The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479896.144
Zhang, J., Kwigizile, V., and Oh, J. (2016) Automated Hazardous Action Classification Using Natural Language Processing and Machine-Learning Techniques. CICTP 2016: pp. 1579-1590. doi: 10.1061/9780784479896.144

Automated Hazardous Action Classification Using Natural Language Processing and Machine Learning Techniques

Jiansong ZHANG[1], Valerian KWIGIZILE[1], and Jun-Seok OH[1]

[1] Department of Civil and Construction Engineering, Western Michigan University, 4601 Campus Drive, G-238, Kalamazoo, MI 49008-5316; Phone: 269-276-3120; email: {jiansong.zhang, valerian.kwigizile, jun.oh}@wmich.edu

ABSTRACT

Information of hazardous action by involved party could support determination of crash responsibilities. However, the hazardous action information that are explicitly recorded in a crash report may often be inconsistent with the narrative given in a crash report. Identification of such inconsistencies requires a large amount of manual efforts. To address that, in this paper, a new method is proposed to classify hazardous action of a crash report automatically based on the narrative. The proposed method leverages natural language processing (NLP) techniques to extract features from the narratives, and machine learning (ML) techniques to classify the hazardous action of a narrative based on its values to selected features. The proposed method was preliminarily tested on a randomly selected set of crash reports from the State of Michigan. An accuracy of 92.77% and a Kappa statistic of 83.54% were achieved on the testing data, which shows that the proposed method is promising.

INTRODUCTION

In the State of Michigan, U.S.A., traffic accidents are documented in UD-10 traffic crash reports. Each crash report records the crash date, crash time, crash type, number of units involved, weather, road condition, and many other types of information. Two important pieces of information in a UD-10 crash report are the hazardous action and the narrative of a crash given by the investigating officer. Hazardous action indicates whether a person is “at fault” in a crash, including the following 16 types: none, speed too fast, speed too slow, failed to yield, disregard traffic control, drove wrong way, drove left of center, improper passing, improper lane use, improper turn, improper/no signal, improper backing, unable to stop in assured clear distance, other, unknown, reckless driving, and careless/negligent driving (Michigan Department of State Police Criminal Justice Information Center 2010). Narrative is a description of the crash using a police officer’s words, including the number and type of persons and

vehicles involved, evidence from witnesses of the crash scene, level of injuries of involved persons, etc. The detailed information given in a narrative could be interpreted to determine the hazardous action of a unit (i.e., a person involved in the crash). However, inconsistencies have been observed between the hazardous action explicitly documented in crash reports and the hazardous actions interpreted from the corresponding narratives. This could be confusing when crash reports are used to support determination of crash responsibilities. Because a narrative provides more detailed information of a crash, hazardous action interpreted from the narrative tends to be more convincing than the explicitly documented hazardous action when inconsistency occurs. Identification of such inconsistencies between the narrative and explicitly documented hazardous action in a crash report requires a read-through of the narrative, interpretation of relevant information in the narrative, and understanding of different hazardous action types. This requires a large amount of manual efforts for the hundreds of thousands of crash reports annually generated for a state. To reduce the amount of manual efforts that are needed in this identification process, this paper proposes an automated hazardous action classification method. The proposed method processes the narratives of crash reports using natural language processing (NLP) techniques to generate features to represent the narratives, and train machine learning (ML) classifiers based on feature values and corresponding hazardous action classes in training data (i.e., narratives from selected crash reports and their corresponding hazardous actions based on interpretation) to use for automated classifying the hazardous action of any UD-10 crash report.

