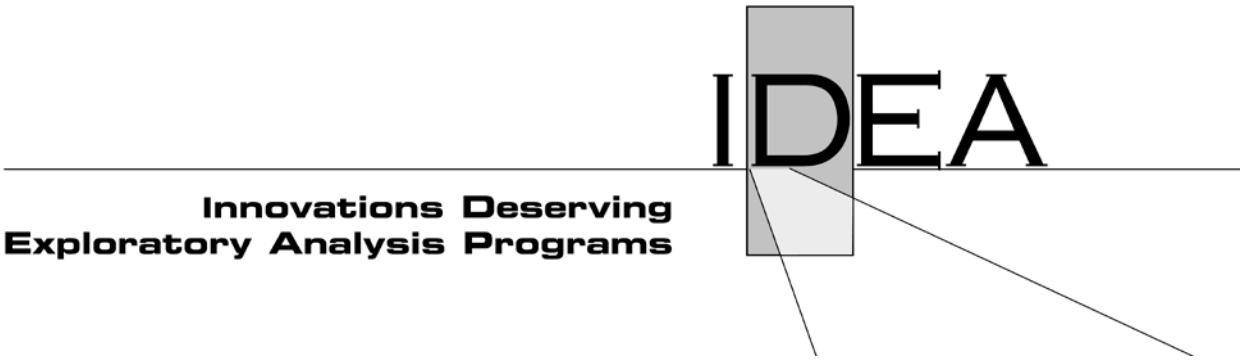


NCHRP IDEA Program

AI Analyzer for Revealing Insights of Traffic Crashes

Final Report for
NCHRP IDEA Project 231

Prepared by:
Jee Woong Park
Cristian Arteaga
University of Nevada at Las Vegas

June 2023

NATIONAL ACADEMIES Sciences Engineering Medicine

TRB TRANSPORTATION RESEARCH BOARD

Innovations Deserving Exploratory Analysis (IDEA) Programs Managed by the Transportation Research Board

This IDEA project was funded by the NCHRP IDEA Program.

The TRB currently manages the following three IDEA programs:

- The NCHRP IDEA Program, which focuses on advances in the design, construction, and maintenance of highway systems, is funded by American Association of State Highway and Transportation Officials (AASHTO) as part of the National Cooperative Highway Research Program (NCHRP).
- The Safety IDEA Program currently focuses on innovative approaches for improving railroad safety or performance. The program is currently funded by the Federal Railroad Administration (FRA). The program was previously jointly funded by the Federal Motor Carrier Safety Administration (FMCSA) and the FRA.
- The Transit IDEA Program, which supports development and testing of innovative concepts and methods for advancing transit practice, is funded by the Federal Transit Administration (FTA) as part of the Transit Cooperative Research Program (TCRP).

Management of the three IDEA programs is coordinated to promote the development and testing of innovative concepts, methods, and technologies.

For information on the IDEA programs, check the IDEA website (www.trb.org/idea). For questions, contact the IDEA programs office by telephone at (202) 334-3310.

IDEA Programs
Transportation Research Board
500 Fifth Street, NW
Washington, DC 20001

AI Analyzer for Revealing Insights of Traffic Crashes

IDEA Program Final Report

NCHRP IDEA 231

Prepared for the IDEA Program

Transportation Research Board

The National Academies

Dr. JeeWoong Park, Principal Investigator

Dr. Cristian Arteaga, Research Associate

University of Nevada Las Vegas

June 30, 2023

ACKOWLEDGEMENTS

The authors would like to express their deep appreciation to the Massachusetts Department of Transportation (MassDOT) for generously providing the invaluable crash data used in this research. We are grateful for the considerable time and effort that Bonnie Polin and the MassDOT team devoted to manually cleaning and removing personal identifiable information from the provided dataset of narratives. We also extend our thanks to Advisory Board members (Rich Cunard, Patricia Leavenworth, Yusuf Mohamedshah) for facilitating the connection with agencies that could provide us with a dataset of crash narratives. We would like to acknowledge the IDEA Panel and Advisory Board members, including the IDEA program manager Inam Jawed, for their valuable feedback and comments throughout the project's execution. Finally, we wish to emphasize that the opinions and findings presented in this document are solely those of the authors and do not necessarily reflect the views of the individuals or organizations that participated directly or indirectly in this project.

TABLE OF CONTENTS

EXECUTIVE SUMMARY

Roadway crashes are a significant public health concern, especially as they are the leading cause of mortality among individuals aged 5 to 29 (*1*). Global estimates indicate that over 1.3 million people lose their lives each year due to road traffic crashes, with vulnerable road users such as pedestrians, cyclists, and motorcyclists accounting for more than 50% of these incidents. These statistics emphasize the urgent need for effective measures to address the incidence of roadway crashes and their associated loss of life and injuries.

Identifying factors that contribute to the severity of crashes can improve our understanding of the problem and help prioritize the implementation of countermeasures. Crash narratives, which provide detailed descriptions of the context and sequence of events leading up to a crash, contain valuable information that can help identify these contributing factors. However, technical limitations have prevented their full utilization. To address this issue, this project developed a tool that uncovers correlations between phrases in crash narratives and severe crashes to identify potential severity contributing factors, as illustrated in Figure 1. The tool requires minimal analyst intervention and is expected to facilitate the use of crash narratives as a valuable information source for data-driven decision making.

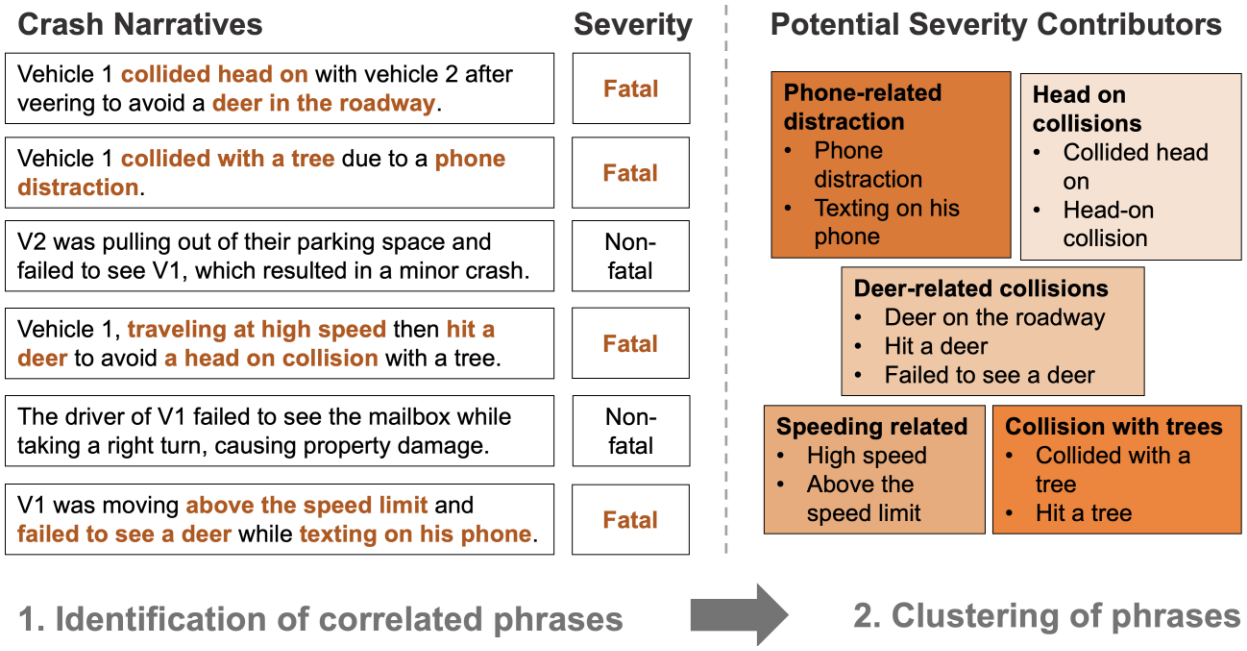


Figure 1. Illustration of the proposed analysis approach

The developed tool utilizes recent advancements in the field of Natural Language Processing (NLP), a subfield of Artificial Intelligence (AI), to analyze crash narratives. The tool takes a raw dataset of crash narratives along with their severity classification as input and outputs a group of clusters of phrases that have been identified as correlated with severe crashes. To achieve this, the tool uses an AI text classifier to establish correlations between the narratives and the crash severity. It then applies an Explainable AI approach to extract correlated phrases and clusters them based on their semantic similarity to synthesize the results and provide an overview of potential severity contributors. The following list summarizes the accomplishments and results of this project:

1. Devised a novel Explainable AI approach to model associations between the inputs and outputs of a text classifier in the form of phrases. The approach addresses the limitations of existing word-level explanation techniques and enables the extraction of explanations at the phrase level. The validation of the approach using an induction-based method showed that it successfully recovers a large proportion of correlations available in a dataset.
2. Identified and evaluated text classification techniques for analyzing crash narratives. An exhaustive investigation of state-of-the-art text classification models was conducted to identify and compare the techniques best suited for establishing correlation patterns between narratives and crash severities. The comparison focused on finding models that

exhibit a convenient balance between predictive performance and computational complexity to make the tool better suited for practical application.

3. Developed a technique to synthesize a large number of correlated phrases. Given the large number of phrases identified as correlated with severe crashes, this project investigated and integrated state-of-the-art techniques for clustering text data to provide a summary or overview of potential severity contributors from the narratives.
4. Implemented a web-based system that integrates the developed analysis approach. This project developed a web-based user interface that enables easy interaction with the approach to identify severity contributors in crash narratives. The web-system can be used as a local web application or as a network server to facilitate use by multiple parties.

IDEA PRODUCT

The IDEA product developed in this project is an open-source software tool that transportation engineers can use to analyze traffic crash data using AI, specifically NLP. NLP has made great strides in understanding human language, as demonstrated by its success in tasks like sentiment analysis, intent detection, and question answering. By leveraging NLP, the developed tool can analyze traffic crash narratives and identify the factors that contribute to the severity of crashes. This provides a promising tool for extracting insights from traffic crash narratives, which allow us to gain a better understanding of traffic crashes and thus design more effective countermeasures.

Traditionally, traffic safety analysis focuses mostly on the severity of traffic crashes using quantitative data available in crash reports, while the rich information in textual descriptions of crashes is largely underutilized due to our limited ability to process large amounts of data; therefore, very little is known about this information. The developed tool offers unprecedented opportunities to extract valuable information from a data source that has been vastly underutilized in the context of traffic safety. By assisting analysts in processing large amounts of crash narratives, the developed tool offers insights that other data types may not reveal, providing further opportunities to identify the crash factors that require urgent attention in the implementation of countermeasures to mitigate future traffic crashes.

CONCEPT AND INNOVATION

Capitalizing on recent technical advances in computer science, particularly AI and NLP, the developed IDEA product provides a novel analytical tool that the current state of practice does not possess. To extract information from crash narratives, the current practice relies on either manual reading of the narratives, which is time-consuming and prone to subjectivity, or in the use of keywords to identify crash factors in the text data, which is inaccurate and error prone. The developed AI-based tool provides a more efficient and effective way to extract important information from these narratives, thereby improving the overall understanding of traffic crashes and minimizing the effort of the analysts on extracting insights from the valuable information contained in narratives.

The goal of the proposed AI-based analyzer is to provide an overview of the phrases in the narratives that have a stronger correlation with more severe crashes. Figure 2 illustrates the processing pipeline used by the developed software tool to achieve this goal. First, the tool takes the raw crash narratives and their severity classification as input. Second, the tool passes those narratives to the AI analysis approach, which provides as output as the clusters of phrases correlated with severe crashes, which reveal potential severity contributing factors.

