capability of the model in performing such a task could help identify the main challenges for developing a more general-purpose model in the field of transportation safety.

The accuracy for the multiclass classification task (for epoch 9) is 31.11% for the test subset and 19.57% for the validation subset. The difference between the test validation results and in different replications of the tests suggests that achieving generalization and assessment of the multiclass model is not possible. For instance, the model's confusion matrix displays that four of the crash types are ignored by the model for both the test and validation dataset (Figure 6). The highly limited sample size negatively affects the replicability of the multiclass model tests.

The model's macro-averaged F -measure for the binary classification is 69.44% for the test subset and 64.25% for the validation subset. These values corroborate that despite the high weighted F -measure of the model for this task, the per-class performance is not as high. It is expected that reducing the imbalance in the dataset could

increase the macro-averaged F -measure for this task. As expected, the multiclass task yielded poorer results. The same overall sample size means that the number of samples for each class is reduced for this task. During most of the training process for the multiclass task, the model identified all the samples as the majority class ("CR").

Although the use of a macro-averaged F -measure as the early stopping criteria improved the results for the model's performance for the binary classification task, it reduced the accuracy for the multiclass task without any significant improvement in the validation macro-averaged $F1$ -measure. The combination of class imbalance and small sample size resulted in poor multiclass performance and inconsistency in results (Table 6). It is recommended that the use of cross-validation methods for future crash typing research that utilize small or imbalanced datasets to perform tests. In cross-validation, instead of performing validation through a third subset that is used to replicate the tests, successive rounds of tests give each data point a chance of being validated against (26).

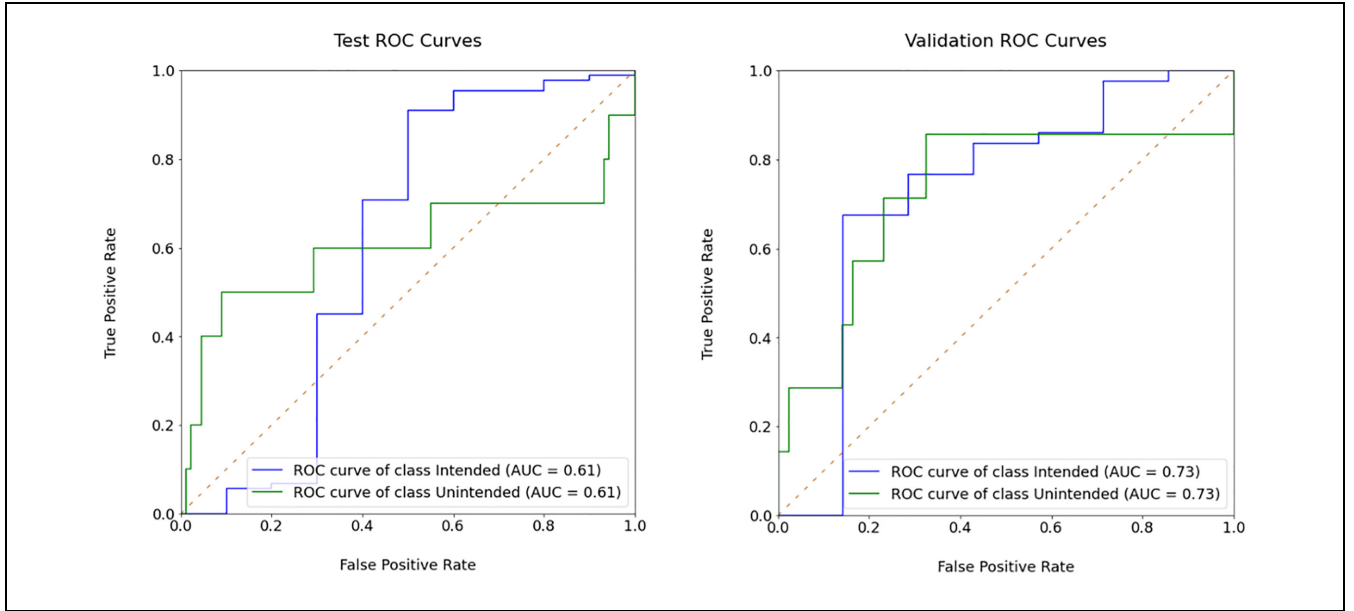


Figure 7. Binary receiver operating characteristics (ROC) curves.
Note: AUC = area under the curve.