BACKGROUND

Natural Language Processing

NLP is a subdomain of artificial intelligence which aims to enable computers to understand and process natural language texts and speeches in a human-like manner (Cherpas 1992). NLP has many different types of tasks such as information retrieval, information extraction, and text classification. Information retrieval aims to retrieve a set of documents relevant to a textual string query (Stanford NLP Group 2009). Information extraction aims to extract desired information from text sources according to predefined information templates (Zhang and El-Gohary 2013). Text classification aims to identify the category or categories to which a piece of text or document belongs (Russel and Norvig 2010; Salam and El-Gohary 2013). To support these NLP tasks, unigrams (i.e., single words), bigrams (i.e., two consecutive words), and n-grams (i.e., a sequence of more than two words) are frequently used as features. For example, Jiffriya et al. (2014) used trigram vector space model to detect plagiarism on electronic text (i.e., a type of information retrieval); Balijepalli (2007) used unigrams and bigrams

in a domain independent system to extract subjective sentences from political blogs (i.e., a type of information extraction); and Verma et al. (2007) used unigrams and bigrams together with some other features in their representation of tweets data (i.e., snippets of text) to classify situation awareness information of the tweets data (i.e., a type of text classification).

Machine Learning and Feature Selection

ML lies at the core of artificial intelligence. ML aims to enable computers to improve automatically through experience (Jordan and Mitchell 2015). Supervised learning and unsupervised learning are two major ML paradigms. In supervised learning, labeled data are given to the learning algorithm to learn a mapping from features to a label prediction. In unsupervised learning, unlabeled data are provided to the learning algorithm to analyze structural properties of the data (Jordan and Mitchell 2015). If labeled data are available or the cost of labeling needed training data is not prohibitive, supervised learning would typically be selected over unsupervised learning. Many ML algorithms have been developed over the years for supervised learning. Naïve Bayes is a simple ML algorithm which applies Bayes' rule to calculate conditional probabilities of labels given the features. It is simple but very effective, and could outperform more complex learning algorithms in certain cases (Domingos 2012). Decision tree is a little more complex than Naïve Bayes. A decision tree ML algorithm learns a tree-like structure to represent the influence of each feature on the final label prediction. Each branch in a decision tree represents a feature value and each leaf node represents a predicated label. Decision trees are human interpretation-friendly because they can be easily interpreted as a set of decision rules (Zaki and Meira 2014). Support vector machines (SVM) ML algorithm has an advantage over other ML algorithms because SVM converts a nonlinear model to a linear model through the use of kernels (Kecman 2005). When using these ML algorithms to conduct supervised learning, labeled data need to be provided. The labeled data are typically represented as values according to a collection of features. What features to use for representing the data is closely related to the selection of a specific ML algorithm, the same set of features typically leads to different performance when using different ML algorithms. For a specific ML algorithm, a collection of discriminating features needs to be selected to sufficiently leverage the learning capacity of the ML algorithm so that a best (or suboptimal but acceptable) performance could be achieved.

The Use of ML in Ground Transportation Crash Analysis

With the advancement in ML, these techniques started to be used in crash analysis of ground transportation. For example, Ona et al. (2012) adapted decision tree ML algorithm to extract decision rules from accident reports for road safety analysis;

Li et al. (2012) trained an SVM model over crash data collected at 326 freeway diverge areas for predicting the injury severity of any individual crash; and Matias et al. (2007) tested SVM and two other ML algorithms on modeling the degree of remedial action required to make roadway suitable for dangerous goods transportation. However, these types of efforts are still limited according to the authors' search over literatures. The authors were not aware of any existing work leveraging narrative data in crash reports for crash analysis, in spite of the detailed crash information described in the narratives.

PROPOSED HAZARDOUS ACTION CLASSIFICATION METHOD

To reduce the amount of manual efforts needed to check the consistencies between the hazardous actions obtained based on interpreting narratives and the hazardous actions explicitly recorded in crash reports, a new hazardous action classification method which automatically classifies the type of a hazardous action based on information from a narrative is proposed, including the following six steps (Figure 1): (1) Data preparation, including preparing both the training data and testing data from crash reports for further ML training and testing steps; (2) narrative processing, which processes narratives from the prepared data using NLP techniques into unigrams, bigrams, and trigrams; (3) feature selection, which selects a specific set of unigrams, bigrams, and trigrams as the features to be used in the ML training and testing steps; (4) machine learning training, which trains ML models using the prepared training data and selects the best performing ML algorithm; (5) classifier tuning, which tunes the parameters of the best performing ML algorithm targeting at its best performance; and (6) final testing, which tests the trained classifier on the prepared testing data using the best ML algorithm and its tuned parameters.