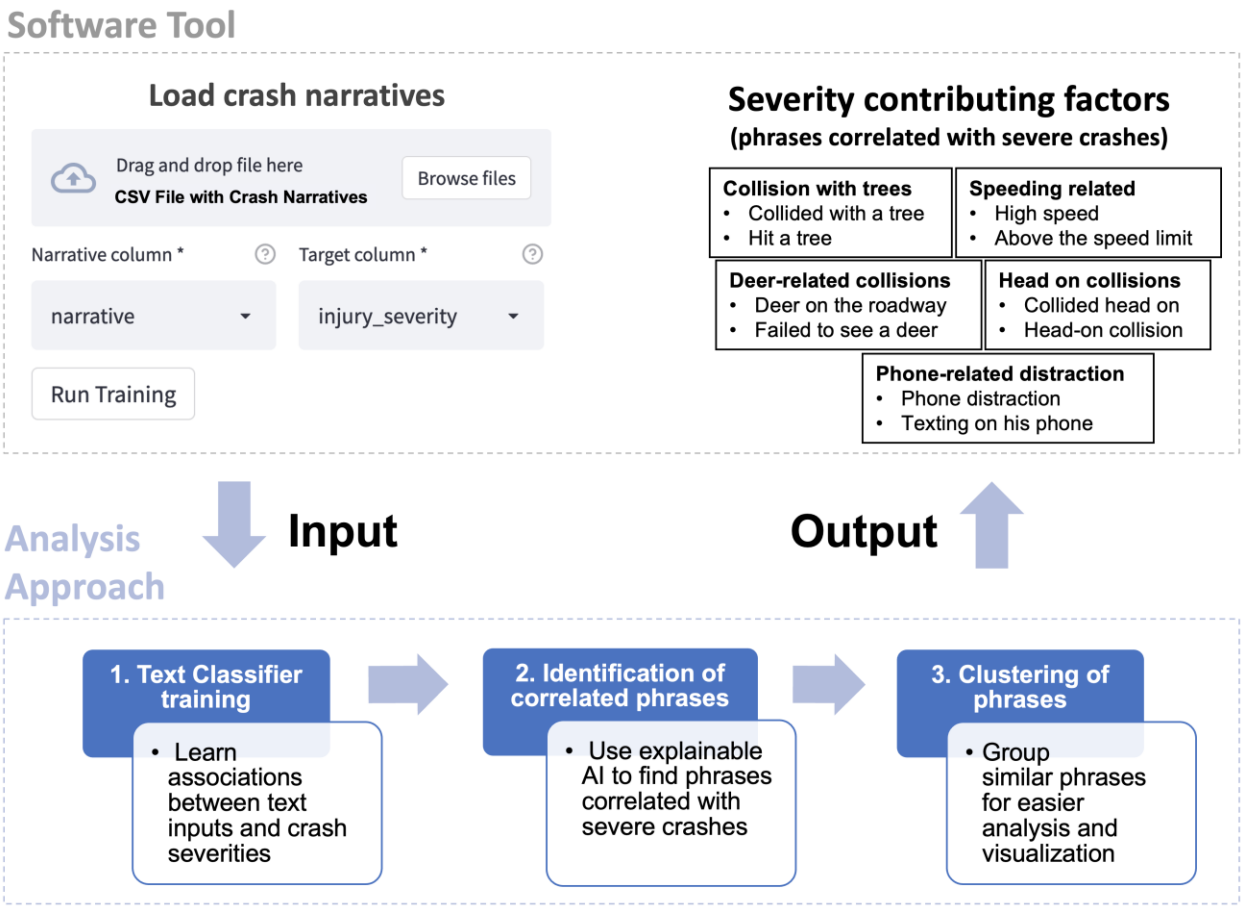


Figure 2. Overview of the proposed software tool and underlying analysis approach

To provide the correlated phrases, the AI analysis approach follows three internal processing steps, as shown in Figure 2 and detailed below.

The first step involves the training of an AI model to determine the relationship between the phrases in crash narratives and the severity of the crashes. To achieve this goal, the approach makes use of text classifiers, which can predict the categories of text paragraphs by learning the associated patterns in the text inputs. The method identifies phrases related to a specific crash severity output using the learned patterns. The text classifier is fed with the crash narrative, and it classifies it based on its internal mechanism, predicting the severity of the crash. At the end of the training, the text classifier will have learned the patterns in the input narratives associated with an output severity level.

The second step uses the text classifier trained in the previous step and an Explainable AI approach to identify phrases correlated with severe crashes. The devised Explainable AI approach combines

a sliding window and a peak detection technique to determine the phrases associated with a specific classification target. Initially, the sliding window technique segments the text input into windows of a predetermined size and then individually passes them through the text classifier to determine their likelihood of being connected to a particular crash severity. Once the output probabilities are obtained for all the windows in the text input, the method converts them to individual word probabilities, which are also known as "word scores."

The third step synthesizes the results of the sliding window and peak detection approaches. Large crash databases typically contain thousands of narratives, which can generate a large number of correlated phrases when analyzed using the sliding window and peak detection approaches. While these individual phrases may offer useful information about potential severity factors, analyzing them one by one can be time-consuming and tedious. To simplify the visualization and analysis of results, the approach summarizes the numerous phrase outputs and presents them in a more accessible format. To achieve this, the approach groups together similar phrases identified as being correlated with the classification output. These phrase clusters offer an overview of the various themes in the narratives associated with severe crashes.

INVESTIGATION

In order to develop an effective AI-based approach for identifying correlations between phrases in traffic crash narratives and the severity of traffic crashes, a comprehensive investigation was conducted. This investigation included a literature review to identify suitable AI models (text classifiers) and Explainable AI approaches for the given task. Additionally, various text classification models were compared in terms of predictive performance and computational complexity, an Explainable AI approach was devised, and the results were validated using statistical analysis and an induction-based method. In this section, we will delve deeper into the investigation process and its findings, which form the basis for the development of the proposed AI-based approach.

Literature Review

The research team conducted a literature review for AI techniques suitable for the classification of crash narratives. The review focused mostly on techniques for modeling of linguistic information based on Neural Networks (NN), given that these techniques have provided the most significant

developments in the NLP literature, as discussed in this section. Also, the investigation included the review of literature in Explainable AI, to identify a technique that can help model the associations of the inputs and outputs of the NN models, as well as a review of techniques to cluster text data.

Text classification models

The application of NNs to model text data started in the early 2000s when the field of Natural Language Processing (NLP) began to see advances with "Neural Language Models." A model proposed by Bengio et al. in 2003 (*2*) used neural networks to learn the joint probability function of a sequence of words, resulting in significant improvements in state-of-the-art models based on word n-grams. Collobert and Weston in 2008 (*3*) used multitask learning to train neural networks for multiple NLP tasks, leading to enhanced generalization of language modeling and improved performance in shared tasks. Another crucial development in 2013 was the introduction of Word2vec by Mikolov et al. (*4*), which provided multidimensional vector representations for words that captured important language properties and could be reused across various NLP tasks. Sustkever (*5*) in the same year proposed a method for training Deep Learning architectures with powerful modeling of sequential data, such as Recurrent Neural Networks and Long-Short Term Memory, which made it possible to use these architectures more widely in different NLP tasks.

Kim et al. (*6*) demonstrated in 2014 that Convolutional Neural Networks (CNN) could greatly enhance the classification performance of text sentences, thanks to their ability to fully exploit parallel processing, which is less effective in Recurrent Neural Networks. In the same year, Sutskever et al. (*7*) made a significant breakthrough in machine translation by introducing a deep learning architecture based on an encoder-decoder strategy. This model could translate a sequence of words from one language to another and considerably improve the state-of-the-art results in neural machine translation. The improvement in machine translation using neural networks was so remarkable that Google integrated this approach into their Google Translate product in 2016 (*8*).

Bahdanau et al. (*9*) proposed in 2015 the "attention mechanism" which is one of the most significant advancements in the NLP domain. The attention mechanism enables NLP models to concentrate on the essential parts of text when processing sentences, resulting in a considerable improvement in the state-of-the-art in neural machine translation. This innovation serves as the foundation of the Transformer architecture (*10*), a novel neural network structure that parallelizes

sequential data processing efficiently. The Transformer architecture has revolutionized both research and practice in NLP, as well as in computer vision and speech recognition (*11–13*). Transformers also use positional encodings (*14*) to address the issue of time-consuming sequential processing of inputs in recurrent neural networks. For NLP, the Transformer architecture is a groundbreaking development, serving as the foundation for several models, ranging from small pre-trained language models (*15–18*) to large language models with more than 100 billion parameters (*19–21*).

In the year 2018, two pre-trained language models, Bidirectional Encoder Representations from Transformers (BERT) (*15*) and Generative Pre-Training (GPT) (*22*), were introduced. These models use the Transformer architecture as their base and have significantly improved the performance of various NLP tasks. BERT and GPT have been used in question-answering models, where they have outperformed humans in challenging tasks like reading comprehension (*23*). These advancements in NLP have brought it closer to human-level performance in complex language modeling tasks, indicating its potential to extract valuable insights from crash narratives. BERT has advanced various NLP applications, but suffers from a potential downside that is its high computational demand due to its quadratic attention mechanism, which poses limitations for training and inference on less sophisticated computational resources. To overcome this issue, researchers have proposed approaches such as DistilBERT (*17*), which applies distillation strategies to transfer knowledge from the original BERT model into a reduced and computationally optimized version. By reducing the size of the original BERT model, DistilBERT is about 60% faster while maintaining the language modeling capabilities above 95%. Because of this balance between high predictive performance and reduced computational demands, this project also uses DistilBERT for the text classification task. Also, given that CNN are convenient in terms of computational complexity while maintaining a consistent level of predictive accuracy, this project will compare the speed and predictive performance of the CNN, BERT, and DistilBERT architectures in order to find out which offers the most convenient trade-off between predictive performance and computational complexity for classification of text narratives.

Explainable AI techniques

For explainable AI techniques, the literature search revealed two potential models that could help uncover the associations between the inputs (narratives) and outputs (crash severities) of the text

classifier. The first model is Local Interpretable Model-Agnostic Explanations (*24*) which is a technique that generates explanations for machine learning models' predictions. It achieves this by approximating the model's predictions using a simpler model that is easier to understand. This simpler model is trained on local data surrounding the specific instance being explained. The weights of the features in this simpler model are then used to produce an explanation. This technique is not tied to any particular machine learning model and is capable of providing instance-level explanations, allowing users to see how the model came up with a specific prediction for a given input. The second method is Shapley Additive Explanations (*25*), which is a system for explaining any machine learning model's output. It is based on the cooperative game theory's concept of Shapley values, which assigns importance values to each feature in a prediction. The Shapley Additive Explanations allow the prediction to be decomposed into the contribution of each feature, and they can explain both global model and instance-level predictions. Additionally, Shapley Additive Explanations can be applied to any model, including deep learning models.

Despite the value of the aforementioned models in the Explainable AI domain, these models suffer from three main limitations. First, for text applications, these techniques provide explanations at the individual-word level, which hinders the extraction of insights at the phrase level, as needed in the developed approach. In addition, these techniques have the disadvantage of being very computationally demanding, as thousands of random perturbations need to be generated in order to produce explanations. Finally, these techniques require a careful definition of hyperparameters in order for them to work properly. Therefore, the research team decided to develop a simplified and less computationally intensive explainability technique by borrowing concepts from the Local Interpretable Model-Agnostic Explanations approach. Despite its simplicity, the proposed technique produces meaningful explanations by applying an intuitive sliding-window approach. The “Explainable AI Approach” section details the devised technique and summarizes the experiments and sensitivity analysis conducted to evaluate it.

Text clustering techniques

To cluster text data, a numerical representation of the text phrases is necessary. Traditional methods for this involve using bag of words and term frequencies, which utilize word counting to create a vector representation of phrases. These methods have been valuable for text mining applications, but they have limitations in capturing the full meaning of language. Modern techniques for

numerically representing text use word embeddings, which provides a fixed length vector to represent each word in a pre-defined vocabulary, as exemplified by word2vec (*26*) and GloVe (*27*). These techniques are potentially useful for clustering tasks; however, they suffer from limitations in representing out-of-vocabulary words, particularly those that are domain-specific or less common. Another drawback is that word2vec and GloVe only provide individual word vectors, and to obtain a phrase-level representation, the individual word vectors must be averaged, resulting in a suboptimal phrase vector representation due to the loss of information caused by the averaging process.

