May 2023

Extracting Patterns of Semantic Roles from Accident Narratives

Soundarya Jayakumar
University of Wisconsin-Milwaukee

EXTRACTING PATTERNS OF SEMANTIC ROLES FROM ACCIDENT NARRATIVES

by

Soundarya Jayakumar

A Thesis Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Master of Science
in Computer Science

at

The University of Wisconsin–Milwaukee
May 2023

ABSTRACT

EXTRACTING PATTERNS OF SEMANTIC ROLES FROM ACCIDENT NARRATIVES

by

Soundarya Jayakumar

The University of Wisconsin–Milwaukee, 2023
Under the Supervision of Professor Rohit J. Kate

Accident databases are filled with rich information about accidents. Analyzing these datasets can reveal useful information which can be used to prevent similar accidents in the future. Policy makers, and safety management organizations can design appropriate measures based on the analysis done to prevent accidents. Besides structured data, crash reports include natural language narratives which contain valuable accident-related information which is otherwise not present in the structured data. Using natural language processing (NLP) techniques one can analyze these narratives and mine hidden patterns of accidents from them. The thesis focuses on developing an algorithm to extract common patterns of semantic role labels from the narratives of accidents. These patterns capture frequently occurring sequences of verbs and their arguments. In this work, the developed algorithm was applied to accident narratives and the resulting patterns were assessed for their accident-related information.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
SRL	Semantic Role Labelling
TOPS Lab	Wisconsin Traffic Operations and Safety Laboratory
WisDOT	Wisconsin Department of Transportation
SRL	Semantic Role Labeling
BERT	Bidirectional Encoder Representations from Transformers
OOP	Object Oriented Programming
LDA	Latent Dirichlet Allocation
OSHA	Occupational Safety and Health Administration
BIO	Beginning Inside Outside

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Rohit J. Kate for his time and support throughout my entire graduate studies. His valuable comments and motivation strengthened me in successfully delivering my thesis. I would like to thank all my thesis committee members: Prof. Susan Mcroy and Prof. Xiao Qin for their time and sharing valuable suggestions.

I also want to thank my family and friends for morally supporting me throughout my graduate studies.

Introduction

1.1. Background and Problem Statement

Roadways, the most commonly used mode of transportation, has faced some of the most gruesome accidents than any other modes of transportation. This is due to several factors like weather, road repairs, inattentiveness while driving and emergency situations. According to the Department of transportation of the state of Wisconsin [16], based on the WisDOT-Traffic Crash Database, a total of 136,007 crashes have occurred on the public roadways from 2015 to 2020.

In order to build an optimized operational management of roadways, understanding the nature of crashes is crucial. Although the structured data of accident databases carry good evidence of crashes, the natural language narratives which are unstructured are of major attention when it comes to finding interesting patterns. The accident report databases tend to possess large, error filled, and complex information along with missing and/or redundant data [2]. On the accident sites, based on the severity of the accidents, the structured data being filled in the report becomes messy. The officer in charge often fails to provide some information for the optional fields. But when it comes to narratives, the natural flow of human language provides most of the highlighted features related to the accidents occurring in a site. The analysis based upon the structured data only is not sufficent [3].

The rate of accidents is increasing every year worldwide. In order to establish work safety principles, analyzing the historical crash reports is crucial. Unstructured data has a lot of scope to derive patterns to analyze the nature and cause of crashes. This in turn helps build robust work safety principles to prevent future crashes of the same nature. But the true potential for extracting meaningful patterns lies in the narratives of the crash reports. Being the unstructured part of the crash record, it requires powerful techniques to extract patterns from them.

1.2. Literature Review

The authors of [10] have analyzed the text data present in the construction accident reports using NLP techniques. They have proposed five base models using support vector machine, linear regression, k-nearest neighbor, decision tree, naive bayes, and an ensemble model to build classifiers for classifying the records according to the cause of accidents. [11] proposes a natural language data augmentation-based training framework for automatic information extraction from safety related documents. Their model was then validated to an accident news report dataset. They have employed text augmentation algorithms to mitigate the limitation in the data sources and the lack of large-scale datasets. In [12], a chemical accident database has been analyzed using natural language techniques to identify causal relationships about the accidents that occurred. They have enriched the database by using web scraping techniques and populated with pre-defined ontology. The ontology based chemical database yielded accident-related information from the database. In [13], the authors have identified the accident process in different fields like manufacturing, chemicals, construction and service using narrative records. They stress how valuable narratives could be in extracting complex accident patterns which otherwise is not possible. First, the records are categorized into respective sectors following which the patterns are extracted. Text mining and Latent Dirichlet Allocation (LDA) algorithms have been used to perform the first task of categorizing the dataset into different sectors and the factors associated with the matching sectors. For this work, the authors have chosen the Occupational Safety and Health Administration (OSHA) dataset which identifies the five sectors mainly scale-intensive, facility- intensive, supplier-dominated, market-dominated, and service-dominated patterns. This study was also done for OSHA which ensures safe working conditions by setting standards thereby improving livelihood. Another study which was conducted for OSHA by [14] focused on developing an unsupervised clustering algorithm to cluster the database into different sectors using NLP techniques. The work zone crashes have been analyzed and misclassified reports have been identified in [1]. The narrative records in the crash reports have been utilized for this task as well. The dataset consisted of 300K crash reports acquired from the Wisconsin Department of Transportation. It is already classified as work zone and non-work zone crash records from which the misclassified records are identified. The Noisy-OR method and unigram + bigram methods have been used to compare the results of misclassified records.

To the best of our knowledge, no previous work has used semantic role labeling to find patterns of accidents.

1.3. Motivations and objectives

Icy road conditions or other weather conditions, emergency situations like driving an ambulance, fire vehicles are also a common factor of accidents. For a particular region, given the data, if one could find the commonly occurring patterns, it will help the target users know the most frequently occurring cause and location for the accidents. The datasets are sparse with the required information when it comes to inspecting the structured data. There seems to be a bias or irregularities while entering the right data in the forms explaining the accidents. [3]

The motivation is to make the best use of the natural language narrative recorded by the officer in charge during the accident and find patterns of accidents. These narratives contain the key information of how the accidents have occurred, the reason, nature and cause of the accident along with the environmental conditions in the accident zone.