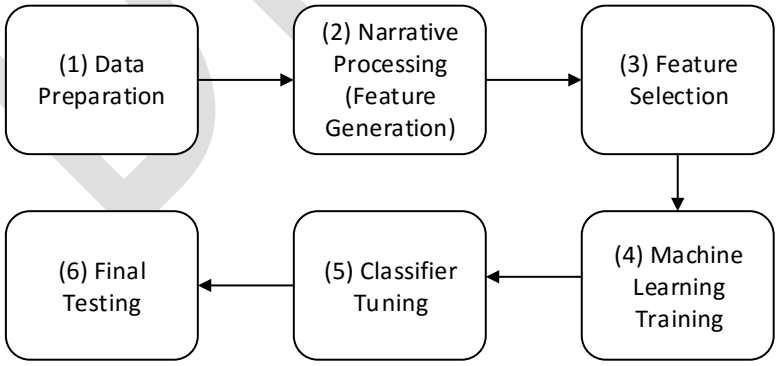

Figure 1. Proposed hazardous action classification method

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479896.144
Zhang, J., Kwigizile, V., and Oh, J. (2016) Automated Hazardous Action Classification Using Natural Language Processing and Machine-Learning Techniques. CICTP 2016: pp. 1579-1590. doi: 10.1061/9780784479896.144

Data Preparation

The data preparation step aims to prepare both training and testing data for the machine learning training and testing steps. The training and testing data come from randomly selected crash reports. A general rule of thumb of splitting a data source is to assign 60% to 80% of the data to the training set and the remaining data to the testing set (Borovicka et al. 2012). In the proposed method an arbitrary split of 70% to 30% between the training and testing data sets is adopted. The optimal split depends on the type of data and is outside the scope of this paper. Within the training set and testing set crash reports, handwritten ones are removed and narratives are extracted. Manual interpretation of the narratives are then conducted and the selected hazardous actions are documented together with their source narratives.

Narrative Processing (Feature Generation)

The narrative processing step aims to generate features for each narrative. NLP tools are used to process a narrative to generate unigrams, bigrams, and trigrams. Before the generation of unigrams, bigrams, and trigrams, stemming is applied to each word to avoid different forms of a word being recognized as different words. In the collection of narratives (including the narratives in both the training set and the testing set), the frequency (i.e., number of occurrences) of each unigram, bigram, or trigram is also calculated. The unigrams, bigrams, and trigrams with a frequency of at least two are preliminarily selected as candidate features.

Feature Selection

Feature selection aims to select from the candidate features to find the ones that are more indicative of (i.e., in a collective manner) the hazardous action class. We call these features discriminating features. Feature selection is conducted in two steps: (1) search possible feature subsets; and (2) evaluate the correlation of each feature subset and the class value, and pick the subset that has the highest correlation with the class value. Different types of search methods could be utilized in the first step, such as exhaustive search (i.e., search all combinations of features), best first search (i.e., applying the classic best first search strategy), and greedy step-wise search (i.e., applying a forward or backward step-wise search strategy). Each narrative is then represented as a tuple of zeros and ones according to it having or not having the corresponding unigram, bigram, or trigram.

Machine Learning Training

Machine learning training aims to pick a best ML algorithm for the classification task based on comparing several ML algorithms' performance on the training data. Naïve Bayes, decision tree, and SVM are compared in the training. Ten-

fold cross-validation is used in the training of each ML algorithm to avoid overfitting of the trained model on training data.