Newer phrase-level representation techniques have been developed using Transformer-based architectures such as BERT and DistilBERT to address the limitations of traditional word embedding methods. One such technique is Sentence BERT (SBERT) (*28*), which utilizes Siamese BERT architectures to train phrase-level vector representations that accurately evaluate contextual similarity between phrases. This approach has demonstrated better performance than previous phrase representation techniques due to its pre-trained approach specifically aimed at phrase-level similarity. To minimize out-of-vocabulary issues, these transformer-based representations use tokenization schemes based on sub-word modeling, like byte pair encoding, to represent words using smaller pieces that can capture out-of-vocabulary words. Additionally, SBERT produces vector representations at the phrase level instead of the word level, minimizing information loss caused by averaging embeddings at the word level. As a result, the developed analysis approach clusters and synthesizes analysis results using SBERT's vector representations. Given the advantages of the SBERT vector representations, we integrated this approach for the clustering tasks.

Evaluation of Text Classification Models

In order to assess the ability of the classifiers (BERT, DistilBERT, and CNN) to perform well on data from different domains, two different datasets were used for training and evaluation of the classifiers. The first is the IMDB dataset, a repository of 25,000 movie reviews classified as positive or negative. We used a subset of this dataset with 4,000 records to demonstrate that the proposed AI analyzer provides meaningful results in terms of phrases associated with positive or negative reviews, which at the same time validates the ability of the approach to find correlations between text data and a classification output. The second is a dataset of crash narratives provided

by the Massachusetts Department of Transportation (MassDOT). This dataset contains 1,167 crash narratives as well as additional information about the crashes, which include the crash severity and several other crash factors. From here on, we refer to this dataset as the MassDOT dataset.

For the experiments, we used the Huggingface Transformers Python library (*29*) along with the pre-trained "bert-medium" and "distilbert-base-uncased" models. The CNN model was trained from scratch. The experiments were executed using an NVIDIA Tesla T4 GPU with 16 GB of memory. Table 1 summarizes the comparison of predictive performance and computational complexity of the tested models. In terms of predictive performance, the BERT and DistilBERT models offered the highest accuracies. However, the accuracy of the CNN is also very high with the additional advantage of significantly lower training time. Note that, despite using a powerful GPU, the training time for the BERT and DistilBERT models is very high when compared against the CNN model. In addition, the required GPU memory to train the BERT and DistilBERT models is significantly higher than the one required to train the CNN algorithm. Therefore, due to its convenient equilibrium between predictive performance and computational complexity, the CNN algorithm was selected to be included in the developed AI analyzer.

Table 1. Summary of training time and accuracy for the tested classifiers

	IMDB			**MassDOT**		
	Accuracy	**Training Time (min)**	**GPU Memory Usage (GB)**	**Accuracy**	**Training Time (min)**	**GPU Memory Usage (GB)**
BERT	91.10%	38.1	11.1	88.30%	12.2	9.4
DistilBERT	91.30%	42.2	12.6	88.90%	18.6	11.8
CNN	89.40%	0.85	1.4	87.90%	0.3	1.3

Figure 3 shows the learning curves of the CNN model for all the evaluated datasets. These learning curves suggest that the model successfully lowers the loss metric throughout the different training epochs while increasing the predictive performance (accuracy and F1 scores). However, there is a general trend towards overfitting, as the loss for the training and evaluation subsets decrease at a different rate after a certain epoch. Therefore, users of the AI analyzer must select the model with less overfitting, which is the model at the epoch where the training and evaluation loss curves start

to diverge. We will provide instructions for this model selection in the user manual of the software. As specified in the installation instructions of the developed tool, users can benefit from using GPUs to significantly speed-up the training of the text classifiers. Based on the results on Table 1, the CNN classifier only uses 1.4 GB of GPU memory, which implies that even GPUs with low memory availability can be leveraged to improve estimation times.

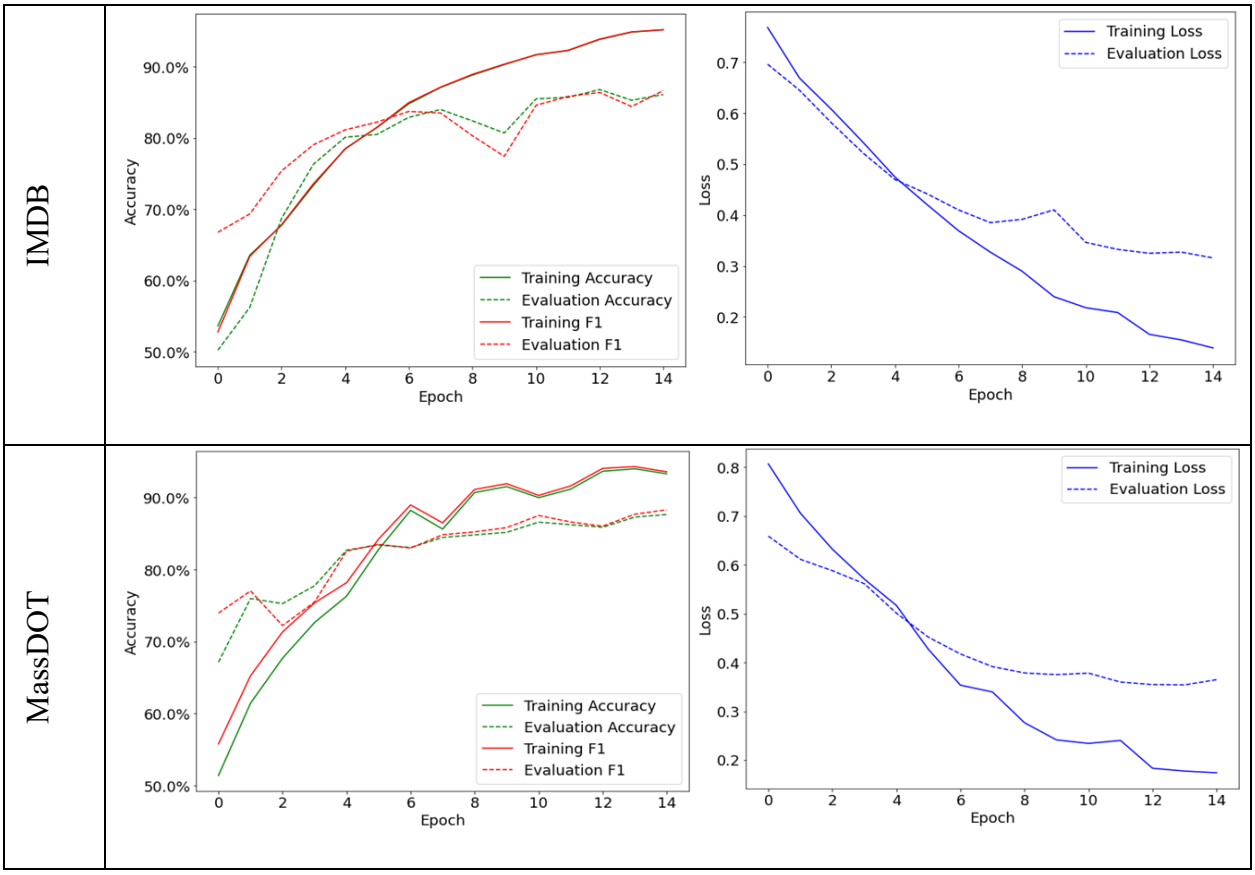


Figure 3. Training curves for CNN model

After model training, the next step corresponds to the development of the explainable AI approach, as described in the next subsection.

Explainable AI Approach

The proposed analysis approach uses a text classifier to learn the patterns in the input narratives associated with a certain severity level. Generally speaking, text classifiers work as black boxes that provide a classification output without providing insights about the relationship between the input text and the classification output. The literature search for Explainable AI methods revealed

that techniques such as Local Interpretable Model-Agnostic Explanations and Shapley Additive Explanations provide explanations for any type of machine learning models. However, these methods are limited to word-level explanations that may convey incomplete or ambiguous ideas. In addition, these methods are computationally intensive because of the stochastic sampling required to generate explanations. To overcome these limitations, we developed an Explainable AI approach that provides explanations in the form of phrases, instead of individual words, which in turn enables a better capturing of language semantics.

Figure 4 illustrates the proposed Explainable AI approach. We use a sliding window of fixed size to evaluate how different parts of the text input affect the output probability of the text classifier. In other words, the text input is split into phrases of fixed size (windows) that are individually fed into the text classifier to measure their probability of association with a certain crash severity. We call this probability a "score". The window scores are translated into individual word scores by taking an average across the different sliding windows that overlap for the same word.

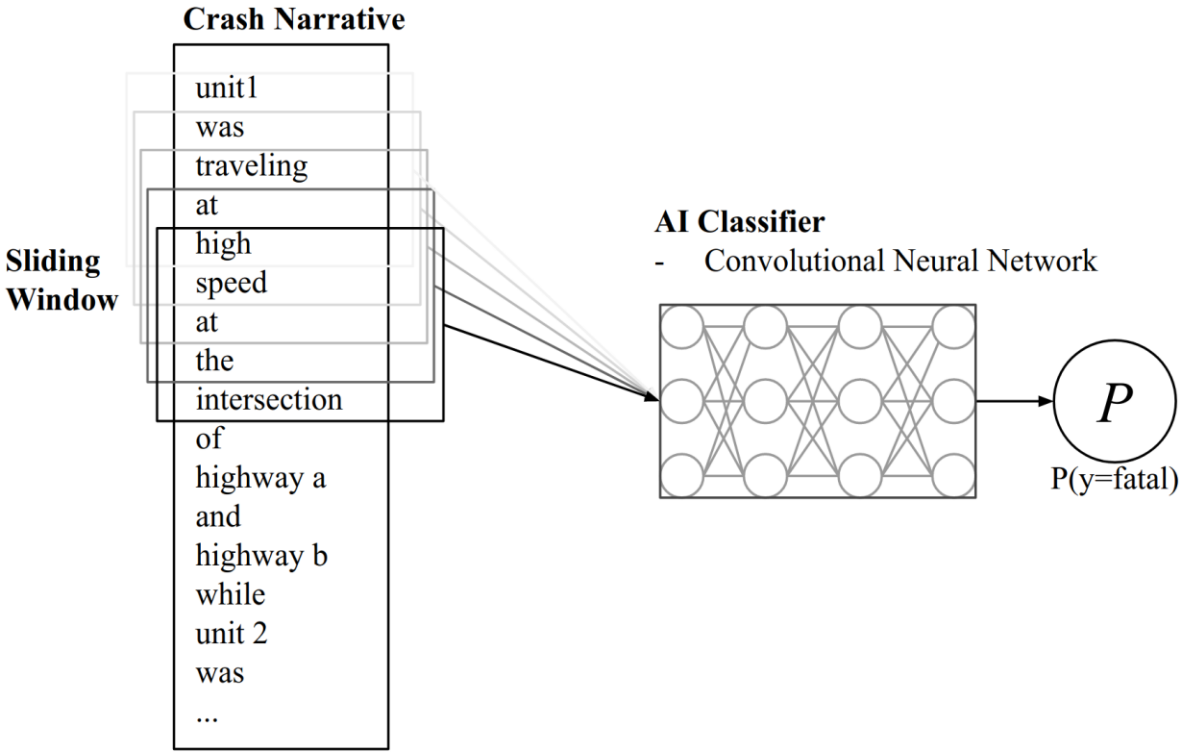


Figure 4. Proposed explainable AI-approach

The individual word scores can be interpreted as a measure of importance or contribution of a word to the output probability. Figure 5 illustrates two sample narratives and their associated plots

of individual word scores. The sliding-window approach assigns larger scores to words that have a stronger correlation with fatal crashes, which are reflected in the word importance plots as peaks scores (higher probability of the crash being fatal). For instance, in the presented examples, the peaks in scores are associated with phrases such as “collided head on” and “a prime mover and a trailer”.