1.4. The Main Contributions

The main contributions of the thesis are – utilizing semantic role labeling for finding accident patterns from narratives, defining patterns of semantic role labels, and developing an algorithm for finding them. The frequency of occurrence of these patterns was also computed from the narratives to judge their prevalence.

Materials and Methods

2.1. Dataset

The dataset consists of a total of 101,510 records. These correspond to the reports generated from the year 2020 in Wisconsin state. Each record has the following features: *'OFFRNARR'*, *'CRSHNMBR'*, *'DOCTNMBR'*, *'DISTACT1'*, *'DISTACT2'*, *'DISTFLAG'*, *'DISTSRC1'*, *'DISTSRC2'*, *'DRVRPC1A'*, *'DRVRPC1B'*, *'DRVRPC1C'*, *'DRVRPC1D'*, *'DRVRPC2A'*, *'DRVRPC2B'*, *'DRVRPC2C'*, *'DRVRPC2D'*, *'DISCON'*, *'DISALL'*, *'INATCON'*, *'INATALL'*. Out of these features, our thesis mainly focuses on *'OFFRNARR'* which corresponds to the natural language narratives recorded by the police in charge at the accident zones. The dataset reports are provided from the *Wisconsin Transportation Portal* (WisTransPortal). This portal comprises several datasets of crash reports that occurred in the state of Wisconsin since 1994. The personal information of the accidents has been removed from the dataset. In the narrative the officers on the scene identify the involved vehicles as *"Unit 1"* or *"Unit 2"*. These databases are maintained by the TOPS Lab [5] for the purpose of research as a means of providing service to the Wisconsin Department of Transportation [4]. Table 1 contains a sample of narratives from the database.

Table 1: Narratives from the accident dataset

ID	Narrative
1	Unit 1 and unit 2 were traveling northbound on antioch road south of 104th street. Unit 1 was traveling behind unit 2. Unit 2 was receiving verbal directions from a passenger in the back seat and braked, preparing to turn left without signaling. Unit 1 saw unit 2 stop in the road ahead of him and swerved to the left to attempt to avoid collision. Unit 2 initiated its left turn and unit 1 struck unit 2. The front passenger side corner of unit 1 struck the front driver's side corner of unit 2. Both units were removed from the scene by wilmot auto.
2	Unit 1 was traveling west on sth 50 in the 12500 block in the far right lane. Unit 2 was traveling west in the 12500 block. Unit 1 changed from the far right lane onto the left lane striking unit2 in the passenger front door area.
3	Unit 1 was traveling n/b inner lane on sherdian rd approaching 49 street. Unit 2 was also traveling n/b outer lane on sheridan rd. Behind unit 1. Unit 1 attempted a right turn from the inner lane of sheridan rd. Onto e/b 49 avenue. While conducting this manuver, unit 1 collided into unit 2 causing minimal damage. Occupants indicated no injuries.
4	Unit 1 was n/b on 28 ave. In the 4100 block and struck unit 2, which was legally parked in front of 4101-28 ave. Unit 1 left the scene n/b on 28 ave.

2.2. Tools and Techniques

2.2.1. Semantic Role Labeling (SRL)

In order to extract useful information from the text data, SRL is used in NLP. It is a technique where the words or phrases in the sentence are assigned their respective semantic role such as agent, goal, and result with respect to a verb. This is helpful in answering the question *"Who did what to whom?"*, given a sentence. Thus, it helps in yielding the predicate argument of a sentence. An example of SRL in a sample sentence "Jack loaded the truck with goods in the market on Tuesday", is shown in Table 2. The verb in the sentence is "loaded".

Table 2: SRL mapping for the sample sentence, where the verb is "loaded"

Jack	**the truck**	**with goods**	**in the market**	**on Tuesday**
ARG0	ARG1	ARG2	ARGM-LOC	ARGM-TMP

In general, the arguments labelled for the verbs are numbered arguments like ARG0, ARG1, ARG2, and so on. The numbered arguments represent semantic roles related to the predicate. In most cases, the meaning represented by the arguments is the same across different sentences. The meaning carried by certain arguments as per the prop bank [17] is shown in the table below. Apart from these numbered arguments, there are argument modifiers ARGM, which carries functional tags like MNR for manner, LOC for locative, TMP for temporal and many others as in table 3.

Table 3: List of arguments in Prop Bank [17]

ARGUMENTS	MEANING
ARG0	Agent
ARG1	Patient
ARG2	Instrument, attribute, benefactive
ARG3	Starting point, attribute, benefactive
ARG4	Ending point
ARGM-TMP	When?
ARGM-LOC	Where?
ARGM-DIR	Where to/from?
ARGM-MNR	How?
ARGM-PRP/CAU	Why?

2.2.2. AllenNLP

AllenNLP provides a complete platform for deep learning and NLP research that comprises existing implemented NLP models in order to support research [6]. It provides a high-level interface to many complex NLP tasks such as vision, language tasks, transformer experiments, etc. AllenNLP is a library which provides APIs which are capable of batching data intelligently, an experiment framework which is modular and an abstraction for low-level and common operations performed with text [9]. We used AllenNLP in this work to obtain semantic role labels for all the sentences of crash narratives.

2.2.3. BERT

BERT is a language representational model which is pretrained using various corpora. This pre-training phase makes BERT achieve an overall understanding of the natural language and one can fine tune this model on a specific task to achieve better results. The functioning of BERT depends on the transformer architecture [8]. BERT is a breakthrough because of its ability to capture the context of every word with respect to the sentence and also known for processing every word parallelly to generate the word embedding. Hence, BERT can exploit the use of GPU for better performance. The AllenNLP system uses BERT for its processing.