Classifier Tuning

Classifier tuning aims to empirically tune the parameters (i.e., if there are adjustable parameters) of the selected ML algorithm to further improve the performance of a trained classifier. For Naïve Bayes, no parameters need to be tuned. For decision tree, two parameters will be tuned: (1) confidence factor, which controls the amount of post-pruning used. The higher a confidence factor is, the more post-pruning will be used; and (2) minimum number of objects, which dictates the minimum number of instances needed at each node. For SVM, two types of kernels will be used, including Gaussian Kernel and Polynomial Kernel. Two parameters will be tuned for both types of kernels used. If the Gaussian Kernel is used, then the soft margin constant and the inverse width parameter of the Gaussian Kernel will be tuned. If the Polynomial Kernel is used, then the soft margin constant and the exponent of the Polynomial Kernel will be tuned.

Final Testing

Final testing aims to test the trained classifier using the selected ML algorithm and its tuned parameters, on testing data. Accuracy and Kappa statistic are used to evaluate the performance of the trained classifier. Accuracy is the portion of the instances that are correctly classified. Kappa statistic removes the correct classification by chance from accuracy.

EXPERIMENTAL TESTING

Experimental set up

To test the proposed hazardous action classification method, 290 recent crash reports from the State of Michigan were randomly selected as the data source, including 148 crash reports in 2013 and 142 crash reports in 2014. Among the 290 crash reports, 14 handwritten reports were removed because the narrative information in handwritten reports were not easily extractable. Narratives from the remaining 276 crash reports were manually interpreted and assigned their hazardous actions. When determining the crash responsibilities, the most important distinction in the hazardous action is whether the hazardous action is “none” or not “none”. Therefore the experiment focused on the binary classification of a narrative into “none” or “hazardous” (i.e., not “none”). Essentially, the interpreted “none” class directly mapped to the “none” type in explicitly recorded hazardous actions, while the interpreted “hazardous” class maps to the remaining 15 types of explicitly recorded hazardous actions in a UD-10 crash

repoart. Through manual interpretation 92 narratives were assigned the "none" class and 184 narratives were assigned the "hazardous" class. Applying the 70% to 30% splitting mechanism between training and testing data, 64 narratives in the "none" class and 129 narratives in the "hazardous" class were collected into the training data set, and 28 narratives in the "none" class and 55 narratives in the "hazardous" class were collected into the testing data set. Figure 2 shows two example narratives that were assigned "none" and "hazardous" classes, respectively.

Narrative	Hazardous Action
DRIVER #1 WAS S/B ON PLAINVIEW AT HIGH RATE OF SPEED. DRIVER #2 WAS W/B ON TIREMAN WHEN DRIVER#1 DISREGUARD STOP SIGN. DRIVER #2 HIT DRIVER #1. DRIVER #1THEN LOST CONTROL; ROLLED OVER ON TO PARKED VEH#3. VEHICLE #3 THEN HIT PARKED VEHCILE #4. VEHICLE#4 THEN HIT PARKED VEHICLE #5. VICTIM IN VEHCILE #1 (REAR PASSENGER) WAS K TYPE DUE TO ACCIDENT. AS VEHICLE #1 BEGAIN TO ROLL, A PEDESTRAIN WAS HIT WHILE STANDING ON THE SIDE OF VEHICLE#3	Hazardous
UNIT 1 DRIVING WEST BOUND TEN MILE IN RIGHT LANE DID NOT SEE PEDESTRIAN IN ROADWAY AND STRUCK PEDESTRIAN. \NUNIT 1 STOPPED IMMEDIATELY AND CALLED 911.	None

Figure 2. Sample narratives and hazardous action classes

Tools used in the experiment

The Python programming language (v.2.7.3) was used to build a program for conducting the experiment. The TextBlob library (Loria 2015) for processing textual data was used to generate unigrams, bigrams, trigrams, and their frequency of the collected narratives (including both training and testing data sets). The Porter Stemmer (Porter 1980) was used to find the stems of the words. The attribute selection tool in Waikato Environment for Knowledge Analysis (Weka) data mining software system (Hall et al. 2009) was used for feature selection. The best first search strategy was used in the feature combination search task of the feature selection step. In the environment of Weka, Weka.classifiers.bayes.NaiveBayes was used for Naïve Bayes ML algorithm, weka.classifiers.trees.J48 was used for decision tree ML algorithm, and weka.classifiers.functions.SMO was used for SVM ML algorithm.