In contrast to the previous metrics that evaluate the model based on the model's class predictions, the ROC is drafted and the AUROC is calculated based on the model's predicted probability of class membership. The model's OvR AUROC performance for the binary classification task is 0.61 for the test subset and 0.73 for the validation subset (Figure 7). This indicates that the model can discriminate between classes. The OvR AUROC for the multiclass task ranges from 0.36 to 0.76 for the test subset and from 0.11 to 0.68 for the validation subset. This indicates that the model is a poor discriminator for the task of multiclass classification (Figure 8).

Based on the previous analysis, it can be concluded that because of the limited available data and the inherent imbalance between crash types, the framework is only capable of performing less complicated classification tasks with high accuracy. Beyond the overall accuracy, the macro-averaged F -measure could be considered as a stopping criteria for future research. Because of the observed importance of maintaining the narrative structure in the preprocessing stage, exploration of a uniform narration guideline and its effects on classification performance is recommended for future research. Even though the assessment of tactics to reduce the impact of data imbalance on the model's performance is outside the scope of this research, it remains an important topic for future exploration. The aforementioned macro- and micro-averaged metrics are recommended for further exploration of data imbalance mitigation tactics, and the results could serve as a baseline point of comparison. In

addition, a more advanced language model with higher accuracy can provide justification of using automated PBCAT category generation, which can later be adopted in guidance documents such as the Model Minimum Uniform Crash Criteria (MMUCC).

Conclusions

This study aimed to develop a robust NLP framework for the crash typing task. The proposed framework's architecture uses a pre-trained BERT_{BASE} embedding network and a fine-tuned linear transformation layer. The model was developed using the PBCAT data from three major cities in Texas. The elevated level of noise in the narrative data necessitated several preprocessing measures that included removing specialized characters, multi-digit numbers, and extra whitespaces, lowercasing the narrative texts, and tokenization. The tokenization was performed using the tokenizer function that was used to pre-train the original BERT model, and a maximum of 512 initial tokens were used from each narrative text. The model was trained using 70% of the data, 20% were used to adjust the hyper-parameters, and the remaining 10% were used to validate the findings. The framework was evaluated for the task of binary classification (intentionality) and multiclass classification (pedestrian maneuver). The model's accuracy for the binary classification was 86%, with the weighted average F -measure of 84.33% and macro-averaged F -measure of 64.25%. The model's accuracy for the multiclass classification was 19.57%, with the