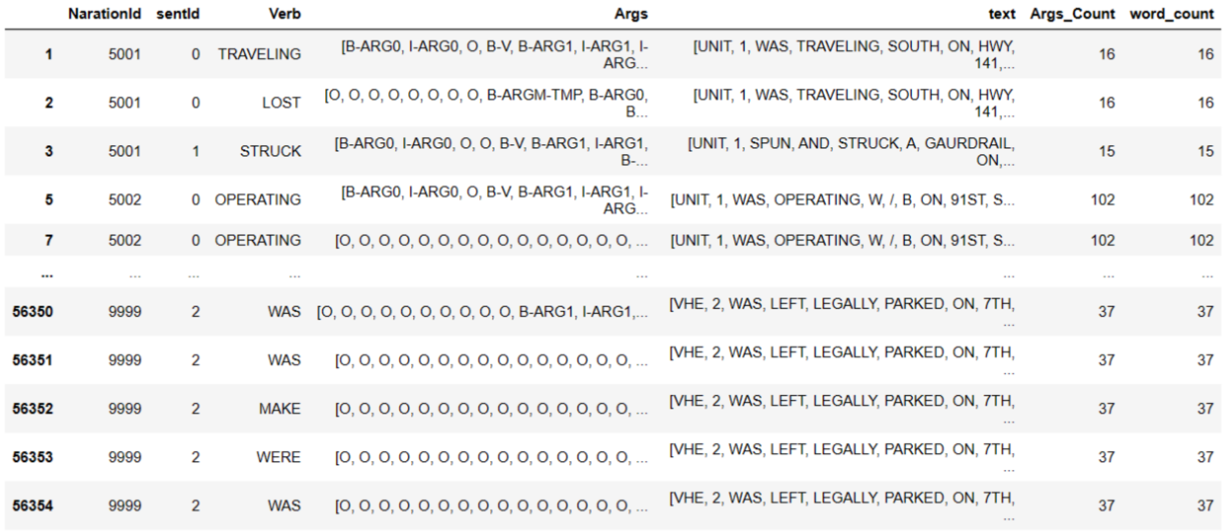
2.3. Methodology

2.3.1. Data Preparation

Each narrative from the dataset consists of multiple sentences in them which convey the complete details about the accident that occurred. In our thesis, we have made use of the *"Predictor"* class of the AllenNLP on each of these sentences in order to generate the SRL for each verb in the sentence. The generated output now provides a clear picture of "*Who did what to Whom?*". The "*structured-prediction-srl-bert*" is the model card that is used in the predictor function of AllenNLP. This model card is based on the BERT model that makes use of only the linear classification layer and no other parameters. The predictor generates arguments for each and every verb in a sentence. Using these verbs and their respective arguments, a data frame is created using pandas framework. The data frame also contains the sentence id as well the narrative id to which the sentences of those verbs belong to. The features of the data frame are as shown in Table 4.

Table 4: Features constructed after the SRL generation for each sentence.

Features	Description
NarrationId	Unique id for each narrative
sendId	Unique sentence id for each of the sentences in the narrative
Verb	Verb present in a sentence
Args	List of all the arguments generated by the AllenNLP predictor for each word
text	List of all the words for which a corresponding argument is generated by the AllenNLP
Args_count	Total number of Arguments generated for each sentence
word_count	Total number of words present in the text feature



	NarationId	sentId	Verb	Args	text	Args_Count	word_count
1	5001	0	TRAVELING	[B-ARG0, I-ARG0, O, B-V, B-ARG1, I-ARG1, I-ARG...	[UNIT, 1, WAS, TRAVELING, SOUTH, ON, HWY, 141,...	16	16
2	5001	0	LOST	[O, O, O, O, O, O, O, O, B-ARGM-TMP, B-ARG0, B...	[UNIT, 1, WAS, TRAVELING, SOUTH, ON, HWY, 141,...	16	16
3	5001	1	STRUCK	[B-ARG0, I-ARG0, O, O, B-V, B-ARG1, I-ARG1, B-...	[UNIT, 1, SPUN, AND, STRUCK, A, GAURDRAIL, ON,...	15	15
5	5002	0	OPERATING	[B-ARG0, I-ARG0, O, B-V, B-ARG1, I-ARG1, I-ARG...	[UNIT, 1, WAS, OPERATING, W, /, B, ON, 91ST, S...	102	102
7	5002	0	OPERATING	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, ...	[UNIT, 1, WAS, OPERATING, W, /, B, ON, 91ST, S...	102	102
...	...	...	...	...	...	...	...
56350	9999	2	WAS	[O, O, O, O, O, O, O, O, O, O, B-ARG1, I-ARG1,...	[VHE, 2, WAS, LEFT, LEGALLY, PARKED, ON, 7TH, ...	37	37
56351	9999	2	WAS	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, ...	[VHE, 2, WAS, LEFT, LEGALLY, PARKED, ON, 7TH, ...	37	37
56352	9999	2	MAKE	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, ...	[VHE, 2, WAS, LEFT, LEGALLY, PARKED, ON, 7TH, ...	37	37
56353	9999	2	WERE	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, ...	[VHE, 2, WAS, LEFT, LEGALLY, PARKED, ON, 7TH, ...	37	37
56354	9999	2	WAS	[O, O, O, O, O, O, O, O, O, O, O, O, O, O, O, ...	[VHE, 2, WAS, LEFT, LEGALLY, PARKED, ON, 7TH, ...	37	37

Figure 1: Data frame constructed with the features of Table 4.

The "*text*" feature of the data frame corresponds to the argument values for each of the arguments in the "*Args*" column.

The data frame is used to structure the data using the "*nested class*" concepts of Object-Oriented Programming (OOP). A hierarchy of classes is composed from the available data format. This hierarchy helps maintain the origin of each verb which can be traced back to its own sentences and in turn to the narrative from which they came from.

For example, let *N1* be a sample narrative comprising of two sentences *s1* and *s2*. Let *v1*, *v2*, *v3* be the verbs from these sentences. After semantic role labelling using AllenNLP, for each of these verbs we get arguments and their corresponding words list. They can be represented as $a_1 \ldots a_n$ and $w_1 \ldots w_n$. respectively. The nested class structure can be visualized as in Figure 2.

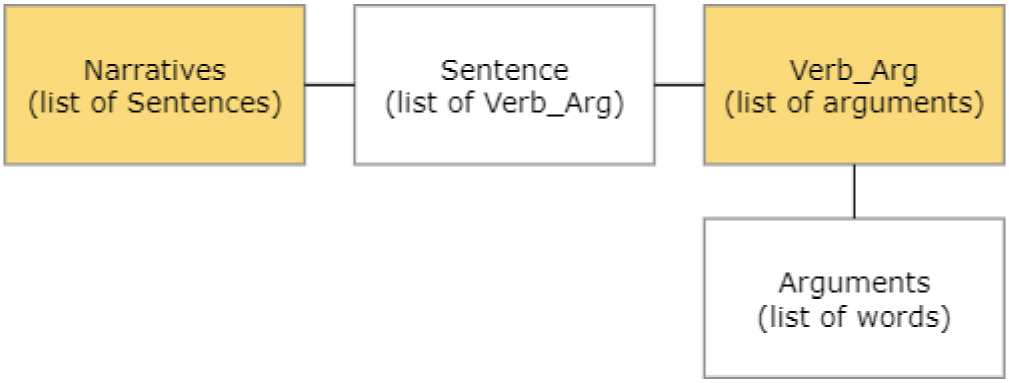


Figure 2: Nested class hierarchy for data in each narrative.