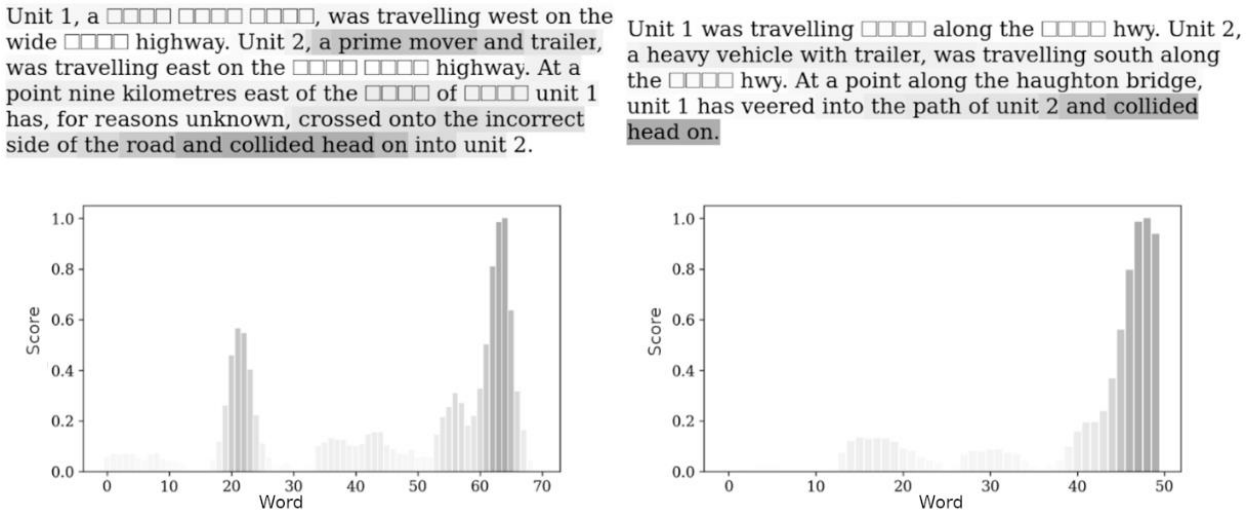


Figure 5. Illustration of individual word scores

After computing the individual word scores, the developed approach uses a peak detection approach to determine the sequence of words that exhibit the highest correlation with the severity of the crash, as illustrated in Figure 6. To extract the sequence of words (phrases), the peak detection technique uses a filtering threshold based on *z*-scores, which is a technique that has been previously used in transportation studies (*30*, *31*). The filtering threshold δ is defined as a certain number of standard deviations above the mean of the word scores. The approach determines suitable values for the δ parameter through a sensitivity analysis, as explained in the next section.

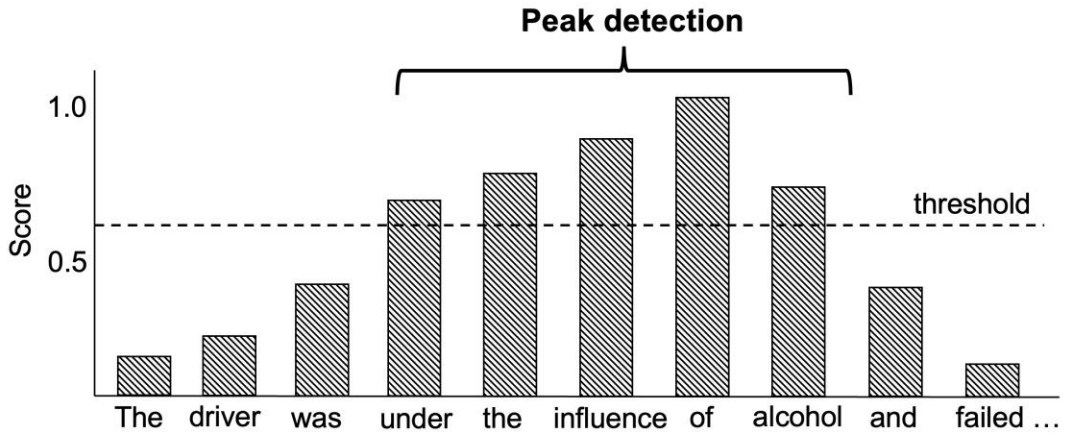


Figure 6. Illustration of the mechanism that detect peaks in word scores

Sensitivity Analysis of the Parameters of the Approach

As described in the previous section, the sliding window method provides a quantitative metric of word importance. However, the size of the sliding window may affect the spatial distribution of the individual word scores. Therefore, we conducted a sensitivity analysis to determine the extent at which the size of the sliding window, which here we refer to as "window size", affects the word scores. Figure 7 shows the spatial distribution of individual word scores for different window sizes (ws), where the horizontal axis represents the word position and the vertical axis the word score. These scores correspond to randomly selected records from 1) a dataset of narratives provided by the Massachusetts Department of Transportation (MassDOT), and 2) movie reviews from the Internet Movie Database (IMDB) for validation; we used the IMDB for more robust validation for general applicability, especially for better handling variance in language. After substantial analysis, larger window sizes (bigger than 5) generate a spatial distribution with more identifiable peaks while potential challenges with flat spatial distributions when the window size is excessively increased. With further analysis, we found that window sizes between six and ten words provide convenient spatial distributions from which peaks in word scores can be identified. This is an intuitive result, as English phrases that contain few words (ws < 5) usually cannot express complete ideas. At the same time, when phrases are too long, there is a high chance multiple ideas are being expressed.

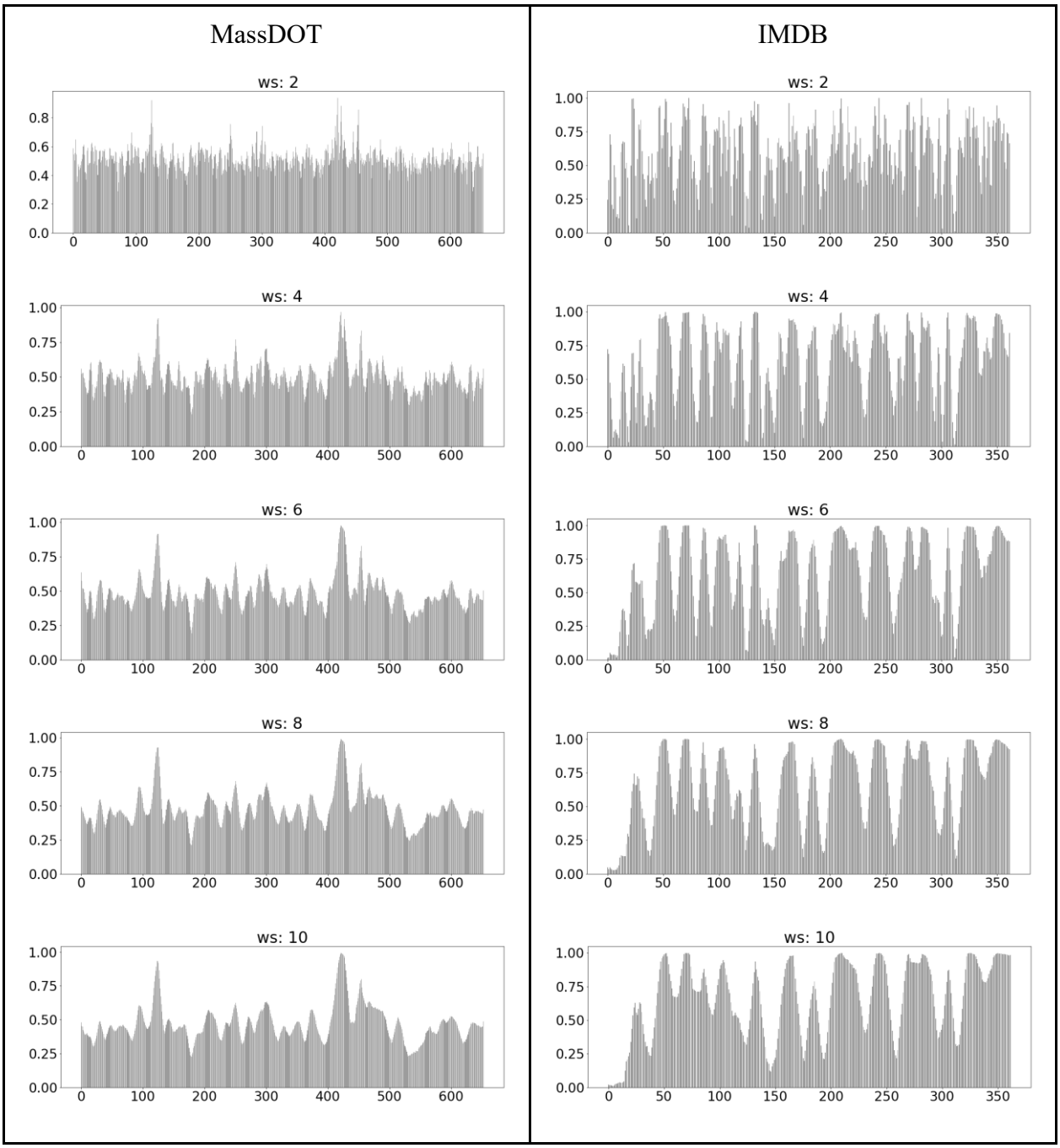


Figure 7. Effect of window size (ws) on the spatial distribution of individual word scores.

After computing the individual word scores, it is necessary to identify the sequence of words (phrases) where the peaks in scores occur. This can be achieved by considering peaks as the set of scores that significantly deviate from the mean scores. Therefore, we identify the peaks by filtering keeping the scores above a certain number of standard deviations δ from the mean. Figure 8 shows the effect of the number of standard deviations used to filter out scores on the length of the phrases

returned by the peak-detection mechanism. As expected, a high value of standard deviations used as filtering threshold results in fewer and shorter phrases that have a high correlation with the classification output. At the same time, small standard deviations used as filtering threshold results in numerous and longer phrases, given that phrases with lower correlation are also included in the results. Therefore, the developed approach will provide a mechanism to adjust this standard deviation parameter, so analysts can set a desired level of correlation for the extracted phrases and at the same time control the number and length of the phrases returned by the analyzer.