EXPERIMENTAL RESULTS AND DISCUSSION

Experimental results

In the feature generation step, 2950 unigrams, 11063 bigrams, and 16114 trigrams were generated. Eliminating those features with a frequency of only one, 1450 unigrams, 2183 bigrams, and 1522 trigrams were teased out as feature candidates. Through feature selection, 99 unigrams and bigrams were then selected to constitute

the feature set to use in machine learning. It was interesting to notice that no trigrams were selected. Figure 3 shows part of the selected features. The terms appear in Figure 3 were already stemmed using Porter Stemmer.

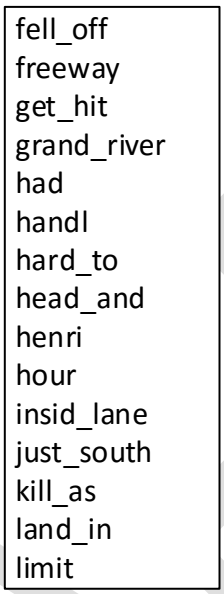

Figure 3. Sample unigrams and bigrams in the selected features

Using the selected 99 features and 10-fold cross validation, the performance of Naïve Bayes, decision tree, and SVM were shown in Table 1. Naïve Bayes outperformed the other two ML algorithms and was therefore selected for final testing. Naïve Bayes does not have parameters to adjust so the classifier tuning step was skipped in the experiment. In final testing, the Naïve Bayes classifier trained on the training data was tested on the testing data. As a result of the final testing an accuracy of 92.77% and a Kappa statistic of 83.54% were achieved (Table 2).

Table 1. Machine learning training results

ML algorithm	Training Accuracy
Naïve Bayes	83.94%
Decision tree	67.88%
SVM	80.32%

Table 2. Final testing results

ML algorithm	Accuracy	Kappa statistic
Naïve Bayes	92.77%	83.54%

The published version is found in the ASCE Library here: http://ascelibrary.org/doi/abs/10.1061/9780784479896.144
Zhang, J., Kwigizile, V., and Oh, J. (2016) Automated Hazardous Action Classification Using Natural Language Processing and Machine-Learning Techniques. CICTP 2016: pp. 1579-1590. doi: 10.1061/9780784479896.144

Discussion

In our experiment, the trained Naïve Bayes classifier only made 6 mistakes in classifying the testing data comparing to manual interpretation results. Figure 4 visualizes the errors made by our classifier. The top left corner of Figure 4 highlights the four instances (i.e., Instances 1, 14, 18, and 26) that were incorrectly classified as "hazardous," and the bottom right corner of Figure 4 highlights the two instances (i.e., Instances 33 and 60) that were incorrectly classified as "none." An error analysis was conducted on the incorrectly classified instances. Table 3 listed all the incorrectly classified instances of narratives with their instance numbers, interpreted hazardous action classes, predicated hazardous action classes, dominant features (i.e., the unigrams or bigrams that they included), and the distribution of training instances between "none" and "hazardous" classes for each dominant feature. A main observation is that most of these dominant features have a relatively flat distribution of training instances between "none" and "hazardous" classes. For example, the bigram "vehicle_1" has a distribution of training instances of 15 to 19 between "none" and "hazardous" classes, which is about 44% to 56%. It is likely the case that these features (when dominating the prediction) would not be as discriminating as other features who had more tilted distribution in training instances.