The common patterns are extracted from a list of narrative objects by using the data from the structured nested classes. Each "*Narrative*" class object consists of a unique narrative id and list of "*Sentence*" objects. Each *"Sentence"* object is composed of a sentence id and list of *"Verb_arg"* class objects. Each *"Verb_arg"* object comprises of a verb name, a list of arguments, *"arg"*. "*Arg"* consists of "*arg_name*" and a list of actual words found in the "text" part of the semantic role labelled result.

The semantic role labelling performed by AllenNLP, generated arguments for the verbs in the narratives. These arguments are in the BIO notation where "B" is the beginning of the argument, "I" being the inside and "O" being the outside part. For our algorithm, we have combined the "B-" and the subsequently following "I-" arguments as a single entity. For instance, B-ARG0, I-ARG0, I-ARG0 is combined as ARG0 by appending its respective values as well, if they occur in a sequence.

The arguments "O" and its respective words are not considered while computing common patterns. This is because, words of outside arguments do not carry any important information. Thus, semantic role labelling helps us focus on the primary and important words in a sentence with respect to the verbs in a sentence.

2.3.2. Common Pattern Extraction Algorithm

The common pattern extraction algorithm is divided into two phases.

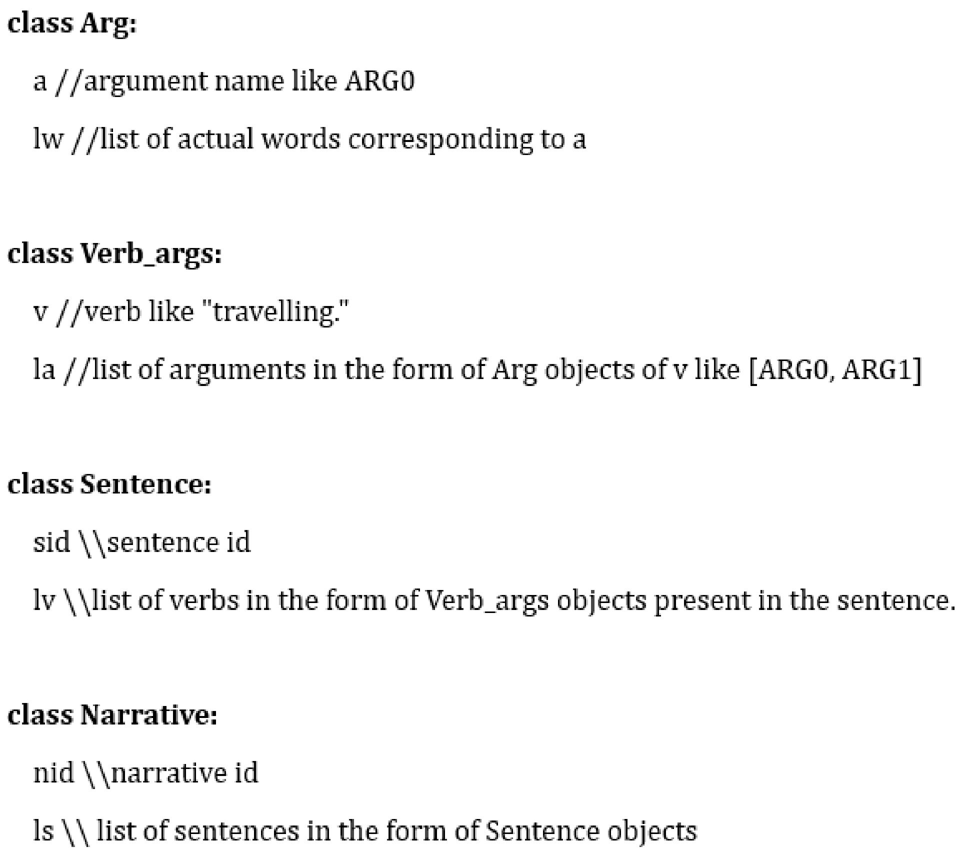
Phase 1: Extracting patterns from narratives

Phase 2: Finding the frequency of the patterns.

2.3.2.1. Extracting patterns from narratives:

The main core of the algorithm is to find the patterns in the narratives. The patterns are semantic roles in the form of verb and argument pairs. The semantic roles for all the verbs generated by AllenNLP are structured as nested class objects as discussed in the data preparation section. The algorithm also maintains data structure to store unique patterns and avoid duplicates.