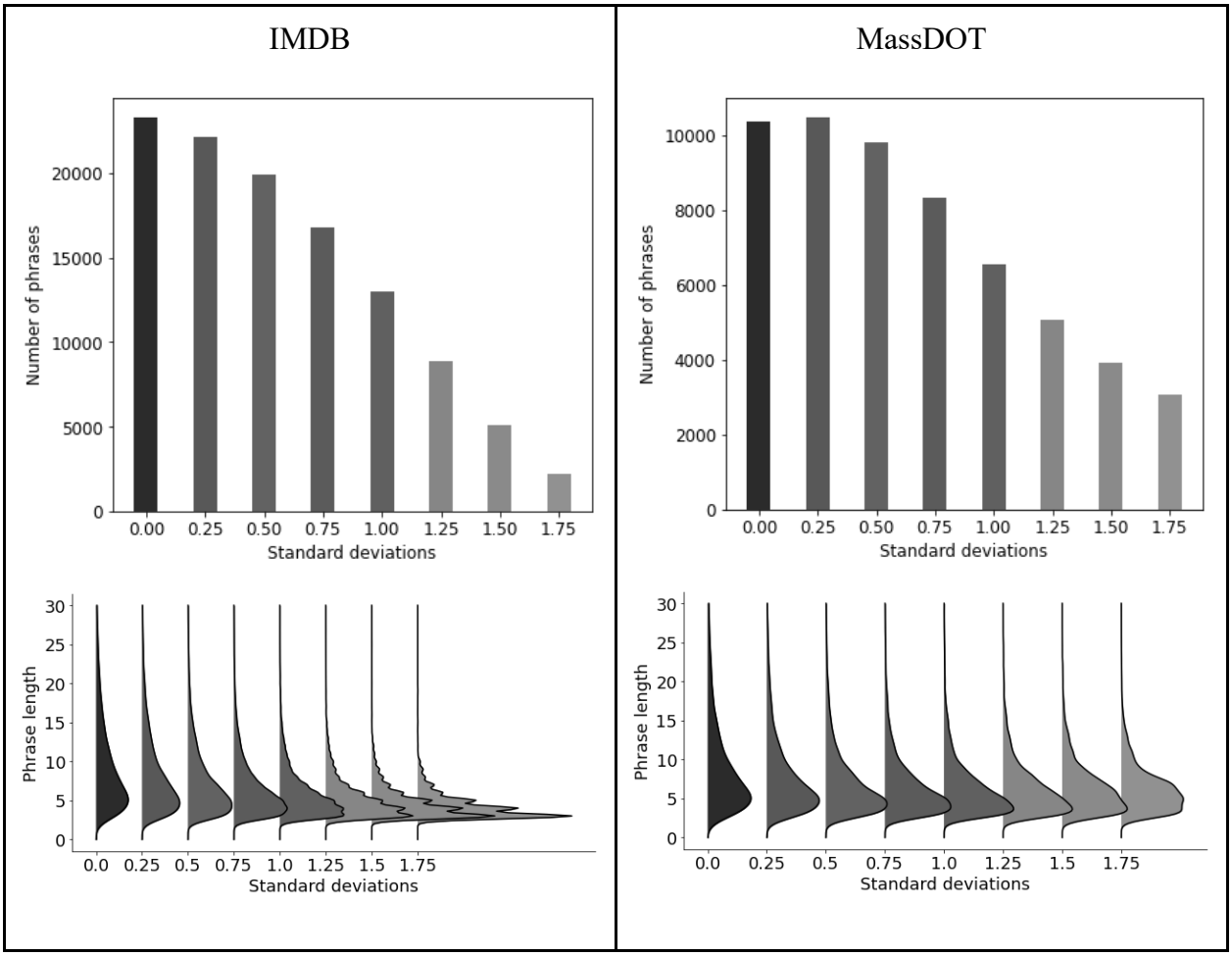


Figure 8. Effect of the number of standard deviations used as threshold for peak detection on the number and length of the returned phrases.

Results and Validation

Figure 9 illustrates the results returned by the proposed system on the MassDOT dataset, which correspond to clusters of phrases correlated with severe crashes. These phrases describe elements associated with severe crashes, such as ambulances transporting people to hospitals, officers arriving at the scene, guardrails struck, pedestrians struck while on a crosswalk, head-on crashes with poles, speeding, alcohol involvement and conditions related to rainy weather. We adopted three validation strategies to evaluate whether the proposed system is properly identifying correlations between the text narratives and severe crashes. First, we compared the results of the proposed system against the results of classic statistical analysis. Second, we used an induction-based method to validate that the proposed system identifies phrases intentionally induced in the narratives. Third, we used a database of movie reviews, for which we have a general idea of existing correlations and evaluated the ability of the system to identify such correlations.

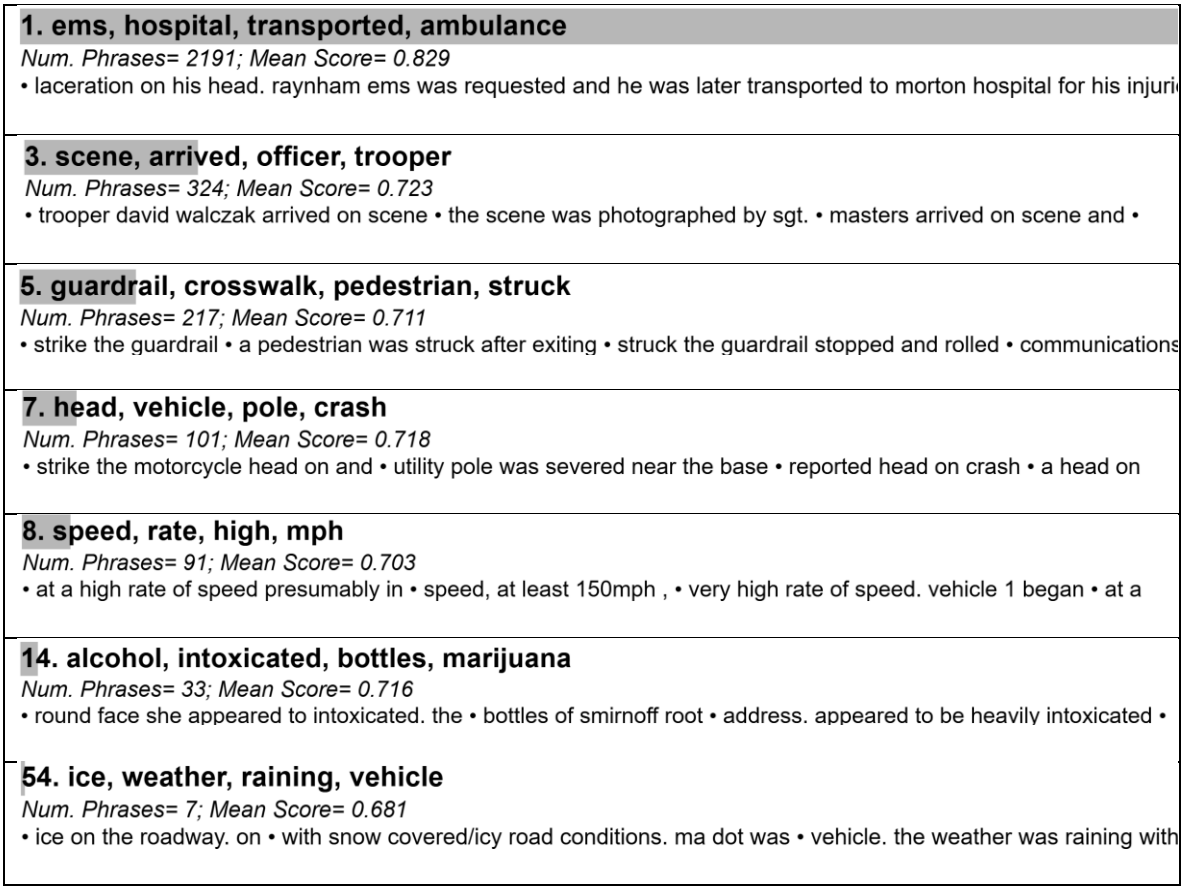


Figure 9. Examples of phrases returned by the proposed approach for the MassDOT dataset

Validation by comparison against statistical analysis

Along with the crash narratives, the MassDOT provided us with quantitative crash data. We performed a standard regression-based analysis (Logistic Regression) to identify the significance of the crash factors for the severity of the crash and compare them with those identified by the proposed system. To estimate the parameters of the Logistic Regression model, this study used an open-source library for estimation of logit-based models called *xlogit* (*32*), which was developed as part of the execution of this project. Table 2 shows the results of the Logistic Regression analysis. The positive sign in the coefficients indicates that the factor influences severe crashes whereas a negative sign indicates it influences non-severe crashes. The stars represent the statistical significance of the coefficients. The Logistic Regression results indicate that crash factors such as the speed limit, the involvement of pedestrians, the head-on nature, and the lack of road lighting can be considered statistically significant factors that influenced high-severity crashes. The comparison of the results of this statistical analysis in Table 2 with those returned by the proposed system in Figure 9 reveals a high level of similarity of results. For instance, both approaches identified that the involvement of pedestrians, speeding, and head-on nature are potential severity contributors.

The proposed system can provide a superior understanding of the factors affecting the severity of the crashes by leveraging additional information in the narratives that might not be present or might be miscoded in the quantitative crash data. Our analysis identifies several discrepancies in results. For example, the quantitative Logistic Regression analysis yields a negative sign for the "Alcohol-related" variable, which indicates that this factor is associated with less severe crashes. This is a counter-intuitive result that is possibly incorrect because the use of alcohol or drug tends to negatively impact crashes. A possible reason for this inconsistency is the potential errors in the coding for alcohol and drug-related variables in the quantitative data, as acknowledged by the data providers. However, the analysis completed by our developed method with narratives suggests that crashes involving alcohol or drug use are highly severe, which we claim as our system's added capability to potentially identify incorrectly coded entries from quantitative data. In sum, these results validate the ability of the proposed system to identify crash severity contributors.

Table 2. Logistic regression results for severity analysis on MassDOT data

Coefficient	Estimate	Std.Err.	z-val	P>|z|	
AADT	0.0000	0.0000	-0.0145	0.798	
Speed Limit	0.0151	0.0042	3.5482	0.00152	**
Hit Pedestrian	3.149	0.7018	4.4868	3.67E-05	***
Hit a Tree	0.6413	0.4029	1.5915	0.225	
Hit a Pole	0.4468	0.3398	1.3148	0.336	
At Intersection	-0.2284	0.2052	-1.1128	0.429	
At Work zone	-1.05	0.4751	-2.2098	0.0696	.
Alcohol-Related	-0.978	0.1631	-5.9932	1.65E-08	***
Drug-Related	0.12	0.2173	0.5523	0.685	
Nature-Head On	1.4522	0.4025	3.6073	0.00123	**
Nature-Rear End	-0.1127	0.2368	-0.4759	0.712	
Nature-Side Swipe	-0.9132	0.2549	-3.5816	0.00135	**
Nature-Single Vehicle	-0.2957	0.2405	-1.2292	0.375	
Nature-Unknown	-0.341	0.6409	-0.5321	0.692	
Weather-Cloudy	-0.2752	0.2068	-1.3305	0.329	
Weather-Rain	-0.937	0.3982	-2.3528	0.0503	.
Weather-Snow	-0.3918	0.4797	-0.8166	0.571	
Weather-Unknown	-0.3223	0.3015	-1.0691	0.45	
Dark-No Road Lighting	0.5596	0.2015	2.7767	0.0171	*
Dark-Road Lighting	-0.0083	0.1787	-0.0468	0.797	
Dawn	-0.1113	0.4428	-0.2514	0.773	
Dusk	0.2324	0.4569	0.5087	0.701	
Unknown Lighting	0.2668	0.6362	0.4194	0.73	
Road Condition-Ice	-0.1774	0.6031	-0.2941	0.764	
Road Condition-Snow	0.1644	0.503	0.3268	0.756	
Road Condition-Unknown	0.4955	0.4903	1.0106	0.479	
Road Condition-Wet	0.5406	0.3329	1.6237	0.214	
Distraction-Electr. Device	0.1537	0.402	0.3823	0.741	
Distraction-Other	0.3238	0.2865	1.1302	0.421	
Age 18-20	0.5233	0.3583	1.4605	0.275	
Age 21-24	0.1269	0.3253	0.39	0.739	
Age 25-43	0.3438	0.3047	1.1285	0.422	
Age 35-44	0.2794	0.3224	0.8666	0.548	
Age 45-54	0.1735	0.3741	0.464	0.716	
Age > 54	-0.1787	0.4224	-0.4231	0.729	
Log-Likelihood= -686.4; Significance: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0					

Validation using an induction-based method

To validate that the proposed approach properly identifies the correlations between phrases and a given target, we intentionally induced correlations between some pre-designed phrases and the analysis target. Follow-up analysis with both the MassDOT and the IMDB datasets then evaluated

the proposed approach with respect to its ability to identify such correlations. We randomly selected a subset of *N* narratives, 90% from the positive class and 10% from the negative class, in which we incorporated the designed phrases. The 10% of phrases in the negative class account for potential real-life conditions, in which perfect data separation is uncommon. Table 3 presents the results of this evaluation. To better isolate the induced correlation, we designed the phrases such that their content described topics differently to those in the narratives. Without this strategy, it would be difficult to evaluate whether the correlation was induced, or it was already present in the data. In addition, we paraphrased the designed phrases in four alternative ways to make the identification more challenging for the proposed system, as it needs to identify phrases written in potentially different ways. The results in Table 3 indicate that the approach successfully identifies the induced correlation by extracting a large proportion of the incorporated phrases. Also, when the phrases are more recurrent in the narratives, the proposed system does a better job at recovering them.