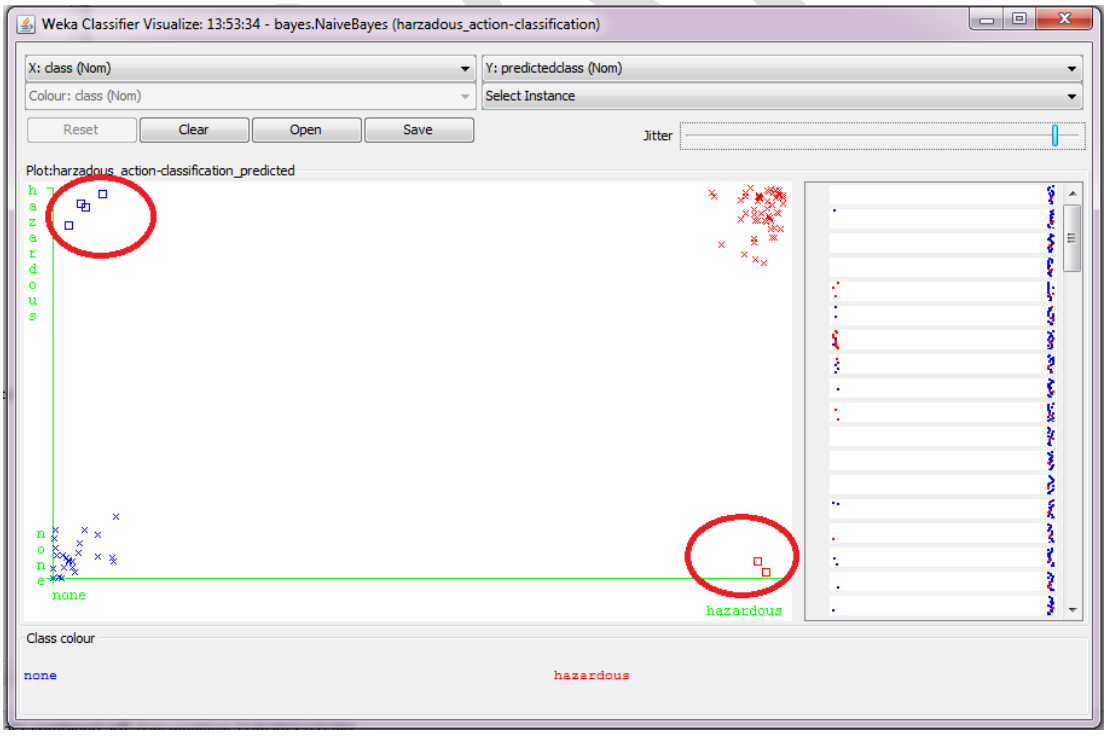

Figure 4. Classifier error visualization

Table 3. Error analysis results

Instance number	Interpreted class	Predicted class	Dominant features	Training instances distribution	
				None	Hazardous
1	none	hazardous	vehicle_1	15	19
14	none	hazardous	vehicle_1	15	19
			rd	18	21
18	none	hazardous	rd	18	21
			attempt	1	1
26	none	hazardous	vehicle_1	15	19
			accid	6	27
33	hazardous	none	west	18	16
			had	18	20
60	hazardous	none	bound	11	10
			is_a	2	4
			rd	18	21
			west	18	16

CONCLUSIONS

Hazardous action information in crash reports could be used to support determination of crash responsibilities. However, inconsistencies have been observed between the narrative of a crash report and the corresponding hazardous action that is explicitly documented. Manual identification of such inconsistencies would be very time consuming and tedious for the hundreds of thousands of crash reports annually generated in a state. To help with such identification, this paper proposed an automated hazardous action classification method with six steps. The method utilizes natural language processing techniques and machine learning techniques. Unigrams, bigrams, and trigrams are used as features of narratives in crash reports to train machine learning classifiers. Naïve Bayes, decision tree, and support vector machines ML algorithms are experimentally compared based on their training performance on training data, and the ML algorithm with the best performance is tuned to train the final classifier. The proposed method was tested on binary hazardous action classification (i.e., “hazardous” or “none”) of narratives from 276 randomly selected crash reports in the State of Michigan from 2013 to 2014. Naïve Bayes ML algorithm outperformed the other two ML algorithms in the experiment, and achieved an accuracy of 92.77% and a Kappa statistic of 83.54% in final testing. This good performance shows that the proposed automated hazardous action classification method is promising. An error analysis was conducted where we found that the errors made were likely to be caused by features which were not very discriminating when they dominated the classification results.