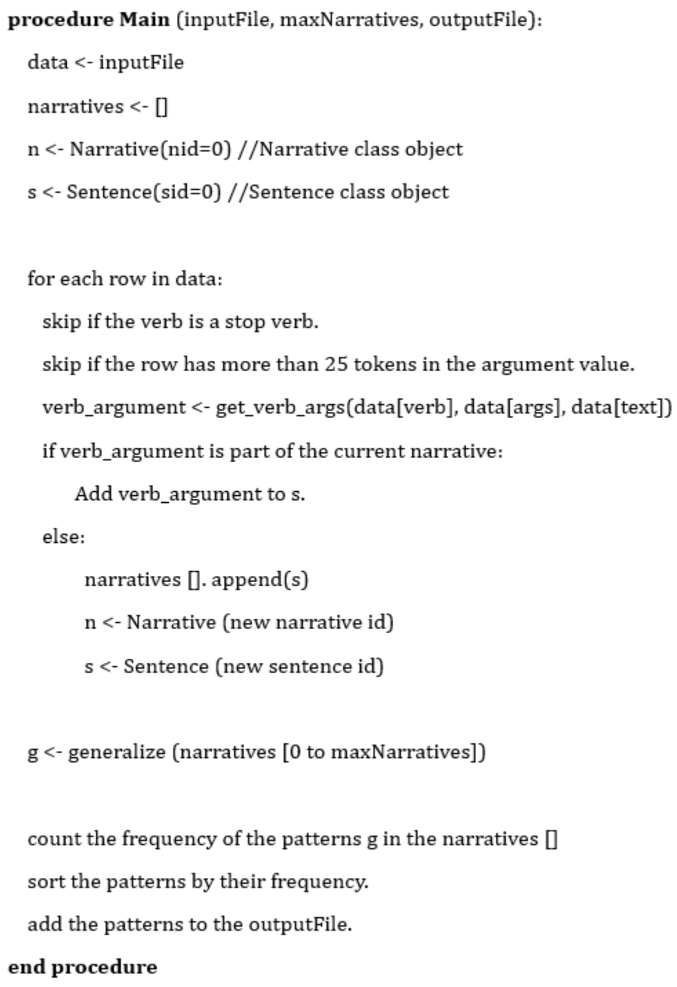
In order to find the patterns, our algorithm exploits this structure by penetrating deep to the lowest class level. The hierarchy of classes helps us maintain the verbs and its respective arguments tied together.



```
class Arg:
  a //argument name like ARG0
  lw //list of actual words corresponding to a

class Verb_args:
  v //verb like "travelling."
  la //list of arguments in the form of Arg objects of v like [ARG0, ARG1]

class Sentence:
  sid \\sentence id
  lv \\list of verbs in the form of Verb_args objects present in the sentence.

class Narrative:
  nid \\narrative id
  ls \\ list of sentences in the form of Sentence objects
```

Figure 3: Pseudocode of the class structure with attributes only

The list of words corresponding to these patterns is then used to find the commonly occurring words among them, which in turn makes the common pattern. The pseudocode for the class structure depicting their attributes alone is shown in Figure 3. The main procedure in Figure 4 accepts input file, maximum number of narratives to find patterns for and the output file name in which the common patterns get stored along with their frequencies. The input file consists of the verbs, semantic roles and respective words for all the narratives as shown in Figure 1.



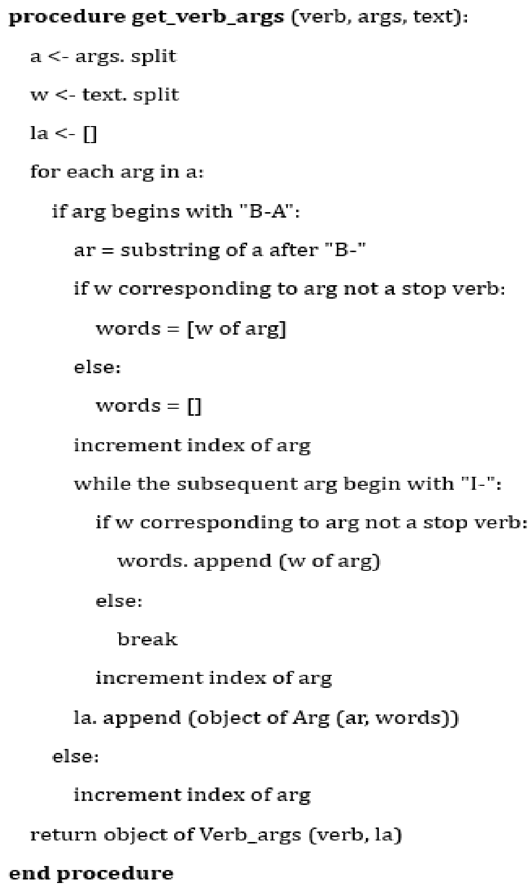
```
procedure Main (inputFile, maxNarratives, outputFile):
  data <- inputFile
  narratives <- []
  n <- Narrative(nid=0) //Narrative class object
  s <- Sentence(sid=0) //Sentence class object

  for each row in data:
    skip if the verb is a stop verb.
    skip if the row has more than 25 tokens in the argument value.
    verb_argument <- get_verb_args(data[verb], data[args], data[text])
    if verb_argument is part of the current narrative:
        Add verb_argument to s.
    else:
         narratives []. append(s)
         n <- Narrative (new narrative id)
         s <- Sentence (new sentence id)

  g <- generalize (narratives [0 to maxNarratives])

  count the frequency of the patterns g in the narratives []
  sort the patterns by their frequency.
  add the patterns to the outputFile.
end procedure
```

Figure 4: The main function of the common pattern extraction algorithm

Let us consider the maximum number of narratives from which we want to extract the common patterns be three. This is considered as the second parameter for the *Main()* procedure.

In the *Main()* procedure, each row of the input as in Figure 1, is processed individually. Here, the processing of a row is skipped if the total number of semantic label values in

‘Args’ field is greater than 25. Otherwise, the verb, values of *‘Args’* and values of *‘text’* field are passed on to the *get_verb_args()* procedure.



```
procedure get_verb_args (verb, args, text):
  a <- args. split
  w <- text. split
  la <- []
  for each arg in a:
    if arg begins with "B-A":
      ar = substring of a after "B-"
      if w corresponding to arg not a stop verb:
        words = [w of arg]
      else:
        words = []
      increment index of arg
      while the subsequent arg begin with "I-":
        if w corresponding to arg not a stop verb:
          words. append (w of arg)
        else:
          break
        increment index of arg
      la. append (object of Arg (ar, words))
    else:
      increment index of arg
  return object of Verb_args (verb, la)
end procedure
```

Figure 5: Function to get the verb arguments for a given verb, arguments, and its corresponding words.

For example, let us consider the verbs ‘travelling’ and ‘receiving’ from the input data frame as in Figure 1. The *get_verb_args ()* goes through every item of the ‘*argument list’* to construct the arguments for each verb in a certain manner. There are several arguments for one single verb like B-ARG0, I-ARG0, B-ARG1 etc. The arguments are converted to a form such that there is one single entity for ARG0, ARG1 etc. This is achieved by combining the intermediate components for the respective argument and stripping the prefixes. The *get_arg_verbs()* procedure as in Figure 5 , thus generates arguments for the verbs and returns the list of arguments wrapped around its respective verb with the ‘*Verb_arg*’ class structure. For the two verbs considered, the output of *get_verb_args()* is as below.

```
TRAVELING:
        ARG0: UNIT 1 UNIT 2
        ARG1: NORTHBOUND
        ARGM-LOC: ANTIOCH ROAD SOUTH 104TH STREET

RECEIVING:
        ARG0: UNIT 2
        ARG1: VERBAL DIRECTIONS
        ARG2: PASSENGER BACK SEAT
```

The semantic role representation is formatted by the ‘*get_verb_args()*’ procedure for all the rows of input in the above manner.