Table 3. Results of induction-based validation on the MassDOT and IMDB datasets

	Induced phrases	**Induced**	**Recovered**
MassDOT	- The future belongs to those who believe in the beauty of their dreams	45	34 (75.6%)
	- Believing in beautiful dreams makes you owner of the future	90	73 (81.1%)
	- Those who have beautiful dreams and believe in them possess the future - If you believe in your beautiful dreams, the future belongs to you	180	173 (96.1%)
IMDB	- Generalization is the ability of a model to perform accurately on unseen examples	45	36 (80.0%)
	- Models that perform accurately on unseen examples have generalization ability	90	78 (86.5%)
	- Performing accurately on unseen examples is called generalization ability - The ability to perform accurately in unseen examples is known as generalization	180	176 (97.7%)

Validation using a dataset with known correlations

The IMDB dataset contains movie reviews and their classification as positive or negative. Using this dataset, we evaluated whether the proposed system identifies meaningful descriptions of positive movie reviews. Figure 2 shows selected examples of phrases identified by the proposed approach as correlated with positive reviews. These phrases describe topics such as great movies,

good music/performances, worth-watching materials, and murder and world war movies, all of which are meaningful descriptions of positive movie reviews. This meaningful identification of phrases correlated with positive movie reviews supports to some degree the ability of the proposed system to identify correlations between text data and a given analysis target.

Figure 10. Examples of phrases returned by the proposed approach for the IMDB dataset

Software Tool

The developed software tool has two main components. The front-end component, which is a web interface to interact with the AI approach, and the backend component, which is the programming logic that implements the developed AI approach.

The front-end component was developed using the Streamlit open-source software, which enables a seamless development of web interfaces for applications that require minimum interaction from the user. As illustrated in Figure 11, web-based user interface provides a functionality to upload the text data in CSV format. After uploading the data, the user selects the column in the CSV file that contains the text narratives as well as the output of the analysis. Finally, before beginning the training of the text classifier, the user can change the number of training epochs, if needed, or

select a column that contains Unique IDs in order to link the identified phrases back to the narratives when displaying the results.

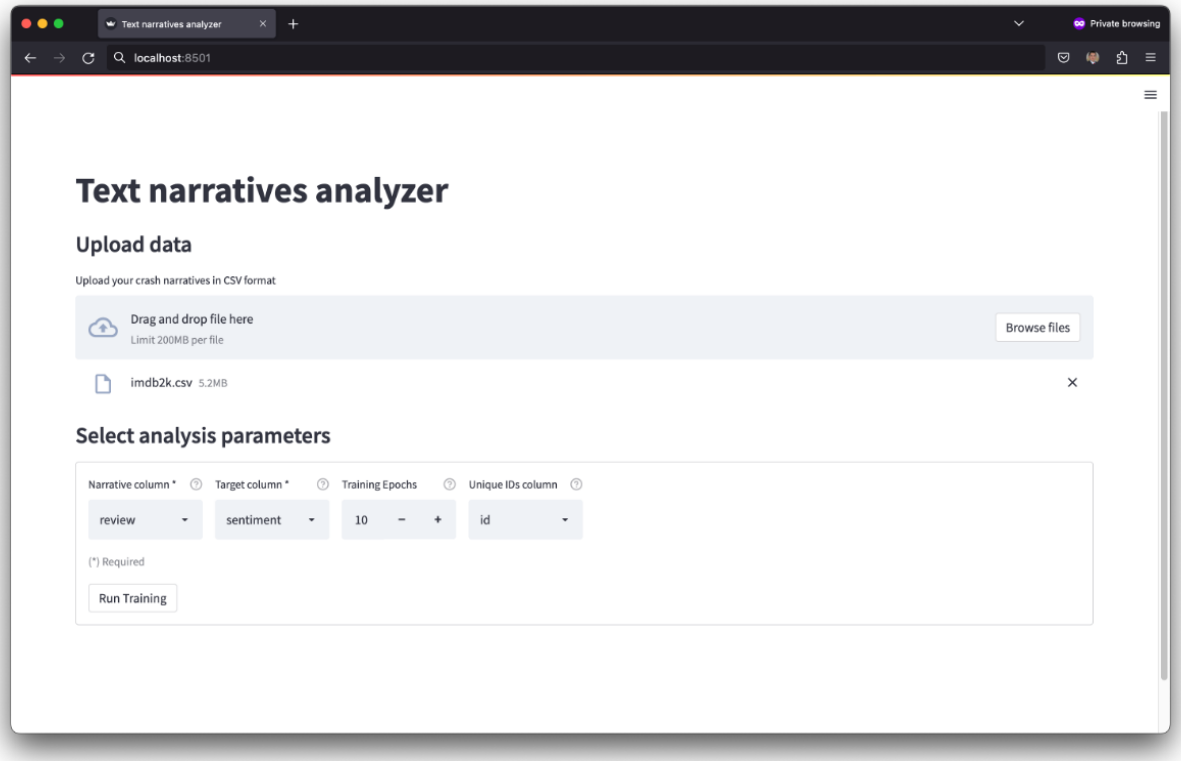
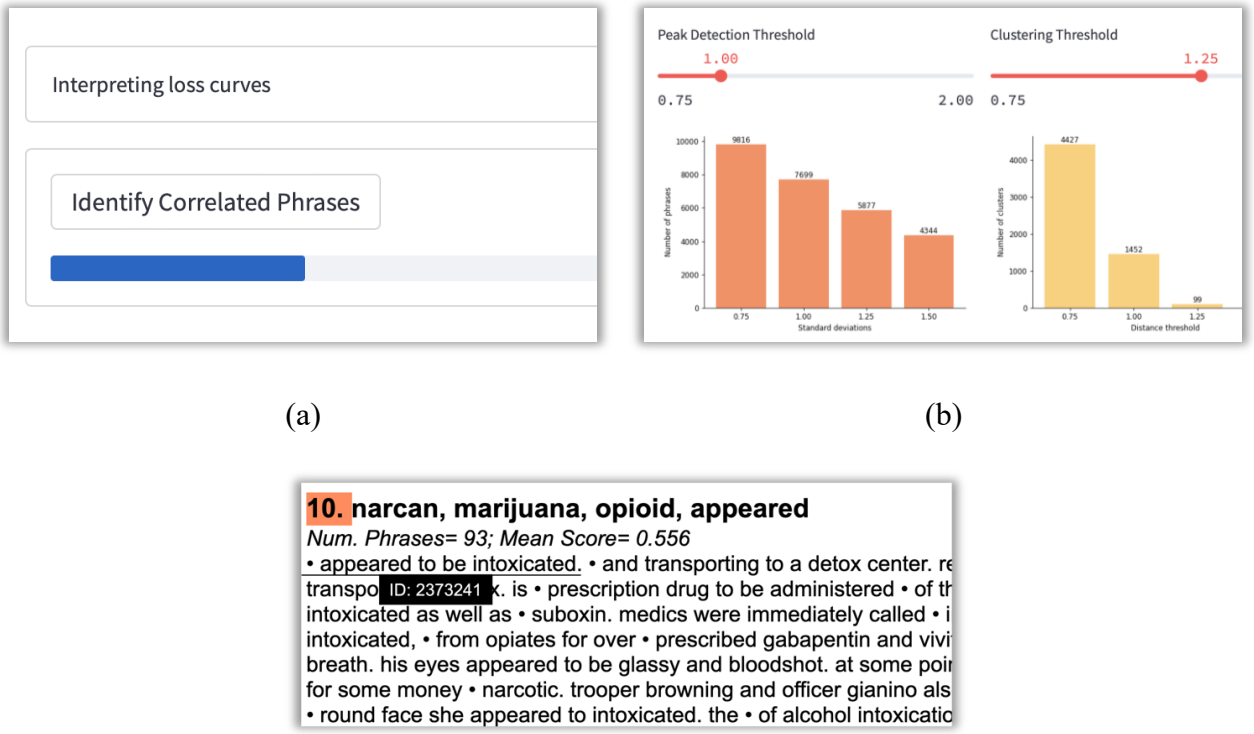


Figure 11. Web-based user interface of the developed tool

The back-end component was developed using the open-source Python programming language, which is the most popular language for implementation of machine-learning (ML) and AI solutions. Libraries utilized for the back-end development include the Numpy , Scipy, ScikitLearn (*33*) libraries, which are standard libraries for ML workflows in Python. Also, we leveraged the PyTorch package for implementation of the Deep Learning models and the Transformers (*29*) package for the fine tuning of the BERT and DistilBERT Models. Finally, we used the Sentence Transformer package (*28*) to compute the word embeddings for the clustering task.

During the development of the project, some of the IDEA panel members suggested to implement enhancements in the usability and functionalities of the tool. To enhance the usability and visual appearance of the developed system, we implemented some enhancements to the graphical interface as follows:

- First, as shown in Figure 12(a), we included a progress bar for all elements that involved an extensive background processing. This helps the user have a better idea of the time that the system will take to complete a given task. Examples of tasks for which we implemented a progress bar include the training of the text classifier, the identification of correlated phrases, and the clustering of the similar phrases to synthesize the results.
- Second, in order to enhance the usability of the tool, we included additional information about the parameters of the underlying analysis method, as shown in Figure 12(b). The additional information includes a sensitivity analysis for the Peak Detection Threshold and Clustering Threshold parameters, which help users understand the effect of these parameters on the outcome of the system. Using this information, users can adjust the degree of correlation and the number of clusters that they expect in the results. Although the documentation will include detailed explanations for these parameters, including the sensitivity analysis in the graphical interface helps users better interact with the system.
- Third, we implemented a mechanism to link identified phrases with the source crash report, by presenting the report ID, as illustrated in Figure 12(c). When the user hovers the mouse pointer over any of the phrases, the system displays a black and rectangular tooltip with the unique identifier of the crash report. This helps users to identify the origin of the phrase, which can facilitate further investigation of the report, if needed.



(a) (b)

(c)

Figure 12. Illustration of enhancements implemented on the graphical interface of the system

After implementing the visual and usability enhancements to the graphical interface of the system, we conducted two meetings with traffic safety personnel from our partner agencies to showcase the system, as described in the following subsection.

Feedback from Partner Agencies

To better understand the needs of transportation agencies and improve the system to better address such needs, we conducted meetings with collaborators from the Nevada and Massachusetts Department of Transportation.

On December 1st of 2022, we conducted a meeting with traffic safety personnel from the Nevada Department of Transportation (NDOT). Below we provide a summary of the feedback obtained in this meeting.

- Suggestion: Provide indications regarding the "target column" in the tool or in the documentation. This suggestion seeks to help users better understand the role of the target column (usually injury severity) in the analysis. This is because, at first glance, it is not

intuitive what is the expected information and format for this target column. We acknowledged this can be better explained in the tool or documentation, so we have included a description for this column in the tool's usage tutorial.

- Question: How does the system handle narratives from multiple agencies, written in different styles? We answered this question by explaining how the system works and emphasizing the underlying text classifier, which finds correlations between text data and a target field regardless of the data source. Also, we explained that the tool was tested in narratives from two different agencies (Massachusetts and Queensland, Australia), as well as a database from a domain different to crash data (movie reviews dataset) and the test results yielded meaningful results regardless of the origin of the dataset. The ability of the system to work across datasets of different nature suggests that data from different agencies will not impose significant limitations for the system.
- Question: How does the system handles potential ambiguity in phrases that contain negative statements (e.g., people transported to the hospital vs. people refusing being transported to the hospital)? We answered this question by explaining that the system finds recurrent patterns in the narratives. Therefore, the system will identify and clusters repetitions of similar statements regardless of their nature (affirmation or negation). This is not a limitation, as crash severity contributors may appear in the narratives in either affirmation or negation statements.
- Use case: Identification of specific mentions of crashes with animals. The collaborators from NDOT suggested that the system can be of great help to identify severe collisions involving animals. Given that the system picks up recurrent mentions of phrases correlated with severe crashes, there is high likelihood it may identify a cluster where collisions with animals is the main topic. However, the collaborators indicated that a tool to search for specific information in the narratives, regardless of the severity status, would be of great help to spot inconsistencies in the quantitative portions of the crash reports.