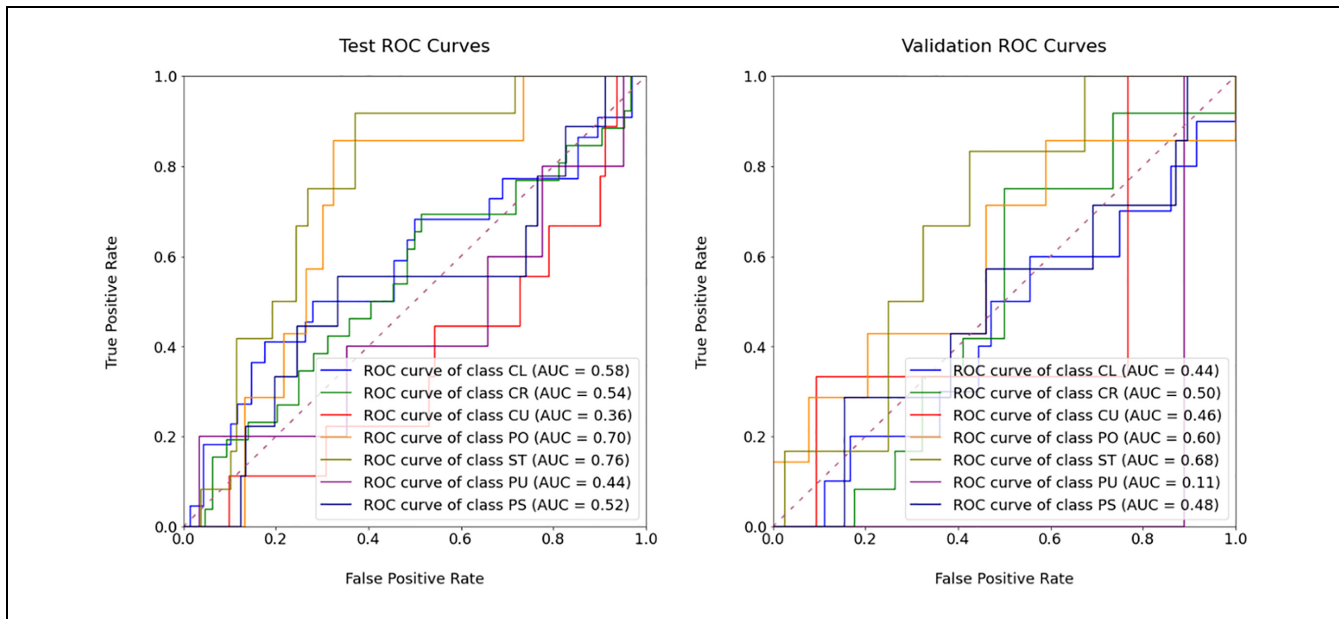


Figure 8. Multiclass receiver operating characteristics (ROC) curves.

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

weighted averaged F -measure of 12.54% and macro-averaged F -measure of 7.58%. The resulting evaluations demonstrate the framework's potential for the binary classification task; however, the results are inconsistent for the multiclass classification task.

Currently, millions of crash narratives are available from various crash databases across the U.S.A. However, the crash narrative data, like any other unstructured data, offers no or little insight and cannot be utilized or incorporated into decision-making processes. PBCAT, which includes pre-drawn diagrams of various situations and dropdown menus, has provided a potential way to index and store the unstructured crash narrative data in an organized fashion. Combining PBCAT crash typing and textual data mining techniques can pave a new way to automatically process crash narrative data from the unstructured to the structured, and further extract the useful crash attributing information underlying them. The findings of this research demonstrates that the small size of the datasets, the inconsistent narration styles, and extreme imbalance could be some of the significant factors preventing the development of more complex crash typing models. For future research, it is recommended to develop NLP models with more exposure to crash narratives or similar reports in their pre-training stage. The use of cross-validation evaluation methods is recommended in the future to prevent the inconsistencies observed in the evaluation of multiclass tasks. This study recommends the macro- and micro-averaged metrics and the

AUROC metric used in this study in future research to explore data imbalance mitigation tactics.

This study has limitations, too. The assessment of efficacy of various methods that can be reduce the impact of data imbalance on the classification capability of NLP models in regard to crash narrative data is an interesting topic that unfortunately was beyond the scope of this research. However, this research provides a report of metrics such as macro and weighted $F1$, recall, and precision. These metrics, unlike accuracy, demonstrate the model's averaged classification performance for each class. These performance metrics could be used in future research to assess the efficacy of various techniques in reducing the negative effects of data imbalance. ROC curves are most often used to assess the performance of binary classification models.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: S. Das; data collection: S. Das, A.H. Oliaee; analysis and interpretation of results: S. Das, A.H. Oliaee; draft manuscript preparation: S. Das, A.H. Oliaee; M. Le, M.P. Pratt, J. Wu. 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

Subasish Das  <https://orcid.org/0000-0002-1671-2753>
 Amir Hossein Oliaee  <https://orcid.org/0000-0002-0574-2690>
 Minh Le  <https://orcid.org/0000-0003-0129-1615>
 Jason Wu  <https://orcid.org/0000-0001-5438-5565>