After semantic role labeling results are consolidated, each sentence is represented as a list of verbs and its arguments. The algorithm generalizes two sentences by first finding their common sequence of verbs. Next, each common verb and its argument structure is generalized using the procedure described earlier. As an example, consider the following two sentences:

Sentence 1: “*The driver lost control and the vehicle fell in the ditch*.”

Sentence 2: “*Driver lost his control and the car fell in nearby ditch.”*

Their semantic role representations will be as follows:

Sentence 1:

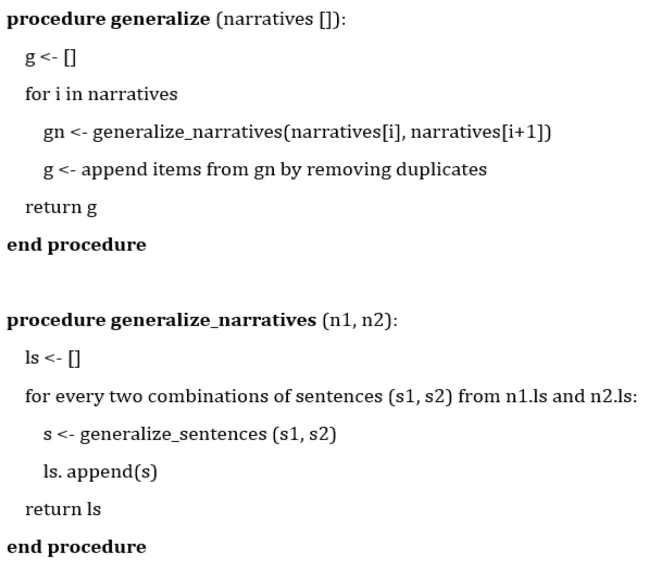
LOST [ARG0: “the driver”, ARG1: “control”] - FELL [ARG0: “the vehicle”, ARG1: “in the ditch”]

Sentence 2: LOST [ARG0: “driver”, ARG1: “his control”] - FELL [ARG0: “the car”, ARG1: “in nearby ditch”]

The sequence of common verbs here is: LOST-FELL. The argument structures will be generalized for each verb. For LOST, it will be [ARG0: “driver”, ARG1: “control”] and for FELL, it will be [ARG1: “in ditch”]. Note that stop-words, like “*the*”, are ignored and only sequence of common words are considered (thus skipping “*nearby*” in this example). Also note that, ARG0 for FELL is missing (or generalized over). Thus, the common pattern from the two sentences will be:

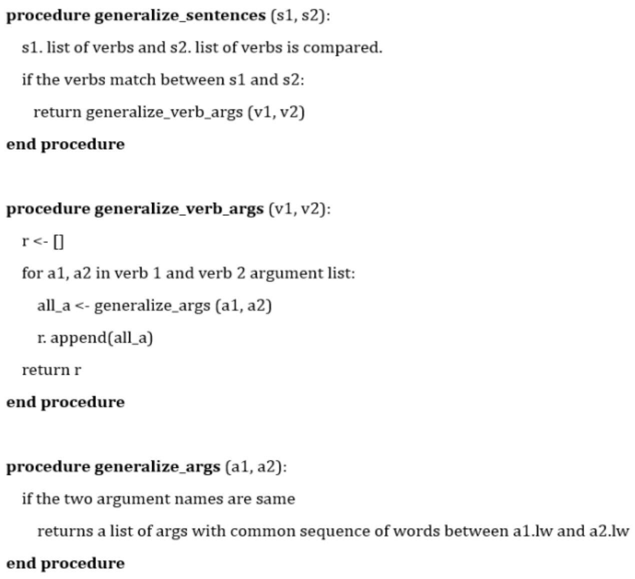
Common pattern: LOST [ARG0: “driver”, ARG1: “control”] – FELL [ARG1: “in ditch”].

Two narratives are generalized by generalizing every pair of sentences in them. In this process, the duplicate patterns are removed. The following pseudocode in Figure 6 and Figure 7 shows the process.



```
procedure generalize (narratives []):
  g <- []
  for i in narratives
    gn <- generalize_narratives(narratives[i], narratives[i+1])
    g <- append items from gn by removing duplicates
  return g
end procedure

procedure generalize_narratives (n1, n2):
  ls <- []
  for every two combinations of sentences (s1, s2) from n1.ls and n2.ls:
    s <- generalize_sentences (s1, s2)
    ls. append(s)
  return ls
end procedure
```

Figure 6: Set of procedures which generalizes patterns from narratives.



```
procedure generalize_sentences (s1, s2):
  s1. list of verbs and s2. list of verbs is compared.
  if the verbs match between s1 and s2:
    return generalize_verb_args (v1, v2)
end procedure

procedure generalize_verb_args (v1, v2):
  r <- []
  for a1, a2 in verb 1 and verb 2 argument list:
    all_a <- generalize_args (a1, a2)
    r. append(all_a)
  return r
end procedure

procedure generalize_args (a1, a2):
  if the two argument names are same
    returns a list of args with common sequence of words between a1.lw and a2.lw
end procedure
```

Figure 7: Pattern generalizing procedures in the sentence, verbs, arguments, and word level.

However, in this work we found that concatenating sequence of verbs from all the sentences in a narrative into one long sequence and then generalizing them yielded better results because, then the sequence of verbs goes across sentences.

2.3.2.2. Finding the frequency of the patterns

This is the second phase of the algorithm. Once the patterns are obtained in the first phase, we want to check for the frequency of those patterns in the narratives. Each pattern is checked for its occurrences in the all the narratives. They are then sorted based on their frequencies having the most commonly occurring pattern in the top. Thus, the target audience can get important information from the patterns.

Results

From the dataset, 30,000 narratives were considered. These narratives were subjected to semantic role labelling using AllenNLP package. They are then processed and stored in a data frame which has about 247,000 rows of data. Each row corresponds to the verbs from the narratives belonging to different or same sentences.

Upon sending the processed input to the "*common pattern extraction*" algorithm, nearly 1.8 million patterns were generated. These patterns are then counted for their frequency. Some of the interesting patterns are discussed as follows.