On December 8th of 2022, we conducted a meeting with traffic safety personnel from the Massachusetts Department of Transportation (MassDOT). In this meeting, we provided a live demo of the major functionalities and analyses enabled by the tool. Afterwards, we collected the participant's feedback and comments regarding usability and relevance of the results. Below we provide a summary of the feedback we obtained in this meeting.

- Use case: The keywords identified by the tool can be used as a search criterion to identify and help quantify the under-reporting of certain types of crashes. Our collaborator from MassDOT mentioned that the system offers potential to help address inconsistencies in the coded portion of the crash reports, as phrases identified by the system provide an indication of how officers describe certain crash circumstances. The collaborator specifically mentioned that the output of our system provides an extensive set of phrases and keywords to track down potential records miscoded as non-involving alcohol in the crashes, and this output allows them to further identify inconsistencies in recorded data.
- Use case: Contrast the findings of the tool against those collected by health institutions to fix inconsistencies in some factors. The tool could be used in narratives or descriptions from health institutions to support the identification of inconsistencies of crash factors such as the severity levels and DUI involvement.
- Suggestion: A tool for search of information in narratives would be of significant help to identify certain types of crashes. Given that inconsistencies in the coded portion of the crash reports are common, a system to automatically identify issues for specific types of crashes could be of significant help to minimize the amount of underreported or miscoded fields in the reports. As this is out of the scope of this project, we mentioned that this can be achieved in a future investigation and that we would be interested in proposing this as future NCHRP-IDEA project.

Potential Limitations of the Analysis Approach and Software Tool

When using an AI-based approach, it is essential to consider its potential limitations, which may affect its effectiveness and usefulness in specific contexts. Identifying and understanding these limitations is crucial in making informed decisions about the application of the approach and potential impact. In this section, we will highlight some of the potential limitations of the software tool to provide an overview of its capabilities and help users make informed decisions.

Bias prone algorithms

Algorithmic bias can occur when the data used to train the model is biased or when the algorithm itself is biased. This can result in unfair or discriminatory outcomes, such as the model making

incorrect assumptions or predictions based on race, gender, or other protected characteristics. For instance, for analysis of crash narratives, the software tool may identify phrases describing arbitrary groups from specific gender, race, or background as frequently associated with fatal crashes, so users should exercise caution when interpreting the results and take steps to mitigate potential biases.

Focus on pure correlations

The tool focuses on finding correlations, but for certain analysis this might cause inconvenient or noisy results. For instance, for analysis of crash narratives, the tool may identify as correlated with severe crashes the phrases that describe people transported to the hospital or involvement of fire departments to help victims. Although these phrases are clearly correlated with fatal crashes, they do not offer any insights on potential crash severity contributors. Therefore, users need to take this into consideration when analyzing the output of the tool.

Inherent random-based training and potentially unstable results

The training of the underlying text classifiers used to identify correlations involves random processes, which may result in different results every time the software is used. This is because the training process involves an optimization routine that may take different paths at every execution depending on the initialization of the underlying neural network. This may result in the text classifier paying more or less attention to certain types of phrases for different executions. For instance, for analysis of crash narratives, the text classifier may focus on phrases that describe alcohol-involvement in the crashes, whereas in another execution it may focus on phrases that describe involvement of pedestrians. To mitigate this potential issue, it is recommended to execute the tool multiple times and analyze the results of multiple executions.

Imperfect clustering

The clustering of text phrases is a problem that has no perfect solution up until now. The developed tool uses a clustering based on deep neural networks for semantic similarity, which is among the most sophisticated existing mechanisms for text clustering. However, the clustering results are still imperfect, and you might find some rare instances of phrases of different nature in the same cluster. This should be rare but still possible.

Exploratory nature of the analysis

The tool supports an analysis of exploratory rather than conclusive nature. Given that text or linguistic data may carry ambiguity and incomplete or inaccurate information, it is important to exercise caution when interpreting the results of the tool and avoid using the output of the tool as a single source for decision making. The tool seeks to offer a mechanism to easily extract insights from large databases of crash narratives, which can help analysts confirm or expand their understanding of crash factors, but given the noisy nature of linguistic data, the developed tool should be used in conjunction with classic analysis techniques based on robust statistical methods, such as Logit, Probit, and Ordered Logit.

PLANS FOR IMPLEMENTATION

This section describes the activities we conducted in order to transfer to practice the developed IDEA product. These activities include the release of the developed software under an open-source license, the hosting of a webinar directed at transportation agencies to showcase the software and its functionalities, the creation of a website to share the source codes, installation instruction, and user manual for the software.

Release of the software using an open-source license

In order to distribute and share the system to a broader audience, we released the developed approach under an open-source license. We published the source code and documentation of the tool in GitHub.com, the largest and most popular repository for open-source projects. GitHub.com enables visibility, collaboration, and community engagement, as this is a popular platform where data analysts search for projects, contribute with their own work, and allows users to engage with the project's community through discussions, forums, and pull requests. This is expected to enable other agencies across the U.S. to download it and analyze their own crash narratives. The open-source nature of the software will allow end-users to freely explore and revise the algorithm by integrating further evolving techniques. The specific open-source license used for the release of the software is the GNU General Public License v3.0, which allows users to freely redistribute it and/or modify it under the terms of the GNU General Public License, as published by the Free Software Foundation.

Hosting of a webinar directed at transportation agencies

On April 18th of 2023, we conducted a webinar directed at transportation agencies. The goal of the webinar was to make the tool known to potential transportation engineers who might benefit from extracting insights from crash narratives. During the webinar we provided step-by-step instructions on how to install and use the tool. We invited professionals from different public and private transportation agencies to the webinar. The contact information for these professionals was identified based on an online search in websites from multiple Departments of Transportations (DOTs) as well as the LinkedIn platform. The search focused on professionals with positions related to traffic safety, such as Traffic Safety Engineers, Crash Data Managers, Traffic Safety Researchers, and Managers of Traffic Safety Programs. Based on this search, we collected contact information and sent invitations via email to a total of 35 professionals in the traffic safety domain. From these 35 invitations, 15 people replied to our invitation and registered for the webinar, and 11 people attended the webinar. Figure 13 displays a summary of the position and organization of the professionals who participated in the webinar. This figure shows that a large proportion of webinar attendants were Traffic Safety Engineers and Crash Data Analyst, who are practitioners in the field, as well as Assistant Professors, who are researchers in the field. Also, Figure 13 shows that the webinar had the participation from professionals from public State agencies (Oregon DOT, Virginia DOT, and MassDOT), as well as private organizations that focus on crash data analysis (WSB Engineering).

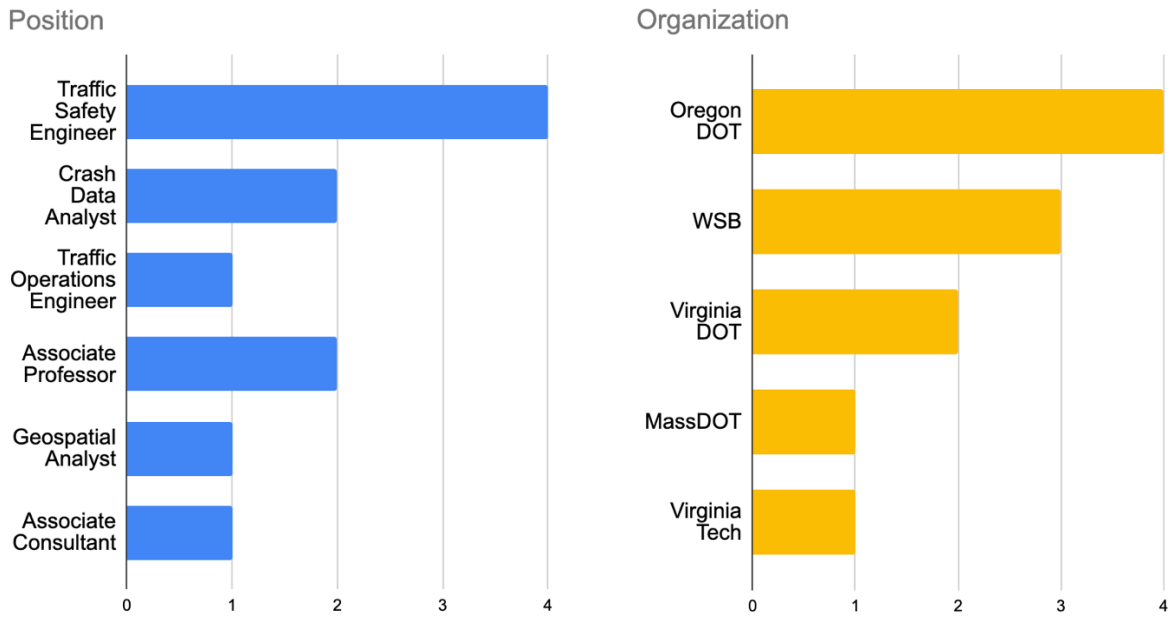


Figure 13. Positions and organizations of the professionals who participated in the webinar

Creation of a website to share information about the tool

We created a website to share information related to the developed software tool. The website is hosted at the following URL:

https://jeewoongpark.faculty.unlv.edu/research/tna/

As shown in Figure 14, the website contains six main sections, and the contents of these sections are as follows:

- The "Overview" section provides a general explanation of the tool and a graphical representation of the underlying approach and what it accomplishes.
- The "Installation" section provides step by step instructions on how to download and install the software as well as additional necessary software dependencies.
- The "Tutorial" section that explains how to use the tool in a step-by-step fashion by including instructions on how to prepare the data, select analysis parameters, run the approach through different steps, and analyze the results.
- The "License" section provides a detailed description of the GNU General Public License v3.0, which was the open-source license used to release the tool's source code.

- The “Limitations” section explains potential limitations of the software. The goal of this section is to explain users how to exercise caution when analyzing the results of the tool to avoid potential pitfalls.
- The “Acknowledgments” includes information about the funding sources that supported this project.
- The “Forum” section allows users to engage into conversations about the tool.

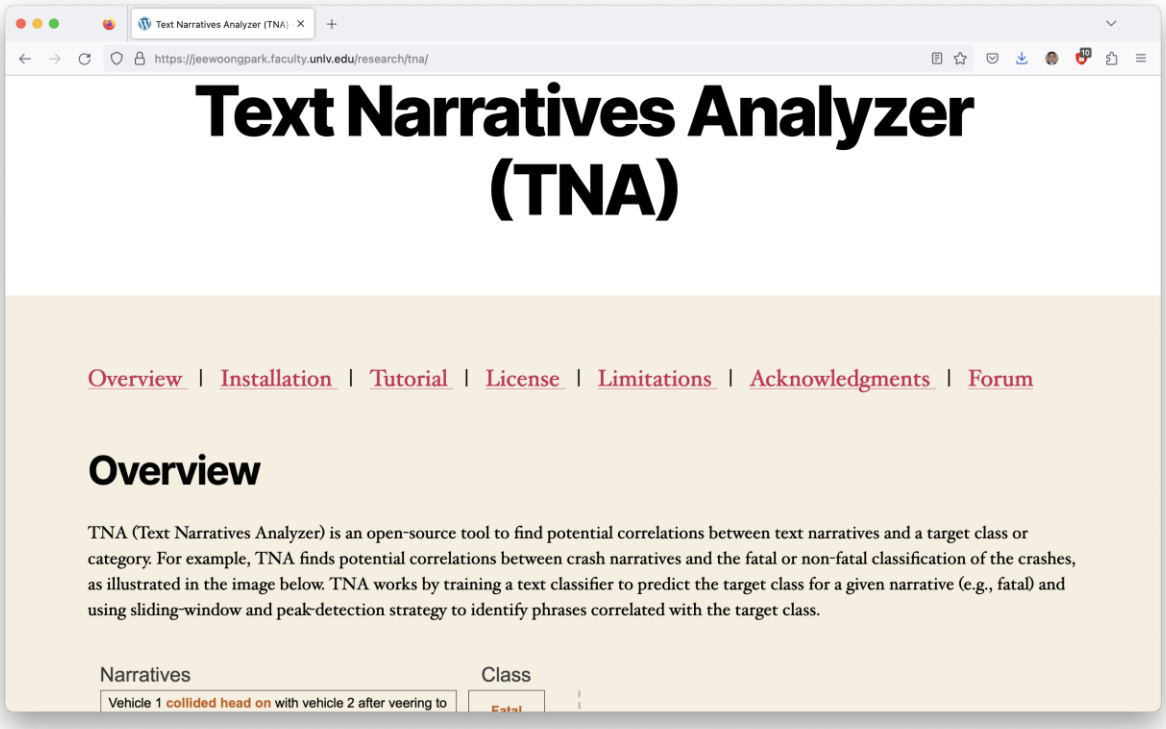


Figure 14. Illustration of the sections and contents of the created website

Presentation at transportation conferences

The research outcomes of this project were presented at the 2023 TRB Annual Meeting, which is the largest transportation conference in the world, where thousands of researchers and practitioners from the transportation and traffic safety domains present their latest work.