References

1. NHTSA. Traffic Safety Facts 2020 Data: Pedestrians. *Traffic Safety Facts*, 2022.
2. NHTSA. Traffic Safety Facts 2019 Data: Pedestrians. *Traffic Safety Facts*, 2021.
3. 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.
4. Federal Highway Administration. Develop Pedestrian and Bicycle Crash Analysis Tool. *Public Roads*, Vol. 63, No. 5, 2000, p. 54.
5. Lopez, D., L. C. Malloy, and K. Arcoletto. 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. <https://doi.org/10.1016/j.aap.2021.106475>.
6. NHTSA. 2020 FARS/CRSS Pedestrian Bicyclist Crash Typing Manual: A Guide for Coders Using the FARS/CRSS Ped/Bike Typing Tool. Report No. DOT HS 813 250. National Highway Traffic Safety Administration, Washington, D.C., 2022, p. 86p.
7. Vavrova, M., C. Chang, S. Kumar, D. Ruiz, S. Gonzalez, and M. D. Benitez. *North Texas Bicycle and Pedestrian Crash Analysis: Final Report*. Report No. FHWA/TX-21/0-6983-1. Texas Department of Transportation, Austin, 2021, p. 213p.
8. 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. <https://doi.org/10.1016/j.jsr.2021.03.005>.
9. 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*, 2016. 2601: 72–83.
10. Chavis, C., Y.-J. Lee, and S. Dadvar. *Analysis of Bicycle and Pedestrian Crash Causes and Interventions*. Report No. DDOT-RDT-2018-01. District Department of Transportation, Washington, D.C., 2018, p. 368p.
11. Banks, G. C., H. M. Woznyj, R. S. Wesslen, and R. L. Ross. A Review of Best Practice Recommendations for Text Analysis in R (and a User-Friendly App). *Journal of Business and Psychology*, Vol. 33, No. 4, 2018, pp. 445–459.
12. Kwayu, K. M., V. Kwigizile, J. Zhang, and J.-S. Oh. Semantic N-Gram Feature Analysis and Machine Learning-Based Classification of Drivers' Hazardous Actions at Signal-Controlled Intersections. *Journal of Computing in Civil Engineering*, Vol. 34, No. 4, 2020, p. 04020015.
13. 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. <https://doi.org/10.1093/tse/tdaa010>.
14. Kwayu, K. M., V. Kwigizile, K. Lee, and J.-S. Oh. Discovering Latent Themes in Traffic Fatal Crash Narratives Using Text Mining Analytics and Network Topology. *Accident Analysis & Prevention*, Vol. 150, 2021, p. 105899.
15. 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 [cs]*, 2019.
16. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, et al. Transformers: State-of-the-Art Natural Language Processing. *Proc., Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, 2020.
17. 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 30, NIPS, 2017.
18. Taylor, W. L. “Cloze Procedure”: A New Tool for Measuring Readability. *Journalism Quarterly*, Vol. 30, No. 4, 1953, pp. 415–433. <https://doi.org/10.1177/107769905303000401>.
19. Kingma, D. P., and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv Preprint arXiv:1412.6980*. <http://arxiv.org/abs/1412.6980>. Accessed June 9, 2022.
20. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. *Distributed Representations of Words and Phrases and Their Compositionality*. Advances in Neural Information Processing Systems 26, NIPS, 2013.
21. Pennington, J., R. Socher, and C. Manning. Glove: Global Vectors for Word Representation. *Proc., Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014.
22. Joulin, A., E. Grave, P. Bojanowski, M. Douze, H. Jegou, and T. Mikolov. FastText.Zip: Compressing Text Classification Models. *arXiv Preprint arXiv:1612.03651*, 2017, p. 13.
23. Wu, Y., M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. <http://arxiv.org/abs/1609.08144>. Accessed June 28, 2022.
24. Hossin, M., and M. N. Sulaiman. A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*, Vol. 5, No. 2, 2015, pp. 1–11. <https://doi.org/10.5121/ijdkp.2015.5201>.
25. Hand, D. J. *A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems*. *Machine Learning* 45, 2001, pp. 171–186.
26. Refaeilzadeh, P., L. Tang, and H. Liu. Cross-Validation. In *Encyclopedia of Database Systems* (L. Liu, and M. T. Özsu, eds.), Springer, New York, NY, 2016, pp. 1–7.