Pattern 1:

Frequency: 16

Verbs: 1

LOST:

ARG1: ***CONTROL***

ARGM-LOC: ***ICY ROADWAY***

The pattern conveys the information that the accident has occurred because the driver has lost control on an icy roadway which has occurred 16 times. There are also similar patterns which convey the same information each occurring '*n*' number of times. This emphasizes that icy road conditions being one of the major reasons for most of the accidents.

Pattern 2:

Frequency: 17

Verbs: 2

LOST:

ARG0: ***UNIT***

ARG1: ***CONTROL***

ENTERED:

ARG1: ***DITCH***

The above pattern reveals that the vehicle involved in the accident has lost control and fell into a ditch. This has occurred 17 times in the narratives. As discussed earlier, different variation of patterns revealing the same piece of information exists in the output patterns result. Thus, one can understand that losing control is one of the major factors contributing to crashes in that zone.

Here is another interesting pattern where it shows the highway number in which the accident has occurred with a good frequency. This helps one understand that the particular area has witnessed considerable accidents.

Pattern 3:

Frequency: 3

Verbs: 4

TRAVELING:

ARG0: ***UNIT ONE***

ARG1: ***I39/90***

LOST:

ARG1: CONTROL

CRASHED:

ARGM-DIR: INTO

REPORTED:

ARG1: INJURIES

The pattern 4 reveals that some accidents have occurred because the driver fell asleep.

Pattern 4:

Frequency: 3

Verbs: 3

STATED:

ARG1: HE

FELL:

ARG1: HE

ARG2: ***ASLEEP***

ARGM-TMP: WHILE DRIVING

***STATED**:*

ARG1: HE

Similarly, patterns even reveal other interesting information which are otherwise ignored. The below pattern 5 is one of the longest patterns obtained from our result set. It conveys that the accident has occurred due to speeding by the driver involving someone dead in the accident scene.

Pattern 5:

Frequency: 2

Verbs: 5

***TRAVELING**:*

ARG0: UNIT

*ARGM-MNR: **HIGH SPEED***

***TRAVELING**:*

ARG0: UNIT

ARG1: WEST

***CAME**:*

ARG1: UNIT

ARG2: REST

***DIED**:*

ARG1: UNIT

ARGM-LOC: SCENE

SEE:

ARG1: POLICE

Out of the 1.8 million patterns generated in the result set, a sample set of some interesting patterns has been included in the appendix section of the thesis.

Limitations

The common pattern extraction algorithm could generate more meaningful patterns when it is provided with dataset of a particular type of accidents, for example data related to single vehicle crashes. However, the dataset we have utilized contained no specific types of crashes, instead they were general crashes. If, for example, the dataset contained only accidents from a construction zone, patterns would be more focused, domain-specific and rich in details. Another limitation is due to the accuracy of the semantic role labelling. The SRL performed by AllenNLP sometimes yields large number of tokens for the arguments. If the SRL is performed more accurately, more interesting patterns could be yielded.

Conclusion

In this work, we developed a method to obtain patterns of semantic role labels. Semantic role labeling information, in the form of verbs and their argument structures, provides the most meaningful representation of a sentence. The patterns consist of sequences of verbs along with their arguments. These were obtained by generalizing semantic role labeling information from sentences. The method was applied to accident narratives and several patterns were found whose prevalence was measured by their frequency of occurrence in the narratives. The insights from such patterns could be used to analyze the causes and manners of accidents.

In future, restricting the narratives to particular types of accidents could lead to more specific and insightful patterns. The research could be extended by merging the patterns of different combinations as a single pattern in a meaningful way. Actual experts who will use the system could be asked to analyze the system patterns and evaluate how useful it is for them.

REFERENCES

[1] Sayed, M. A., Qin, X., Kate, R. J., Anisuzzaman, D. M., & Yu, Z. (2021). Identification and analysis of misclassified work-zone crashes using text mining techniques. Accident Analysis & Prevention, 159, 106211.

[2] Macedo, J. B., Ramos, P. M., Maior, C. B., Moura, M. J., Lins, I. D., & Vilela, R. F. (2022). Identifying low-quality patterns in accident reports from textual data. International journal of occupational safety and ergonomics, 1-13.

[3] Abay, K. A. (2015). Investigating the nature and impact of reporting bias in road crash data. Transportation research part A: policy and practice, 71, 31-45.

[4] https://transportal.cee.wisc.edu/services/crash-data/

[5] https://topslab.wisc.edu

[6] https://allenai.org/allennlp/software/allennlp-library

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[8] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[9] Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., ... & Zettlemoyer, L. (2018). Allennlp: A deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640.

[10] Zhang, F., Fleyeh, H., Wang, X., & Lu, M. (2019). Construction site accident analysis using text mining and natural language processing techniques. Automation in Construction, 99, 238-248.

[11] Feng, D., & Chen, H. (2021). A small samples training framework for deep Learning-based automatic information extraction: Case study of construction accident news reports analysis. Advanced Engineering Informatics, 47, 101256.

[12] Single, J. I., Schmidt, J., & Denecke, J. (2020). Knowledge acquisition from chemical accident databases using an ontology-based method and natural language processing. Safety Science, 129, 104747.

[13] Suh, Y. (2021). Sectoral patterns of accident process for occupational safety using narrative texts of OSHA database. Safety science, 142, 105363.

[14] Chokor, A., Naganathan, H., Chong, W. K., & El Asmar, M. (2016). Analyzing Arizona OSHA injury reports using unsupervised machine learning. Procedia engineering, 145, 1588-1593.