CONCLUSIONS

The novel AI-based tool that identifies correlations between phrases in traffic crash narratives and the severity of the crashes developed by this project offers significant contributions to the knowledge base of the transportation community. First, by leveraging the power of AI and NLP, the tool can extract valuable insights from textual crash narratives that were previously inaccessible to safety analysts. This allows for a more comprehensive and accurate understanding of the factors that contribute to traffic crashes. Second, the tool's ability to identify correlations between phrases and crash severity can help transportation agencies identify high-risk areas and develop targeted countermeasures to reduce the frequency and severity of crashes.

To develop the AI-based tool, we conducted an investigation that resulted in four major outcomes. Firstly, the devising of a novel Explainable AI approach that enables the extraction of correlations in the form of phrases between inputs and outputs of a text classifier. This approach overcame the limitations of existing word-level explanation techniques and was validated to successfully recover a large proportion of correlations in a dataset. Secondly, an identification and evaluation of state-of-the-art text classification models suitable for establishing correlation patterns between crash narratives and severities. Thirdly, a clustering-based technique to synthesize a large number of correlated phrases and provide a summary of potential severity contributors. Lastly, a web-based system with a user-friendly interface that enables easy interaction with the developed AI approach to identify severity contributors.

To validate the results obtained from the developed tool, we employed three methods. Firstly, we compared the results with those obtained from classic statistical analysis on quantitative data, which yielded a consistent level of similarity, thus suggesting the approach enables the extraction meaningful potential severity contributing factors from the crash narratives. Secondly, we employed an induction-based method, where correlations were intentionally induced in the narratives, to evaluate the ability of the underlying analysis approach to identify such correlations. The results of this validation indicated that phrases recurrently mentioned in the narratives for a given category can be successfully recovered by the approach with a high level of confidence. Lastly, we performed analysis on a dataset with known correlations, in which the approach managed to identify phrases that describe coherent associations with a given category of the text narratives.

Despite the promising results of the developed tool, we acknowledge that, similar to any AI implementation that involves the use of human-generated text data, there exist potential limitations that users need to take into consideration. These potential limitations include algorithmic bias, a focus on pure correlations, which may lead to noisy results, the use of inherent random-based training that may result in unstable and inconsistent results, imperfect clustering, and the exploratory nature of the analysis. Users should exercise caution when interpreting the output and avoid using it as a single source for decision-making, but rather use it in conjunction with classic analysis techniques based on robust statistical methods. Given the relevance of these potential limitations, this document dedicates a section entitled "Potential Limitations of the Analysis Approach and Software Tool" to discussing the potential limitations and potential pitfalls that users must consider when using and interpreting the results provided by the tool.

To facilitate the implementation and use of the developed tool, we published the source code and underlying analysis approach using an open-source license. This will allow other organizations throughout the United States to download and analyze their own crash narratives. By making the software open-source, users will have the freedom to use the tool without limitations of commercial licenses but will be additionally able to modify and enhance the tool and underlying algorithms to better fit their analysis needs, if needed. Also, we have created a website (https://jeewoongpark.faculty.unlv.edu/research/tna/) that contains all the necessary information to download and install the tool. Also, this website contains a step-by-step tutorial in written and video format that explains how to prepare the data, select analysis parameters, train the AI models, and interpret the results.

GLOSSARY

AI	Artificial Intelligence
NN	Neural Networks
DNN	Deep Neural Networks
NLP	Natural Language Processing
CNN	Convolutional Neural Networks
BERT	Bidirectional Encoder Representations from Transformers
GPU	Graphic Processing Unit
IMDB	The Internet Movie Database
ML	Machine Learning
GPT	Generative Pre-Training
NDOT	Nevada Department of Transportation
MassDOT	Massachusetts Department of Transportation

REFERENCES

1. WHO. *Global Status Report on Road Safety 2018*. 2018.
2. Bengio, Y., R. Ducharme, P. Vincent, C. Jauvin, J. U. Ca, J. Kandola, T. Hofmann, T. Poggio, and J. Shawe-Taylor. *A Neural Probabilistic Language Model*. 2003.
3. Collobert, R., and J. Weston. A Unified Architecture for Natural Language Processing. 2008.
4. 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*, 2013, pp. 3111–3119.
5. Sutskever, I. *Training Recurrent Neural Networks*. University of Toronto, CAN, 2013.
6. Kim, Y. Convolutional Neural Networks for Sentence Classification. 2014.
7. Sutskever, I., O. Vinyals, and Q. v. Le. Sequence to Sequence Learning with Neural Networks. *Advances in Neural Information Processing Systems*, Vol. 4, No. January, 2014, pp. 3104–3112.
8. Le, Q., and M. Schuster. A Neural Network for Machine Translation, at Production Scale. *Google AI Blog*. https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html.
9. Bahdanau, D., K. H. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *International Conference on Learning Representations*, 2015.
10. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30* (I. Guyon, U. v Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), Curran Associates, Inc., pp. 5998–6008.
11. Lin, T., Y. Wang, X. Liu, and X. Qiu. A Survey of Transformers. 2021.
12. Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale. 2020.
13. Lu, L., C. Liu, J. Li, and Y. Gong. Exploring Transformers for Large-Scale Speech Recognition. 2020.

14. Dufter, P., M. Schmitt, and H. Schütze. Position Information in Transformers: An Overview. *Computational Linguistics*, Vol. 48, No. 3, 2022, pp. 733–763. https://doi.org/10.1162/coli_a_00445.

15. Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistic*, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

16. Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv: 1907.11692*, 2019.

17. Sanh, V., L. Debut, J. Chaumond, and T. Wolf. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *ArXiv:1910.01108*, 2019.

18. Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut. ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations. *International Conference on Learning Representations (ICLR)*, 2020.

19. Thoppilan, R., D. de Freitas, J. Hall, N. Shazeer, A. Kulshreshtha, H.-T. Cheng, A. Jin, T. Bos, L. Baker, Y. Du, Y. Li, H. Lee, H. S. Zheng, A. Ghafouri, M. Menegali, Y. Huang, M. Krikun, D. Lepikhin, J. Qin, D. Chen, Y. Xu, Z. Chen, A. Roberts, M. Bosma, V. Zhao, Y. Zhou, C.-C. Chang, I. Krivokon, W. Rusch, M. Pickett, P. Srinivasan, L. Man, K. Meier-Hellstern, M. R. Morris, T. Doshi, R. D. Santos, T. Duke, J. Soraker, B. Zevenbergen, V. Prabhakaran, M. Diaz, B. Hutchinson, K. Olson, A. Molina, E. Hoffman-John, J. Lee, L. Aroyo, R. Rajakumar, A. Butryna, M. Lamm, V. Kuzmina, J. Fenton, A. Cohen, R. Bernstein, R. Kurzweil, B. Aguera-Arcas, C. Cui, M. Croak, E. Chi, and Q. Le. LaMDA: Language Models for Dialog Applications. 2022.

20. Shoeybi, M., M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism. 2019.

21. Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin,

S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language Models Are Few-Shot Learners. 2020.

22. Radford, A., K. Narasimhan, T. Salimans, and I. Sutskever. *Improving Language Understanding by Generative Pre-Training*. 2018.

23. Devlin, J., and M.-W. Chang. Open Sourcing BERT: State-of-the-Art Pre-Training for Natural Language Processing. https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

24. Ribeiro, M. T., S. Singh, and C. Guestrin. "Why Should I Trust You?" 2016.

25. Lundberg, S. M., P. G. Allen, and S.-I. Lee. A Unified Approach to Interpreting Model Predictions. 2017.

26. Mikolov, T., K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *International Conference on Learning Representations*, 2013.

27. Pennington, J., R. Socher, and C. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543.

28. Reimers, N., and I. Gurevych. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. 2019.

29. Wolf, T., L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush. Transformers: State-of-the-Art Natural Language Processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6.

30. Hong, Y., Y. Xin, H. Martin, D. Bucher, and M. Raubal. A Clustering-Based Framework for Individual Travel Behaviour Change Detection. No. 208, K. Janowicz and J. A. Verstegen, eds., 2021, pp. 4:1–4:15.

31. Dons, E., M. Laeremans, J. P. Orjuela, I. Avila-Palencia, A. de Nazelle, M. Nieuwenhuijsen, M. van Poppel, G. Carrasco-Turigas, A. Standaert, P. de Boever, T. Nawrot, and L. Int Panis. Transport Most Likely to Cause Air Pollution Peak Exposures in Everyday Life: Evidence from over 2000 Days of Personal Monitoring. *Atmospheric Environment*, Vol. 213, 2019, pp. 424–432. https://doi.org/10.1016/j.atmosenv.2019.06.035.

32. Arteaga, C., J. Park, P. B. Beeramoole, and A. Paz. Xlogit: An Open-Source Python Package for GPU-Accelerated Estimation of Mixed Logit Models. *Journal of Choice Modelling*, Vol. 42, 2022, p. 100339. https://doi.org/10.1016/j.jocm.2021.100339.

33. Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830.

APPENDIX: RESEARCH RESULTS

NCHRP IDEA Program Committee

June 2023

Project Title: Al Analyzer for Revealing Insights of Traffic Crashes

Project Number: NCHRP IDEA 231

Start Date: July 1st, 2021

Completion Date: June 30st, 2023

Principal Investigator:

Jee Woong Park, Associate Professor

E-Mail: jee.park@unlv.edu

Phone: 702-895-3936

TITLE:

Severity Contributors Identification in Traffic Crash Narratives

SUBHEAD:

AI-based tool to process crash narratives and identify potential crash severity contributing factors.

WHAT WAS THE NEED?

The prevalence of roadway crashes and the associated loss of lives and injuries constitute a significant public health concern. It is necessary to identify contributing factors to understand crashes better and prioritize the implementation of countermeasures.

WHAT WAS OUR GOAL?

To develop a tool that can identify potential severity contributing factors by uncovering correlations between the phrases in crash narratives and severe crashes. The tool was aimed at facilitating the use of crash narratives as a valuable information source for data-driven decision making.

WHAT DID WE DO?

This project relied on recent developments in Natural Language Processing and Artificial Intelligence to analyze crash narratives. We identified and evaluated text classification techniques, developed a novel Explainable AI approach, synthesized correlated phrases, and implemented a web-based system that integrates the developed analysis approach.

WHAT WAS THE OUTCOME?

The outcome was a tool that can identify potential severity contributing factors by uncovering correlations between the phrases in crash narratives and severe crashes. The tool trains a text classifier to establish correlation patterns, uses a Explainable AI approach to extract correlated phrases, and clusters phrases based on their semantic similarity to provide an overview of potential severity contributors.

WHAT IS THE BENEFIT?

The developed tool is expected to facilitate the use of crash narratives as a valuable information source for data-driven decision making. It requires minimal analyst intervention and can be used by agencies across the U.S. to analyze their own crash narratives. The open-source nature of the software will allow end-users to freely use the tool as well as explore and revise the algorithm by integrating further evolving techniques.

LEARN MORE

- IDEA project: http://apps.trb.org/cmsfeed/TRBNetProjectDisplay.asp?ProjectID=5191
- Website of the developed tool: https://jeewoongpark.faculty.unlv.edu/research/tna/

IMAGES