REFERENCES

Balijepalli, S. (2007). "Blogvox2: a modular domain independent sentiment analysis system." Master Thesis, University of Maryland, College Park, MD.

Borovicka, T., Jirina, M.Jr., Kordik, P., and Jirina, M. (2012). "Selecting representative data sets." *Advances in Data Mining Knowledge Discovery and Applications*, InTech.

Cherpas, C. (1992). "Natural language processing, pragmatics, and verbal behavior." *The Analysis of Verbal Behavior*, 10, 135-147.

Domingos, P. (2012). "A few useful things to know about machine learning." *Communications of the ACM*, 55(10), 78-87.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. (2009). "The WEKA data mining software: an update." *SIGKDD Explorations*, 11(1), 10-18.

Jiffriya, M., Jahan, M.A., and Ragel, R.G. (2014). "Plagiarism detection on electronic text based assignments using vector space model." arXiv:1412.7782.

Jordan, M.I., and Mitchell, T.M. (2015). "Machine learning: trends, perspectives, and prospects." *Science*, 349(6245), 255-260.

Kecman, V. (2005). "Support vector machines – an introduction." *StudFuzz*, 177, 1-47.

Li, Z., Liu, P., Wang, W., and Xu, C. (2012). "Using support vector machine models for crash severity analysis." *Accident Analysis and Preventions*, 45, 2012.

Loria, S. (2015). "TextBlob: simplified text processing." <https://textblob.readthedocs.org/en/latest/> (Nov. 9, 2015).

Matias, J.M., Taboada, J., Ordonez, C., and Nieto, P.G. (2007). "Machine learning techniques applied to the determination of road suitability for the transportation of dangerous substances." *Journal of Hazardous Materials*, 147, 60-66.

Michigan Department of State Police Criminal Justice Information Center. (2010). "State of Michigan UD-10 traffic crash report manual." < https://www.michigan.gov/documents/UD-10_Manual_2004_91577_7.pdf> (Nov. 7, 2015).

Ona, J.D., Lopez, G., and Abellan, J. (2013). "Extracting decision rules from police accident reports through decision trees." *Accident Analysis and Prevention*, 50, 1151-1160.

Porter, M. (1980). "An algorithm for suffix stripping." *Program (Autom. Libr. and Inf. Syst.)*, 14(3), 130-137.

Russel, S., and Norvig, P. (2010). "Artificial intelligence: A modern approach." 3rd Ed., Prentice Hall, New York, NY.

Salama, D. and El-Gohary, N. (2013). "Semantic text classification for supporting automated compliance checking in construction." *Journal of Computing in Civil Engineering*, 10.1061/(ASCE)CP.1943-5487.0000301, 04014106.

Zhang, J., Kwigizile, V., and Oh, J. (2016) Automated Hazardous Action Classification Using Natural Language Processing and Machine-Learning Techniques. CICTP 2016: pp. 1579-1590. doi: 10.1061/9780784479896.144

Stanford NLP Group. (2009). "Evaluation of information retrieval." IR-book. Cambridge University Press, New York, NY.

Verma, S., Vieweg, S., Corvey, W.J., Palen, L, Martin, J.H., Palmer, M., Schram, A., and Anderson, K.M. (2007). "Natural language processing to the rescue?: extracting 'situation awareness' tweets during mass emergency." In Lada A. Adamic, Ricardo A. Baeza-Yates, and Scott Counts, editors, ICWSM. The AAAI Press, 2011.

Zaki, M.J., and Meira, W.Jr. (2014). "Decision tree classifier." *Data Mining and Analysis, Fundamental Concepts and Algorithms*. Cambridge University Press, New York, NY.

Zhang, J. and El-Gohary, N. (2013). "Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking." *Journal of Computing in Civil Eng*ineering, 10.1061/(ASCE)CP.1943-5487.0000346, 04015014.