[15] https://www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec6-srl.pdf

[16] https://wisconsindot.gov

[17] Palmer, Martha, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. Computational Linguistics, 31(1):71–106

APPENDICES

```
8

---------------
Pattern:

Frequency: 857
Verbs: 1

LOST:
ARG1: CONTROL

23

---------------
Pattern:

Frequency: 546
Verbs: 1

STRUCK:
ARG0: UNIT

29

---------------
Pattern:

Frequency: 518
Verbs: 1

REPORTED:
ARG1: INJURIES


56

---------------
Pattern:

Frequency: 433
Verbs: 1
```

```
REPORTED:
ARG1: INJURIES
ARG1: NO

60

---------------
Pattern:

Frequency: 422
Verbs: 1

CAUSING:
ARG1: DAMAGE

---------------
Pattern:

Frequency: 284
Verbs: 2

TRAVELING:
ARG0: 1

LOST:
ARG1: CONTROL


1202

---------------
Pattern:

Frequency: 125
Verbs: 1

ENTERED:
ARG1: DITCH



478

---------------
Pattern:
```

```
Frequency: 184
Verbs: 1

FAILED:
ARG1: UNIT
ARG2: YIELD

487

---------------
Pattern:

Frequency: 183
Verbs: 1

STRUCK:
ARG1: REAR

14077

---------------
Pattern:

Frequency: 30
Verbs: 1

STRUCK:
ARG0: UNIT
ARG1: TREE


8907

---------------
Pattern:

Frequency: 41
Verbs: 1

STRUCK:
ARG0: UNIT
ARG1: POLE


9054
```

```
---------------
Pattern:

Frequency: 41
Verbs: 1

BACKING:
ARG1: UNIT 1
ARGM-DIR: OUT DRIVEWAY

7047

---------------
Pattern:

Frequency: 48
Verbs: 1

STRUCK:
ARG1: MEDIAN


4917

---------------
Pattern:

Frequency: 57
Verbs: 3

TRAVELING:
ARG0: UNIT

FAILED:
ARG1: UNIT
ARG2: YIELD RIGHT

YIELD:
ARG1: WAY

9737

---------------
Pattern:

Frequency: 39
```

```
Verbs: 2

FAILED:
ARG2: STOP

STOP:
ARGM-LOC: SIGN

9876

---------------
Pattern:

Frequency: 38
Verbs: 1

TRAVELING:
ARG0: 2
ARGM-LOC: W. AVE


9958

---------------
Pattern:

Frequency: 38
Verbs: 2

STATED:
ARG0: UNIT
ARG1: SHE DID NOT

SEE:
ARG0: SHE
ARG1: UNIT

23796

---------------
Pattern:

Frequency: 22
Verbs: 1

ROTATED:
```

```
ARG1: UNIT


43768

---------------
Pattern:

Frequency: 15
Verbs: 1

STRUCK:
ARG0: UNIT
ARG1: FENCE

29372

---------------
Pattern:

Frequency: 19
Verbs: 2

TRAVELING:
ARG0: UNIT

STRUCK:
ARG2: UNIT
ARG1: SIDE


30423

---------------
Pattern:

Frequency: 19
Verbs: 1

CITED:
ARG1: UNIT 1
ARG2: UNSAFE LANE


30424
```

```
---------------
Pattern:

Frequency: 19
Verbs: 2

CITED:
ARG1: UNIT 1
ARG2: FOLLOWING TOO CLOSELY

FOLLOWING:
ARGM-MNR: CLOSELY

29368

---------------
Pattern:

Frequency: 19
Verbs: 1

STRUCK:
ARG2: FRONT
ARG1: FRONT UNIT



43928

---------------
Pattern:

Frequency: 15
Verbs: 1

STRUCK:
ARG1: MEDIAN WALL

64655

---------------
Pattern:

Frequency: 12
Verbs: 1
```

```
TRAVELING:
ARG1: SOUTH
ARG1: STREET



87461

---------------
Pattern:

Frequency: 10
Verbs: 1

PAYING:
ARG0: HE
ARGM-NEG: NOT
ARG1: ATTENTION

87498

---------------
Pattern:

Frequency: 10
Verbs: 1

STATED:
ARG0: DRIVER
ARG1: SLIPPERY

88858

---------------
Pattern:

Frequency: 10
Verbs: 1

TRAVELING:
ARGM-LOC: US 51



126308
```

```
---------------
Pattern:

Frequency: 8
Verbs: 2

DRIVING:
ARGM-MNR: TOO FAST

LOST:
ARG1: CONTROL

206808

---------------
Pattern:

Frequency: 6
Verbs: 1

DEPLOYED:
ARG1: FRONT AIRBAGS

260967

---------------
Pattern:

Frequency: 6
Verbs: 1

LOOKED:
ARG1: PHONE


623567

---------------
Pattern:

Frequency: 3
Verbs: 4

TRAVELING:
ARGM-TMP: UPON INVESTIGATION
ARG0: UNITS
```

```
ARG1: S / B
ARGM-LOC: LANE 1
ARGM-LOC: HEAVY

STOP:
ARG1: DRIVER
ARGM-NEG: NOT
ARGM-TMP: TIME
ARGM-ADV: REAR ENDING UNIT 2 CAUSING DAMAGE

CAUSING:
ARG1: DAMAGE UNIT 2 BUMPER TRUNK

CITED:
ARG1: DRIVER


1832942

---------------
Pattern:

Frequency: 2
Verbs: 2

WANTED:
ARG0: HE
ARG1: HIMSELF

KILL:
ARG0: HE
ARG1: HIMSELF

1833077

---------------
Pattern:

Frequency: 2
Verbs: 6

TRAVELING:
ARG0: UNIT 1

LOST:
ARG0: UNIT
```

```
ARG1: CONTROL

ROTATED:
ARGM-MNR: CLOCKWISE

CAME:
ARG2: REST

CAME:
ARG1: UNIT
ARG2: LANE

REST:
ARG1: UNIT
ARGM-LOC: LANE



1843034

---------------
Pattern:

Frequency: 2
Verbs: 2

ADMITTED:
ARG1: PHONE

USING:
ARG1: PHONE

1832942

---------------
Pattern:

Frequency: 2
Verbs: 2

WANTED:
ARG0: HE
ARG1: HIMSELF

KILL:
ARG0: HE
```

```
ARG1: HIMSELF

1833077

---------------
Pattern:

Frequency: 2
Verbs: 6

TRAVELING:
ARG0: UNIT 1

LOST:
ARG0: UNIT
ARG1: CONTROL

ROTATED:
ARGM-MNR: CLOCKWISE

CAME:
ARG2: REST

CAME:
ARG1: UNIT
ARG2: LANE

REST:
ARG1: UNIT
ARGM-LOC: LANE


23796

---------------
Pattern:

Frequency: 22
Verbs: 1

ROTATED:
ARG1: UNIT

260967

---------------
```

Pattern:

Frequency: 6
Verbs: 1

LOOKED:
ARG1: PHONE